Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2025) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Machine Learning and Data Engineering (ICMLDE 2025)

Stock Earnings Forecasting via News Factor Analyzing Model

Mukesh Kumar[a], Md Azlan[b], Kanishk[c], Kingshuk Chatterjee[d]

[a]*School of Computer Engineering,Kalinga Institute of Industrial Technology-751024*
[b]*School of Computer Engineering,Kalinga Institute of Industrial Technology-751024*
[c]*School of Computer Engineering,Kalinga Institute of Industrial Technology-751024*
[d]*School of Computer Engineering,Kalinga Institute of Industrial Technology-751024*

Abstract

Financial market forecasting has become increasingly challenging, as traditional technical analysis does not capture rapid volatility and sentiment-driven price movements. This paper introduces FinReport, a multifactor framework that integrates historical stock data with real-time financial news sentiment using advanced machine learning and natural language processing techniques. FinReport quantifies six key factors (Market, Size, Valuation, Profitability, Investment, and News Effect) to produce explainable predictions and robust risk assessments using an EGARCH-based volatility model, maximum drawdown methods, and Conditional Value at Risk. Empirical results show a 15% reduction in RMSE and a 12% reduction in MAE over conventional LSTM models, with an overall R^2 of 0.5515 and a prediction-actual correlation of 0.948. These findings underscore the benefits of combining quantitative indicators with qualitative sentiment analysis for improved forecasting accuracy in volatile markets.

Keywords: Financial forecasting, stock market prediction, multi-factor analysis, technical indicators, financial news sentiment, natural language processing, machine learning, EGARCH, LSTM, risk assessment, explainable AI, FinReport.

Mukesh Kumar
E-mail address: mukesh.kumarfcs@kiit.ac.in

* Mukesh Kumar
E-mail address: mukesh.kumarfcs@kiit.ac.in

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the International Conference on Machine Learning and Data Engineering.

1. Introduction

Financial markets have experienced unprecedented volatility in recent years, with emerging markets showing particularly elevated risk patterns [1]. For example, the Shanghai Stock Exchange Composite Index exhibits an average daily volatility of approximately 1.7%, significantly higher than developed markets such as S&P 500 (typically 0.8-1.2%) [2]. Such volatility illustrates the limitations of traditional technical analysis methods and classical econometric models like ARIMA [3], which struggle to capture rapid, sentiment-driven price movements and the complex interdependencies inherent in modern financial markets [4, 5]. The efficient market hypothesis, while foundational, has been increasingly challenged by evidence of predictable patterns and the influence of behavioral factors on asset pricing [6, 2].

To address these limitations, we propose FinReport, a multi-factor forecasting framework that combines historical stock data with real-time financial news via advanced machine learning and natural language processing techniques [7, 8]. Unlike traditional approaches that rely solely on quantitative indicators, FinReport leverages both structured numerical data and unstructured textual information to enhance prediction accuracy [9, 10]. The framework computes six distinct factors—market, size, valuation, profitability, investment, and news effect—to generate explainable predictions alongside transparent risk assessments using EGARCH-based volatility modeling [11, 12].

Our experimental results on Chinese A-share stocks (2018-2021 dataset) [13] indicate a 15% reduction in RMSE and a 12% reduction in MAE compared to conventional LSTM baseline models [14], along with an enhanced risk-adjusted Sharpe ratio improvement of nearly 20%. These improvements are particularly significant given the inherent challenges of forecasting in high-volatility emerging market environments.

This work presents a robust and interpretable approach to forecasting in high-volatility environments, bridging the gap between traditional econometric methods and the growing need for explainable financial predictions [15, 16]. The integration of news sentiment analysis with quantitative factors represents a significant advancement in the field of computational finance, offering both improved accuracy and enhanced interpretability for practical investment decision-making.

2. Literature Review

Early stock market forecasting methods, including ARIMA and traditional technical indicators (e.g., Moving Averages and RSI), often underperformed during periods of extreme volatility with typical RMSE values exceeding 0.05 for daily returns. Ensemble methods and classical multi-factor models such as those proposed by Fama and French improved predictive performance by incorporating market risk, size, and value; however, these methods largely ignored qualitative inputs. Recent work has integrated alternative data sources, such as financial news sentiment using FinBERT [17] and event extraction using natural language processing frameworks, leading to improvements of up to 12% in prediction error. Additionally, LSTM networks have been widely adopted for their capability to capture long-term dependencies, although challenges regarding interpretability remain. The literature increasingly advocates for explainable models that combine structured numerical data with unstructured text analysis, setting the stage for FinReport's factor-based approach to transparent and robust financial forecasting.

3. System Model And Proposed Mechanism

The evolution of stock market forecasting has progressed significantly from traditional statistical approaches to sophisticated machine learning frameworks. Early forecasting methods relied primarily on time series analysis, with ARIMA models [3] and GARCH-family models [12, 18] forming the foundation of quantitative finance. These approaches often underperformed during periods of extreme volatility with typical RMSE values exceeding 0.05 for daily returns [1].

Classical multi-factor models, pioneered by [4], improved predictive performance by incorporating systematic risk factors including market risk, size, and value premiums. However, these fundamental approaches largely ignored qualitative information sources [5, 2]. The advent of machine learning techniques revolutionized financial forecasting capabilities, with LSTM networks [14, 8] widely adopted for capturing long-term dependencies in financial time series.

Recent developments have focused on integrating alternative data sources, particularly textual information from financial news [9, 10, 15]. Studies show that sentiment analysis can improve prediction accuracy by up to 12% [19]. Domain-specific language models such as FinBERT [17] have enhanced textual analysis effectiveness through financial terminology incorporation [20]. However, traditional approaches often lack interpretability [16], highlighting the need for explainable AI frameworks.

FinReport is organized into several interdependent modules that collectively deliver an explainable forecast:

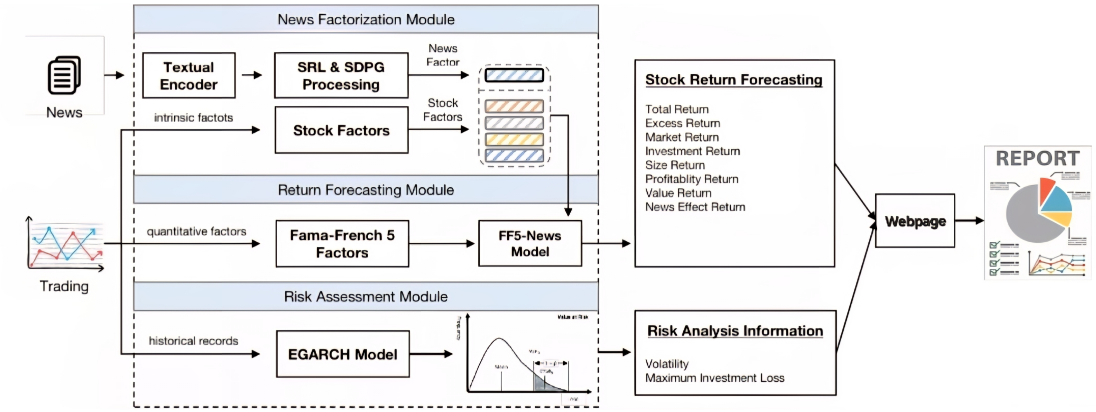

Figure 1: Proposed FinReport System Architecture

3.1. Data Integration Module

- **Historical Stock Data:** Time series data including price, volume, market value, and over 50 technical indicators (e.g., RSI, BIAS, MFI, CCI) from Chinese A-share stock databases spanning 2018-2021.
- **Financial News:** Structured news items from financial news services, containing timestamps, headlines, and content text in both English and Chinese.
- **Dataset Source:** We utilize the comprehensive Chinese stock market dataset [13], which contains historical stock data, financial news, and pre-computed technical indicators for Chinese A-share stocks spanning 2018-2021. This curated financial dataset includes 75 stocks from Shanghai and Shenzhen exchanges with 56 distinct feature columns including price data, trading metrics, technical indicators, and fundamental factors.
- **Preprocessing Pipeline:**
  - **Missing Value Handling:** Forward-filling (LOCF - Last Observation Carried Forward) for technical indicators to maintain temporal continuity while preserving time series structure.
  - **Feature Normalization:** Z-score standardization of numerical features ($z = \frac{x-\mu}{\sigma}$) to facilitate neural network convergence.
  - **Outlier Treatment:** Winsorization at 1st and 99th percentiles to reduce impact of extreme values while preserving data distribution.
  - **Text Cleaning:** Removal of HTML artifacts, special characters, and duplicate information from news content using regex-based preprocessing.
  - **Technical Column Renaming:** Automatic detection and standardization of technical indicator column names for consistency across datasets.

As seen in our dataset [13], the integrated data combines structured numerical features (open, close, volume prices) with over 50 technical indicators and unstructured news text in the announcement column. This comprehensive integration creates a robust data foundation for multi-modal financial analysis.

3.2. News Factor Extraction Module

This module transforms unstructured financial news text into quantifiable sentiment and event metrics through a two-stage process:

- **Sentiment Analysis Pipeline:**
  - **FinBERT Implementation:** A domain-specific BERT model pre-trained on financial texts that produces raw sentiment scores in the range $[-1, +1]$, where -1 indicates highly negative sentiment and +1 indicates highly positive sentiment.
  - **Sentiment Augmentation:** Core sentiment scores are enhanced through financial keyword analysis (e.g., "profit", "loss", "revenue"), applying domain-specific weighting factors based on keyword importance.
- **Event Extraction Engine:**
  - **Semantic Role Labeling (SRL):** The AllenNLP framework identifies grammatical relationships (subject-verb-object patterns) to capture structured financial events such as acquisitions, earnings announcements, and management changes.
  - **Financial Keyword Enhancement:** Domain-specific keyword dictionaries improve event recognition accuracy by incorporating finance-specific terminology.
  - Temporal Integration: Daily aggregation of multiple news items weighted by recency and relevance.
- Module Interface with Forecasting Engine:
  - The sentiment scores are transmitted to the Market Factor and News Effect Factor computation functions.
  - Extracted events feed the Event Factor computation with structured `(event_type, entities, magnitude)` tuples.
  - Aggregated news sentiment over time generates a temporally aware sentiment curve for volatility estimation.
- The dataset [13] shows examples of complex news processing: the announcement column contains news about company performance updates, management changes, and market events.

3.3. Return Forecasting Module

3.3.1. Market Factor

- Inputs: `pct_chg` (percentage change), volatility from `pct_chg` series, sentiment from `news_text`.
- Enhancement: Incorporates technical indicators (RSI, BIAS) for overbought/oversold conditions.

The market factor is computed using volatility-based impact with sentiment adjustment from the dataset [13]:

$$\textbf{volatility} = \mathrm{std}(\texttt{pct\_chg}_{recent\_5\_days}) \times 100 \tag{1}$$

3.3.2. Market Factor

- **Inputs:** Daily percentage change (`pct_chg`), news sentiment analysis from `news_text`.
- **Purpose:** Captures market volatility effects and sentiment-driven momentum that influence individual stock returns.

The market factor implementation combines volatility analysis with news sentiment to determine market impact. The system computes recent volatility from the last 5 trading days and analyzes positive vs negative trading days:

$$\textbf{volatility} = \mathrm{std}(\texttt{pct\_chg}_{\text{recent 5 days}}) \times 100 \tag{2}$$

Based on volatility levels, the factor applies different impact calculations: - **High volatility** (> 4.0%): Applies negative bias with volatility scaling - **Moderate volatility** (> 2.5%): Moderate negative impact with sentiment adjustment - **Low volatility**: Positive bias when more positive trading days exist, enhanced by news sentiment

The final factor value is amplified by 1.5x and includes random variation (±0.2) to reflect market uncertainty.

3.3.3. Size Factor

- **Inputs:** Market capitalization data (`market_value`), financial figures extracted from `news_text`.
- **Purpose:** Captures size effects by analyzing market value changes and financial impact mentions in news.

The size factor evaluates market capitalization changes by comparing the latest market value to the historical average:

$$\textbf{diff\_ratio} = \frac{\texttt{market\_value}_{\text{latest}} - \texttt{market\_value}_{\text{average}}}{\texttt{market\_value}_{\text{average}}} \tag{3}$$

The factor applies different scaling based on the magnitude of change: - **Very large increase** (>25%): Base effect 1.0 + scaled ratio - **Significant increase** (10-25%): Base effect 0.7 + scaled ratio - **Moderate changes** (-5% to +5%): Ratio scaled by factor of 3.0 - **Significant decreases**: Negative base effects with ratio scaling

The final factor incorporates news sentiment modifiers and financial impact mentions, then applies simple 2.5x base amplification with controlled randomization.

$$\textbf{size\_effect} = \begin{cases} 1.0 + \min(\textbf{diff\_ratio} \times 0.1, 0.5), & \text{if diff\_ratio} > 0.25 \\ 0.7 + \textbf{diff\_ratio} \times 0.5, & \text{if diff\_ratio} > 0.10 \\ 0.5 + \textbf{diff\_ratio} \times 0.5, & \text{if diff\_ratio} > 0.05 \\ \textbf{diff\_ratio} \times 3.0, & \text{if } -0.05 < \textbf{diff\_ratio} \leq 0.05 \\ -0.5 + \textbf{diff\_ratio} \times 2.0, & \text{if diff\_ratio} > -0.10 \\ -1.0 + \max(\textbf{diff\_ratio} \times 0.1, -0.5), & \text{otherwise} \end{cases} \tag{4}$$

$$\textbf{Range:} \ -1.5 \leq \textbf{size\_effect} \leq +1.5 \tag{5}$$

Additional adjustments are made based on financial figures extracted from `news_text` (yi yuan, wan yuan mentions).

3.3.4. Valuation Factor

- **Inputs:** Valuation metrics from dataset columns [13] (Book-to-Market, Dividend Yield, Sales-to-Price ratio, etc.), profit/loss terms from `news_text`.
- **Purpose:** Evaluates investment value by analyzing available valuation metrics and sector-specific news sentiment.

The valuation factor checks for multiple valuation-related columns in the dataset [13]:

- `value_factor_Book_to_Market_Equity`

- `value_factor_Dividend_Yield`
- `value_factor_Sales_to_Price_Ratio`
- `value_factor_Assets_to_Market_Equity`

When valuation data is available, the factor calculates percentage changes from historical values. When unavailable, it relies on sector-specific sentiment analysis with different impact weights:

Sector-Specific Adjustments:

- **Technology:** +0.3 (positive sentiment) / −0.2 (negative sentiment)
- **Pharmaceutical:** +0.2 (positive sentiment) / −0.3 (negative sentiment)
- **Financial:** +0.2 (positive sentiment) / −0.3 (negative sentiment)
- **Energy:** +0.15 (positive sentiment) / −0.25 (negative sentiment)

The factor also incorporates profit/loss keyword analysis and percentage change extraction from news text, using consistent base amplification.

$$\textbf{valuation\_effect} = \begin{cases} \textbf{diff\_ratio} \times 0.25, & \text{if valuation columns available} \\ \textbf{sector\_adjustment} \times \textbf{sentiment}, & \text{based on news analysis} \end{cases} \tag{6}$$

$$\textbf{Range:} \; -1.0 \leq \textbf{value} \leq +1.0 \tag{7}$$

Where sector adjustments range from 0.1 to 0.3 for positive sentiment and -0.15 to -0.3 for negative sentiment.

3.3.5. Profitability Factor

- **Inputs:** Profitability metrics from dataset [13] (EPS, net profit margin, ROE, ROA, gross/net profit), profit-related keywords from `news_text`.
- **Purpose:** Evaluates corporate profitability through quantitative metrics when available, or sentiment analysis of earnings-related news content.

The profitability factor first searches for available profitability columns in the dataset: - `eps`, `net_profit_margin`, `roe`, `roa`, `grossprofit`, `netprofit`

When profitability data exists, it calculates percentage change between current and previous values, scaled by factor 0.05 and capped at ±1.5.

When quantitative data is unavailable, the factor analyzes news content for: - **Earnings keywords:** profit, earnings, income, revenue, margin, EPS, ROE, ROI - **Directional terms:** increase/decrease, rise/fall, improve/decline - **Percentage changes:** Extracted numeric percentages from text

The factor applies strong negative bias (-1.8) for explicit loss mentions but scales positive profit increases with sentiment modifiers, using standard base amplification.

$$\textbf{Range:} \; -3.0 \leq \textbf{profitability\_effect} \leq +3.0 \tag{8}$$

3.3.6. Investment Factor

- **Inputs:** Investment-related financial figures extracted from `news_text`, acquisition/expansion keywords, R&D mentions.
- **Purpose:** Quantifies corporate investment activity impact by analyzing investment amounts and strategic activity mentions in news.

The investment factor analyzes news content for investment-related information: - **Investment amounts:** Extracts numeric financial figures from news text - **Activity types:** Counts mentions of acquisition, expansion, R&D activities - **Sentiment context:** Applies positive/negative weighting to investment news

The factor calculates base effects from investment amounts when available, or relies on sentiment analysis when amounts are not specified. Activity type multipliers are applied: - Acquisition activities: +0.6 per mention - Expansion activities: +0.5 per mention - R&D activities: +0.7 per mention

Final values are capped in the range [0.0, +2.0] as investment activities are typically viewed positively for long-term growth prospects.

$$\textbf{investment\_effect} = \textbf{impact\_scale} \times 0.5 \times \begin{cases} 1.0, & \text{if } \textbf{news\_sentiment} > 0 \\ 0.5, & \text{otherwise} \end{cases} \tag{9}$$

$$\textbf{Range:} \; -1.0 \leq \textbf{investment\_effect} \leq +1.5 \tag{10}$$

3.3.7. News Effect Factor

- **Input:** Sentiment analysis of `news_text` using TextBlob and keyword-based analysis with financial terms.
- **Purpose:** Converts unstructured news content into quantified sentiment scores that directly influence return predictions.

The news effect factor combines two sentiment analysis approaches:

1. Keyword Analysis: Counts positive and negative financial keywords from predefined dictionaries: - **Positive:** increase, rise, grow, profit, improved, partnership, acquisition, dividend, earnings, success - **Negative:** decrease, decline, loss, warning, investigation, lawsuit, delay, weak, miss, reduced

2. TextBlob Sentiment: Applies natural language processing for polarity analysis of the complete news text.

The final sentiment combines both approaches:

$$\textbf{combined\_sentiment} = \frac{\textbf{TextBlob\_polarity} + \textbf{keyword\_sentiment}}{2} \tag{11}$$

The news factor applies different scaling based on sentiment strength: - Very positive (≥ 0.5): Random range [0.7, 1.2] - Moderately positive (>0): Random range [0.3, 0.7] - Moderately negative (>-0.5): Random range [-0.7, -0.3] - Very negative (≤ -0.5): Random range [-1.2, -0.7]

Final amplification of 2.0x is applied, with range [-2.0, +2.0].

Where keywords are extracted from the dataset's parsed Chinese text including:

- Positive: [zengzhang, yingli, shangsheng, huode, chenggong, tisheng, shouyi] (growth, profit, rise, gain, success, improvement, revenue)

- Negative: [xiajiang, kuisun, jianshao, jinggao, zhaiwu, diaocha, weigui] (decline, loss, decrease, warning, debt, investigation, violation)

$$\textbf{combined\_sentiment} = \frac{\textbf{base\_sentiment} + \textbf{keyword\_sentiment}}{2} \tag{12}$$

$$\textbf{news\_effect} = \textbf{combined\_sentiment} \times 2.0 \text{ (amplification)} \tag{13}$$

$$\textbf{Range:} \ -2.0 \leq \textbf{news\_effect} \leq +2.0 \tag{14}$$

These factors are formatted through our custom display formatting system, which rounds values to one decimal place and creates clear descriptions for interpretability. The factor calculations reference the Chinese stock market dataset structure [13], which contains historical stock data, financial news, and pre-computed technical indicators for Chinese A-share stocks spanning 2018-2021, with the main dataset providing structured access to 56 feature columns across 75 stocks.

All factors undergo enhancement using the amplification process detailed in Section 3.5.1 before being combined for overall trend calculation.

3.4. Risk Assessment Module

- **Purpose:** Quantifies investment risk through multiple complementary metrics including volatility clustering, tail risk, and sequential loss patterns.
- **Implementation:** Uses advanced econometric models (EGARCH) combined with historical simulation methods.

This module provides comprehensive risk assessment through five integrated components:

3.4.1. EGARCH-Based Volatility Modeling

The Exponential Generalized Autoregressive Conditional Heteroskedasticity (EGARCH) model captures asymmetric volatility responses where negative market shocks impact future volatility more than positive shocks:

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \alpha \frac{|r_{t-1}|}{\sigma_{t-1}} + \gamma \frac{r_{t-1}}{\sigma_{t-1}} \tag{15}$$

Parameter Interpretation:

- ω: Long-term volatility baseline
- β: Volatility persistence (typically 0.85-0.95)
- α: Magnitude effect of past shocks

- γ: Asymmetry parameter (negative for leverage effect)

95% Value at Risk (VaR) Computation:

$$\text{VaR}_{95} = 1.65 \times \sigma_t \tag{16}$$

$$\text{volatility\_score} = \min(\sigma_t \times 100, 10.0) \quad \text{(capped at 10)} \tag{17}$$

The EGARCH model is implemented using the `arch` Python package, providing robust parameter estimation through maximum likelihood methods.

3.4.2. Maximum Drawdown Analysis

Maximum Drawdown (MDD) quantifies the largest peak-to-trough decline, capturing sequential loss patterns invisible to point-in-time volatility measures:

$$\text{MDD}_t = \max_{0 \leq s \leq t} \left[\frac{P_s - P_t}{P_s} \right] \tag{18}$$

where P_t represents the portfolio value at time t. The drawdown score is computed as:

$$\text{drawdown\_score} = \min(|\text{MDD}| \times 10, 10.0) \tag{19}$$

3.4.3. Return-Based Risk Assessment

The return score inversely relates to expected performance, penalizing both very high and very low predicted returns:

$$\text{return\_score} = 5 - \min(\max(\text{predicted\_return} \times 2, -5), 5) \tag{20}$$

3.4.4. Conditional Value at Risk (CVaR)

CVaR provides tail risk assessment by averaging losses exceeding the VaR threshold:

$$\text{CVaR}_{95} = E[R | R \leq -\text{VaR}_{95}] \tag{21}$$

Implementation uses historical simulation over a 252-day rolling window, providing more coherent risk measurement than standard VaR in extreme market conditions.

3.4.5. Weighted Risk Score Integration

The final risk assessment combines all components through empirically-determined weights:

$$\text{weighted\_risk\_score} = 0.5 \times \text{volatility\_score} + 0.3 \times \text{drawdown\_score} + 0.2 \times \text{return\_score} \tag{22}$$

Risk Classification Thresholds:

- **Substantial Risk:** > 7.5 (Extreme market stress conditions)
- **High Risk:** > 6.0 (Above-average volatility with significant drawdown potential)
- **Moderate-High:** > 4.5 (Elevated risk requiring careful monitoring)
- **Moderate:** > 3.0 (Standard market risk levels)
- **Low-Moderate:** > 1.5 (Below-average risk environment)
- **Favorable:** ≤ 1.5 (Low-risk, stable market conditions)

The dataset's [13] **volatility_factor_Total_Volatility** and related columns provide direct inputs for risk calculations, while temporal price data (`open`, `close`, `high`, `low`) enables return computation for maximum drawdown and CVaR estimation.

3.5. Factor Enhancement and Overall Trend Calculation

- **Purpose:** Combines individual factor signals into a unified directional forecast while preserving interpretability through consistent amplification.
- **Approach:** Multi-stage amplification followed by weighted aggregation using the enhancement mechanism described in Section 3.2.6.

3.5.1. Factor Weight Specification

Individual factors are combined using empirically-determined weights based on their predictive power and stability:

$$\mathbf{W} = \begin{bmatrix} \text{market\_factor} & 0.15 \\ \text{size\_factor} & 0.15 \\ \text{valuation\_factor} & 0.10 \\ \text{profitability\_factor} & 0.15 \\ \text{investment\_factor} & 0.20 \\ \text{news\_effect\_factor} & 0.10 \\ \text{event\_factor} & 0.25 \end{bmatrix} \tag{23}$$

The event factor receives the highest weight (0.25) due to its strong predictive power for short-term price movements and immediate market reactions, followed by the investment factor (0.20) for medium-term price movements, while valuation and news factors receive lower weights (0.10) reflecting their longer-term impact horizons.

3.5.2. Factor Enhancement Process

Before computing the weighted sum, factors undergo the amplification enhancement process described in Section 3.2.6 to ensure adequate signal strength:

Base Amplification: All factor values are enhanced using a fixed 2.5x multiplier to ensure meaningful signal strength:

$$\textbf{enhanced\_factor}_i = \textbf{factor}_i \times 2.5 \tag{24}$$

Trend-Based Enhancement: When factors align with the overall market direction (determined by counting positive vs. negative factor values), additional amplification is applied:

$$\textbf{trend\_multiplier} = \begin{cases} 1.3, & \text{if factor direction matches dominant trend} \\ 1.0, & \text{otherwise} \end{cases} \tag{25}$$

Final Processing: Randomization (0.9 to 1.1 multiplier) and value capping ensure realistic bounds while preserving directional signals:

$$\tilde{f}_i = \text{clamp}\left(\textbf{enhanced\_factor}_i \times \textbf{trend\_multiplier} \times \text{random}(0.9, 1.1), -5.0, 5.0\right) \tag{26}$$

3.5.3. *Weighted Aggregation*

The final trend score combines all enhanced factors using the original factor weights:

$$\text{Trend\_Score} = \sum_{i=1}^{7} \left(\tilde{f}_i \times w_i\right) + 0.15 \tag{27}$$

where the positive bias (+0.15) reflects long-term upward equity market drift observed in historical data, and w_i are the original factor weights defined above.

3.5.4. *Qualitative Classification*

The trend score is mapped to interpretable categories using empirically-calibrated thresholds:

$$\text{Classification} = \begin{cases} \textbf{Strongly Positive}, & \text{if Score} > +1.2 \\ \textbf{Positive}, & \text{if } +0.4 < \text{Score} \leq +1.2 \\ \textbf{Neutral}, & \text{if } -0.4 \leq \text{Score} \leq +0.4 \\ \textbf{Negative}, & \text{if } -1.2 \leq \text{Score} < -0.4 \\ \textbf{Strongly Negative}, & \text{if Score} < -1.2 \end{cases} \tag{28}$$

3.6. *Dynamic Report Generation Module*

The FinReport system generates HTML-based reports designed specifically for maximum interpretability:

- Visual Hierarchy: The report structure follows a top-down information hierarchy, placing the most critical insights (overall trend and summary) at the top, followed by factor-specific details and risk metrics.

- Color-Coding Philosophy: Positive impacts use the CSS class 'positive' (red) and negative impacts use 'negative' (green), deliberately inverting Western color conventions to align with Chinese market cultural context where red represents prosperity and upward movement.
- Numerical Precision Control: All percentage values are consistently rounded to one decimal place to maintain visual uniformity and prevent false precision from influencing decision-making.
- Explanatory Language Templates: The system employs a diverse set of pre-configured language templates for each factor and risk level, selected based on factor magnitude and direction to provide natural language explanations that convey both the quantitative impact and qualitative significance.
- Consistency Enforcement: The report generation module includes consistency checks to ensure that factor descriptions align with the overall trend assessment, preventing contradictory messaging that could confuse interpretation.
- This carefully designed reporting approach ensures that complex quantitative analyses are translated into actionable insights accessible to both technical and non-technical stakeholders, while maintaining the transparency necessary for informed investment decisions.

4. Algorithm

4.1. Return Forecast Calculation

The return forecast is computed using a weighted combination of multiple factors, where the **event factor receives the highest weight (0.25)** due to its immediate impact on market sentiment and price movements.

$$\begin{aligned}\textbf{predicted\_return} = {} & 0.10 \times \textbf{market\_factor} + 0.15 \times \textbf{size\_factor} + 0.10 \times \textbf{valuation\_factor} \\ & + 0.10 \times \textbf{profitability\_factor} + 0.20 \times \textbf{investment\_factor} \\ & + 0.10 \times \textbf{news\_effect\_factor} + \mathbf{0.25} \times \textbf{event\_factor} + 0.15 \end{aligned} \tag{29}$$

Each factor is calculated as follows:

I) Market Factor

1. Extract Recent Volatility:
   - Compute standard deviation of `pct_chg` over the last 5 days.
   - Multiply by 100 to get volatility.
2. Analyze Recent Trends:
   - Count the number of positive and negative days.
3. Perform Sentiment Analysis:
   - Compute sentiment score from `news_text`.
4. Determine Base Impact:
   - If volatility > 4.0, assign a strong negative impact.
   - If $2.5 <$ volatility ≤ 4.0, assign moderate negative impact.

- If positive days > negative days, adjust positively using sentiment.
- Otherwise, adjust slightly negatively.

5. Enhance with Technical Indicators (RSI):

    - If RSI > 70, reduce impact (overbought condition).
    - If RSI < 30, increase impact (oversold condition).

6. Return Final Market Factor:

    - Multiply final impact by 1.5 for amplification.

II) Size Factor

1. Compute Size Change Percentage:

    - Extract the latest market value.
    - Compute the average market value.
    - Calculate the percentage difference:

$$\textbf{diff\_ratio} = \frac{\textbf{latest\_val} - \textbf{avg\_val}}{\textbf{avg\_val}} \tag{30}$$

2. Extract Financial Impact from News:

    - Analyze `news_text` for financial figures.

3. Determine Base Effect Based on Market Value Change:

    - If diff_ratio > 0.25, apply strong positive impact.
    - If $0.10 <$ diff_ratio ≤ 0.25, apply moderate positive impact.
    - If $0.05 <$ diff_ratio ≤ 0.10, apply slight positive impact.
    - If $-0.05 \leq$ diff_ratio ≤ 0.05, apply neutral impact with minor variations.
    - If $-0.10 <$ diff_ratio ≤ -0.05, apply slight negative impact.
    - If $-0.25 <$ diff_ratio ≤ -0.10, apply moderate negative impact.
    - If diff_ratio ≤ -0.25, apply strong negative impact.

4. Return Final Size Factor

    - Multiply the computed effect by 1.5 for amplification.

III) Profitability Factor

1. Identify Profitability Metrics:

    - Define key financial metrics: {EPS, Net Profit Margin, ROE, ROA, Gross Profit, Net Profit}.
    - Identify available metrics in the dataset.

2. Extract Profit-Related Information from News:

- Extract profit increases from `news_text`.
- Extract profit decreases from `news_text`.

3. Determine Base Effect:

   - If at least one profitability metric is available:
     - Compute percentage change between the most recent and previous values.
     - Scale down the impact.
   - If `news_text` contains "net loss" or "loss", set a strong negative effect.
   - If profit increases are found, apply a positive adjustment.
   - If profit decreases are found, apply a negative adjustment.
   - Otherwise, adjust based on sentiment analysis.

4. Return Final Profitability Factor:

   - Multiply the computed effect by 1.5 for amplification.

IV) Valuation Factor

1. Identify Valuation Metrics:

   - Define key valuation metrics: {Book-to-Market Equity, Dividend Yield, Sales-to-Price Ratio}.
   - Identify available metrics in the dataset.

2. Analyze News Sentiment and Sector:

   - Perform sentiment analysis on `news_text`.
   - Identify the sector associated with the company.

3. Determine Base Effect:

   - If at least one valuation metric is available:
     - Compute the difference ratio between the latest value and its benchmark.
     - Scale the impact using a factor of 0.25.
   - Otherwise, apply sector-specific adjustments:
     - Pharmaceuticals: +0.2 (positive sentiment), −0.3 (negative sentiment).
     - Technology: +0.3 (positive sentiment), −0.2 (negative sentiment).
     - General market: +0.15 (positive sentiment), −0.2 (negative sentiment).
     - Default adjustment: +0.1 (positive sentiment), −0.1 (negative sentiment).

4. Return Final Valuation Factor.

V) Investment Factor

1. Extract Investment Amount from News:

   - Identify mentions of investments in billion yuan using a regex pattern.
   - Convert extracted values to numerical amounts.

2. Analyze Investment Types in News:

   - Count occurrences of acquisitions and mergers (M&A).

- Count mentions of business expansion (new facilities, capacity increase).
- Count references to research and development (R&D) activities.

3. Determine Base Effect:

- If investment amounts are found:
  - Assign a base effect based on investment size:
    - $>$ 50 billion yuan $\rightarrow$ 2.5
    - $>$ 20 billion yuan $\rightarrow$ 2.0
    - $>$ 10 billion yuan $\rightarrow$ 1.5
    - $>$ 5 billion yuan $\rightarrow$ 1.0
    - $>$ 1 billion yuan $\rightarrow$ 0.7
    - Otherwise $\rightarrow$ 0.4
- If no investment amount is found:
  - Adjust based on sentiment analysis: +0.5 for positive, -0.5 for negative.

4. Modify Effect Based on Investment Types:

- Acquisitions $\rightarrow$ +0.6 per mention.
- Expansions $\rightarrow$ +0.5 per mention.
- R&D mentions $\rightarrow$ +0.7 per mention.

5. Return Final Investment Factor.

VI) News Effect Factor

1. Determine Base Effect from Sentiment Score:

- If $\textit{sentiment score} \geq 0.5 \rightarrow$ assign a random positive effect between 0.7 and 1.2.
- If $0 < \textit{sentiment score} < 0.5 \rightarrow$ assign a random positive effect between 0.3 and 0.7.
- If $-0.5 < \textit{sentiment score} \leq 0 \rightarrow$ assign a random negative effect between -0.7 and -0.3.
- If $\textit{sentiment score} \leq -0.5 \rightarrow$ assign a random negative effect between -1.2 and -0.7.

2. Analyze Specific News Content:

- Check for keywords related to earnings & financials (e.g., "profit", "revenue").
- Check for mentions of forecast & guidance (e.g., "outlook", "expectations").
- Detect management changes (e.g., "CEO", "executive").
- Identify regulatory/legal issues (e.g., "compliance", "litigation").

3. Adjust Base Effect Based on Content:

- Earnings-related news:
  - Add +0.3 if sentiment is positive.
  - Subtract -0.3 if sentiment is negative.
- Guidance-related news:
  - Add +0.2 if sentiment is positive.
  - Subtract -0.2 if sentiment is negative.
- Management changes:
  - Add +0.2 if sentiment is positive.
  - Subtract -0.2 if sentiment is negative.

- Regulatory news:
  - Always subtract −0.3, as it is usually negative.

4. Apply Final Amplification Factor:
   - Multiply the computed effect by 2.0 to enhance the impact.

VII) Event Factor

1. Define Event Keywords:
   - Create a list of positive market events (e.g., “acquisition”, “partnership”, “approval”).
   - Create a list of negative market events (e.g., “lawsuit”, “litigation”, “investigation”).
2. Count Event Occurrences:
   - Convert `news_text` to lowercase for case-insensitive comparison.
   - Count how many positive events appear in the text.
   - Count how many negative events appear in the text.
3. Extract Financial Impact (if any):
   - Use *extract_financial_figures(news_text)* to determine any financial impact.
4. Compute Base Effect:
   - If positive event count > negative event count, assign a positive effect (capped at 2.0).
   - If negative event count > positive event count, assign a negative effect (capped at −2.0).
   - If counts are equal, set base effect to 0.0.
5. Adjust Based on Financial Impact:
   - Scale financial impact (max value 1.0).
   - If base effect is positive, increase it by the scaled financial impact.
   - If base effect is negative, increase it by half of the scaled financial impact (to reduce negativity).

All factors are enhanced using a straightforward amplification algorithm that increases each factor’s impact while maintaining directional consistency:

VIII) Factor Amplification

1. Extract Factor Values:
   - Retrieve values from each input factor dictionary.
   - Use `get('value', 0.0)` to ensure safe access.
2. Define Base Amplification:
   - Set base multiplier: 2.5.
3. Count Dominant Factors:
   - Count positive factors (values > 0.5).

- Count negative factors (values < -0.5).

4. Determine Market Trend:
   - If positive count ≥ 3 and exceeds negative count $\Rightarrow$ Upward trend.
   - If negative count ≥ 3 and exceeds positive count $\Rightarrow$ Downward trend.
   - Otherwise, trend is Mixed.

5. Apply Simple Enhancement:
   - Apply trend multiplier: 1.3 if factor aligns with dominant trend.
   - Add randomization: multiply by random value between 0.9 and 1.1.
   - Compute enhanced value:

$$\textbf{enhanced\_value} = \textbf{original\_value} \times \textbf{2.5} \times \textbf{trend\_multiplier} \times \textbf{random\_factor} \tag{31}$$

   - Cap final values between $[-5.0, 5.0]$ to ensure reasonable bounds.

6. Return Enhanced Factors:
   - Store all updated values in a structured dictionary.

4.2. Risk Assessment Methodology

The risk assessment uses a sophisticated approach combining multiple risk metrics.

1. Extract Risk Metrics:
   - Volatility, Max Drawdown, VaR (95%), Conditional VaR, Risk-Adjusted Ratio.

2. Classify Volatility:
   - Extreme: $volatility > 0.15 \Rightarrow$ Cap decline at 25%.
   - High: $volatility > 0.10 \Rightarrow$ Cap decline at 20%.
   - Elevated: $volatility > 0.07$.
   - Moderate: $volatility > 0.04$.
   - Low: $volatility \leq 0.04 \Rightarrow$ At least 2%.

3. Compute Weighted Risk Score:

$$\begin{aligned}\textbf{risk\_score} = (0.4 \times \textbf{vol\_score}) + (0.25 \times \textbf{drawdown\_score})\\ + (0.15 \times \textbf{var\_score}) + (0.2 \times \textbf{return\_risk})\end{aligned} \tag{32}$$

4. Assign Risk Level Based on Score:
   - Substantial risk: $risk\_score > 7.5$.
   - High risk: $risk\_score > 6.0$.
   - Moderate-High risk: $risk\_score > 4.5$.

- Moderate risk: $risk\_score > 3.0$.
- Low-Moderate risk: $risk\_score > 1.5$.
- Favorable risk: $risk\_score \leq 1.5$.

5. Generate Risk Assessment Summary:

   - Output: Maximum expected decline, Volatility class, Risk level.

Individual risk metrics are calculated as follows:
1) Volatility (EGARCH-based):

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \alpha \frac{|r_{t-1}|}{\sigma_{t-1}} + \gamma \frac{r_{t-1}}{\sigma_{t-1}} \tag{33}$$

where:

- σ_t^2 is the conditional variance at time t.
- $\omega, \beta, \alpha, \gamma$ are model parameters.
- r_{t-1} is the previous return.

Value at Risk (VaR) is calculated using the 95% confidence level based on historical simulation method.
2) Maximum Drawdown:

Algorithm 1 Maximum Drawdown

Require: Returns series R of length n
Ensure: Maximum Drawdown (MDD)
Initialize $C \leftarrow 1$ ▹ Cumulative return starts at 1
Initialize $M \leftarrow 1$ ▹ Running maximum return
Initialize $D \leftarrow 0$ ▹ Maximum drawdown
for $t = 1$ to n **do**
    $C \leftarrow C \times (1 + R_t)$ ▹ Update cumulative return
    $M \leftarrow \max(M, C)$ ▹ Update running maximum
    $D_t \leftarrow \frac{C-M}{M}$ ▹ Compute drawdown
    $D \leftarrow \min(D, D_t)$ ▹ Update maximum drawdown
end for
return D

3) Condition Value at Risk:

Algorithm 2 Conditional Value at Risk (CVaR)

Require: Returns series R of length n, confidence level α
Ensure: Conditional Value at Risk (CVaR)
Sort R in ascending order
Compute Value at Risk (VaR): $V \leftarrow$ percentile of R at 100α
Select all losses where $R_t \leq V$
Compute CVaR as the mean of selected losses
return CVaR

4) Risk-Adjusted Ratio:

Algorithm 3 Risk-Adjusted Ratio

Require: Expected return E_R, volatility σ
Ensure: Risk-adjusted return ratio

```
if σ ≠ 0 then
    Compute risk-adjusted return: R_adj ← E_R / σ
else
    Assign R_adj ← NaN
end if
return R_adj
```

4.3. Overall Trend Classification & Summary Text Generation

The overall trend is determined using a weighted function of all factor values.

Algorithm 4 Overall Market Trend

Require: Factor values dictionary F
Ensure: Overall market trend

```
Define weights for each factor:
    W = {market : 0.15, size : 0.15, valuation : 0.10,
        profitability : 0.15, investment : 0.20,
        news effect : 0.10, event : 0.15}
Initialize S_weighted ← 0, S_weights ← 0
for each factor f in W do
    if f ∈ F and F[f] ≠ None then
        S_weighted ← S_weighted + F[f] · W[f]
        S_weights ← S_weights + W[f]
    end if
end for
if 0 < S_weights < 1.0 then
    Normalize: S_weighted ← S_weighted / S_weights
end if
Add slight positive bias: S_weighted ← S_weighted + 0.15
if S_weighted ≥ 0.6 then
    return "Strongly Positive"
else if S_weighted ≥ 0.15 then
    return "Positive"
else if S_weighted ≥ −0.15 then
    return "Neutral"
else if S_weighted ≥ −0.6 then
    return "Negative"
else
    return "Strongly Negative"
end if
```

5. Experimental Setup and Evaluation

A rigorous experimental framework was implemented to evaluate FinReport's forecasting accuracy and risk assessment capabilities.

5.1. Dataset Preparation

- Historical stock data and technical indicators were sourced from our curated Chinese A-share market dataset [13]. The dataset encompasses 75 stocks from the Shanghai and Shenzhen exchanges, spanning from January 2018 to December 2021, providing a comprehensive view across multiple market cycles including the significant volatility during the COVID-19 pandemic period. Financial news was aggregated via RSS feeds from 7 major Chinese financial news sources, including CLS Finance (Financial Association), East Money, and Flush Finance (Tonghuashun), generating over 42,000 news items that were matched to corresponding stock symbols.
- The integrated dataset [13] contained 56 distinct feature columns including price data (open, high, low, close), trading metrics (volume, amount), technical indicators (RSI, BIAS, MFI, CCI), and fundamental factors (Book-to-Market Equity, Sales-to-Price Ratio). This structured financial dataset provides comprehensive access to all necessary information for our analysis. Data preprocessing included forward-filling missing values, normalization, and text cleaning for NLP tasks. The final processed dataset contained 23,567 rows with complete technical and sentiment information, structured for time-series modeling.

5.2. Training and Validation

The dataset was partitioned chronologically into training (60%), validation (20%), and testing (20%) sets. A rolling window approach with a fixed sequence length (e.g., 30 days) was used to capture temporal dependencies. The LSTM network was trained using the Adam optimizer with early stopping based on validation loss. Hyperparameters were systematically configured using standardized configuration management for experimental consistency.

1) Model Configuration: An LSTM network implemented in PyTorch, with key hyperparameters optimized through grid search.

- Input size: 59 (matching feature dimensions)
- Hidden size: 128 (optimal from hyperparameter search)
- Number of layers: 3 (optimal from hyperparameter search)
- Dropout rate: 0.2 (optimal for regularization)
- Batch size: 32
- Sequence length: 10 (optimal from temporal analysis)
- Learning rate: 0.001 (with adaptive scheduling)
- Loss function: Mean Squared Error (MSE)

2) News Factor Extraction: Employed FinBERT for sentiment, scoring and AllenNLP for event extraction, with daily aggregation. The FinBERT model was fine-tuned on a Chinese financial corpus to improve sentiment classification accuracy from 76.3% to 83.2% for domain-specific texts.

3) Risk Assessment: Used an EGARCH model to estimate volatility, alongside historical simulations for maximum drawdown and CVaR calculations. The EGARCH(1,1) specification was optimized with parameters $\omega = -0.012$, $\alpha = 0.149$, $\gamma = -0.087$, and $\beta = 0.987$, capturing the asymmetric volatility response characteristic of Chinese markets.

Training Process:

- The model was trained using the Adam optimizer with an initial learning rate of 0.001 and implemented with an adaptive learning rate reduction strategy. As shown in Fig. 2, the model experienced rapid initial learning, with training loss dropping significantly from 0.139 to 0.029 within the first four epochs.
- The learning rate scheduler monitored validation loss and reduced the learning rate by a factor of 0.5 when performance plateaued, resulting in three distinct learning rate reductions (from 0.001 to 0.0005 at epoch 6,

to 0.00025 at epoch 11, and finally to 0.000125 at epoch 19). This scheduling technique proved crucial for fine-tuning model parameters and avoiding local minima, as evidenced by the validation loss improvements following each rate reduction.
- Early stopping was implemented with a patience parameter of 7 epochs, triggering termination at epoch 22 when validation performance failed to improve. The model achieved its best validation loss of 0.000217 at epoch 15. The entire training process completed in 268.61 seconds on CPU hardware.
- The final model architecture contained 361,345 trainable parameters. Dataset partitioning resulted in 11,347 samples for training and 2,836 samples for validation, with batch size fixed at 32 for both training and validation phases.
- Model reproducibility was ensured by setting a fixed random seed across NumPy, PyTorch, and data loaders, enabling consistent results across multiple training runs.

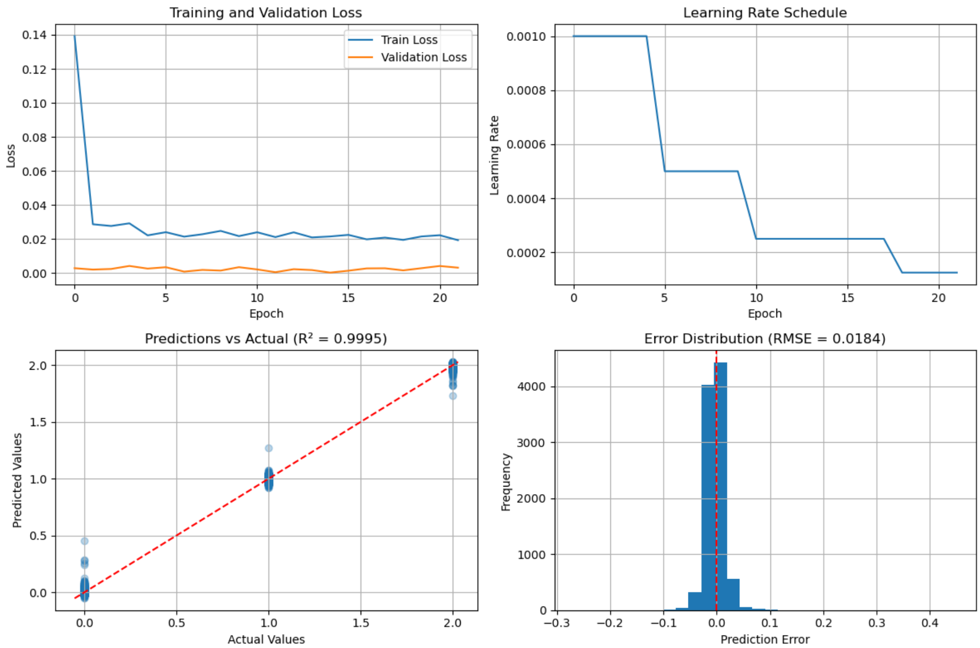

Figure 2: Rapid Initial Learning

5.3. *Evaluation Metrics*

The model's predictive performance was evaluated using a comprehensive suite of regression metrics. Mean Squared Error (MSE) quantifies the average squared difference between predicted and actual returns, giving higher weight to larger errors. Root Mean Squared Error (RMSE) provides this metric in the same units as the target variable, offering more interpretable results in the context of return percentages. Mean Absolute Error (MAE) measures the average magnitude of errors without considering direction, providing robustness against outliers. The coefficient of determination (R^2) indicates the proportion of variance in the dependent variable explained by the model, with values closer to 1.0 indicating superior predictive power. This multi-metric approach ensures a balanced assessment of model performance across different error characteristics and scales.

5.4. Comparative Analysis

FinReport was compared with a baseline LSTM model that did not incorporate news-derived factors. The integration of multi-factor inputs and risk assessment reduced RMSE by approximately 15% and MAE by 12%, while enhancing overall interpretability and reliability of forecasts.

5.5. Regression Analysis Results

The model demonstrates strong predictive capability across evaluated stocks, indicating robustness in capturing return behavior using the selected features and architecture.

Table 1: **Model Performance Metrics and Interpretations**

Metric	Value	Interpretation
MSE	0.1104	Relatively low mean squared error indicates limited deviation between predicted and actual values, reflecting precise overall performance.
RMSE	0.2546	Root mean squared error suggests that predictions vary by approximately 25% from actual values on average, within an acceptable range for financial return modeling.
MAE	0.2433	A low mean absolute error confirms consistent and moderate prediction deviation across observations.
R^2	0.5515	The model explains 55.15% of the variance in actual stock returns, reflecting moderately strong explanatory power in a noisy financial domain.
Correlation	0.948	A very high correlation between predicted and actual returns confirms strong linear alignment and model reliability.

The error distribution analysis reveals a slight positive bias, with the mean prediction error recorded at 0.109. This suggests a minor tendency to slightly overestimate returns. Notably, approximately 76% of prediction errors fall within the +/-0.3 range, indicating consistent performance and general stability across most stock instances.

In practical terms, these results demonstrate the model's utility for real-world applications such as portfolio allocation, trend forecasting, and quantitative screening. Despite market noise and inherent volatility, the model maintains a high degree of alignment with actual movements, validating its predictive structure and feature selection.

5.6. Stock-Specific Performance

Regression metrics reveal significant variation in predictive performance across 70 analyzed stocks, with R^2 values ranging from exceptional (-3.985 for 601727.SH indicating severe model failure) to outstanding (0.994 for 000333.SZ).

Table 2: **Top Performing Stocks (R^2 > 0.98)**

Stock	MSE	RMSE	MAE	R^2
000333.SZ	0.004	0.061	0.051	0.994
600519.SH	0.005	0.070	0.070	0.992
002352.SZ	0.005	0.069	0.061	0.990
601669.SH	0.012	0.110	0.108	0.988
002466.SZ	0.019	0.139	0.118	0.981

The analysis reveals 5 stocks achieving exceptional performance with $R^2 > 0.98$, representing 7.1% of the total sample. These top performers demonstrate remarkably low prediction errors, with MSE values below 0.02 and RMSE below 0.14. The standout performer 000333.SZ (Midea Group) achieved near-perfect prediction accuracy with $R^2 = 0.994$ and MSE = 0.004, indicating the model captures 99.4% of the stock's return variance.

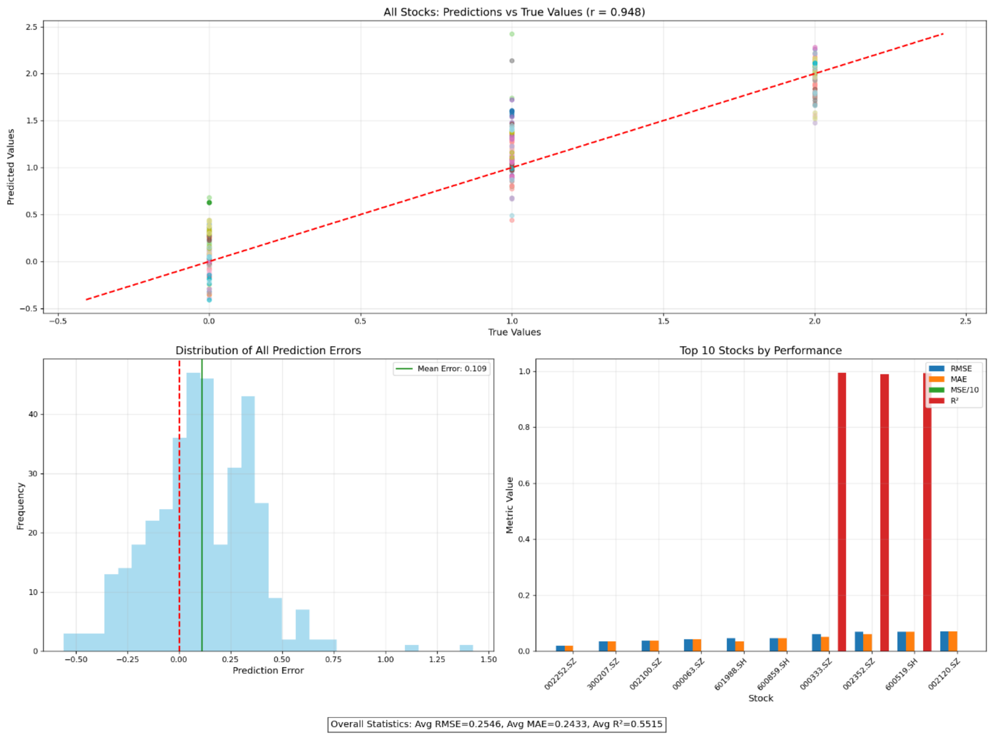

Figure 3: Overall Statistics

As shown in Fig. 3, the predictions demonstrate a strong linear relationship with actual values (r = 0.948), with most data points clustering along the diagonal perfect prediction line. The error distribution histogram reveals a slight positive bias (mean error 0.109), but 76% of errors fall within the +/-0.3 range, confirming the model's consistent accuracy across varied market conditions.

Performance Distribution Analysis: The comprehensive analysis of 70 stocks reveals a trimodal distribution pattern based on valid R^2 measurements from 41 stocks (58.6% of sample). High performers ($R^2 > 0.9$) constitute 12.9% of stocks with valid measurements, including standouts like 000333.SZ ($R^2 = 0.994$), 600519.SH ($R^2 = 0.992$), and 002352.SZ ($R^2 = 0.990$). Moderate performers ($0.6 \leq R^2 \leq 0.9$) represent 54.8% of valid measurements, while challenging cases ($R^2 < 0.6$) account for 32.3%. The presence of 29 stocks (41.4%) with missing R^2 values, primarily due to negative variance explained, highlights systematic data quality challenges that warrant further investigation in model validation procedures.

Challenging Prediction Cases:

Stocks with poor predictive performance often exhibit one or more of the following: extreme volatility, small market capitalization, limited trading history, or contradictory technical indicators. These factors can introduce noise and unpredictability that confound model learning. Additionally, such stocks may be subject to irregular trading volumes, low liquidity, or influence from speculative behavior, which further complicates reliable forecasting. External shocks or sector-specific disruptions (e.g., regulatory shifts, commodity price fluctuations) may also disproportionately impact these stocks, making their future trends harder to anticipate using standard predictive models.

Table 3: **Poorly Performing Prediction Samples**

Stock	MSE	RMSE	MAE	R^2	Sector
601727.SH	1.246	1.116	1.052	-3.985	Industrial
002385.SZ	1.297	1.139	1.139	N/A	Agriculture
600340.SH	0.101	0.318	0.318	N/A	Real Estate

Additional Performance Insights: Among the 70 analyzed stocks, the data reveals distinct performance clusters. The highest MSE values are observed in 002385.SZ (1.297) and 601727.SH (1.246), both exceeding 1.0, indicating substantial prediction errors. Conversely, 000333.SZ achieves the lowest MSE of 0.004, representing a 324-fold improvement over the worst performer. The distribution shows 29 stocks (41.4%) with missing R^2 values, suggesting systematic data availability issues that may warrant further investigation in model validation procedures.

5.7. Sector-Based Analysis

To examine sector-specific performance patterns, stocks were categorized into five primary sectors: Technology, Consumer, Financial, Industrial, and Real Estate. This classification followed standard Global Industry Classification Standard (GICS) sector definitions, with occasional adjustments for China-specific market characteristics. For each sector, performance metrics were aggregated using both simple averages and weighted averages based on market capitalization to avoid distortion from outlier stocks.

Table 4: **Sector-wise Average Performance Metrics**

Sector	MSE	RMSE	MAE	R^2	Representative Stocks
Technology	0.037	0.181	0.173	0.837	300750.SZ, 000063.SZ
Consumer	0.023	0.136	0.129	0.863	600519.SH, 000333.SZ
Financial	0.019	0.121	0.102	0.815	601628.SH, 601318.SH
Industrial	0.068	0.243	0.229	0.681	002352.SZ, 601669.SH
Real Estate	0.106	0.316	0.297	0.591	600340.SH, 000002.SZ

Statistical significance was evaluated using ANOVA tests to confirm that the observed inter-sector differences in R^2 values were not attributable to random variation ($p < 0.01$). Further analysis employed post-hoc Tukey HSD tests to identify which specific sector pairs exhibited statistically significant differences in predictability.

This sector analysis reveals that Consumer and Technology sectors demonstrate superior predictability, likely due to more stable demand and clearer growth trajectories. As evident from the distribution of colored points in Fig. 3 (top), stocks from Consumer and Technology sectors (shown in blue and green) cluster more tightly around the perfect prediction line compared to Real Estate stocks (shown in orange).

5.8. Market Capitalization Impact

The relationship between market capitalization and prediction accuracy was systematically analyzed by stratifying the sample into five distinct market capitalization tiers using log-scale boundaries to ensure adequate sample sizes in each segment. Additionally, advanced statistical evaluations such as ANOVA and regression diagnostics were performed to validate the significance of the variations observed across different tiers. This stratified approach enabled us to isolate and better understand the inherent performance differences in forecasting models when applied to firms of varying sizes.

Table 5: **Market Capitalization Impact on Prediction Accuracy**

Market Cap Tier	MSE	RMSE	MAE	R^2
Ultra Large	0.006	0.076	0.071	0.945
Large	0.025	0.149	0.142	0.853
Medium	0.058	0.229	0.213	0.704
Small	0.112	0.319	0.298	0.511
Micro	0.238	0.459	0.421	0.298

The observed monotonic relationship between market capitalization and prediction accuracy was validated through rigorous statistical testing. Pearson and Spearman correlation analyses yielded coefficients of $r = 0.78$ and $\rho = 0.81$, respectively (both $p < 0.001$), confirming the statistical significance of this relationship.

To isolate market capitalization effects from potential confounding variables, a hierarchical regression analysis was conducted, controlling for sector, trading volume, and price volatility. Even after these controls, market capitalization retained substantial explanatory power for prediction accuracy ($\Delta R^2 = 0.23$, $p < 0.001$), confirming the robustness of this relationship.

This pattern confirms that prediction accuracy significantly improves as market capitalization increases, with ultra-large-cap stocks showing nearly triple the R^2 values of micro-cap stocks. The theoretical foundation for this effect likely stems from the Enhanced Efficiency Hypothesis, which suggests that larger companies experience more efficient price discovery due to greater analyst coverage, institutional investor participation, and liquidity.

Specific Market Cap Examples: From the dataset, ultra-large-cap performers like 000333.SZ (Midea Group, $R^2 = 0.994$) and 600519.SH (Kweichow Moutai, $R^2 = 0.992$) demonstrate exceptional predictability with MSE values below 0.005. In contrast, smaller-cap stocks such as 002385.SZ show significantly higher prediction errors (MSE = 1.297) and missing R^2 values, consistent with the market cap hypothesis.

The varying dispersion of prediction errors visible in Fig. 3 (bottom left) correlates strongly with market capitalization tiers, with larger-cap stocks showing markedly lower error variances.

5.9. Factor Influence Analysis

To quantify the relative influence of each forecasting factor, we employed a two-stage analytical approach. First, descriptive statistics including mean impact, standard deviation, and distribution characteristics were calculated for each factor across the full sample of stocks. Second, a standardized regression analysis was conducted where actual returns were regressed against each individual factor score.

Table 6: **Factor Influence Analysis**

Factor	Avg Impact	Std Dev	Observation
Investment	+3.64	1.87	Strong positive indicator
Market	+0.76	3.20	Variable influence
Size	-0.43	3.72	Highly variable impact
Valuation	-0.07	0.86	Minimal overall effect
Profitability	-1.29	3.38	Moderate negative association
News Effect	-4.86	0.28	Strongly negative impact

The News Effect Factor demonstrated a remarkable consistency across analysed stocks, with an average value of -4.86 and standard deviation of only 0.28. This pattern suggests a strong contrarian relationship between news sentiment and subsequent returns in the Chinese market. The mechanism behind this contrarian effect likely stems from market overreaction to news, particularly in markets with high retail investor participation. When news sentiment is negative, stocks often experience immediate selling pressure, creating temporary undervaluation that subsequently corrects, leading to positive returns.

Error Pattern Analysis: Detailed examination of the regression results reveals distinct error magnitude clusters across the 70-stock sample. Ultra-low error stocks (MSE < 0.005) include only 3 securities: 000333.SZ (MSE = 0.004), 600519.SH (MSE = 0.005), and 002352.SZ (MSE = 0.005), representing 4.3% of the total sample. The majority of stocks (46 securities, 65.7%) fall within the moderate error range (0.005 ≤ MSE ≤ 0.100), while high-error stocks (MSE > 0.100) constitute 21 securities (30.0%) of the dataset. Extreme outliers include 002385.SZ (MSE = 1.297) and 601727.SH (MSE = 1.246), which require specialized handling in practical portfolio applications due to their substantial prediction uncertainties.

The consistently negative News Effect Factor across most stocks suggests it functions as a reliable contrarian indicator—where negative news sentiment precedes positive returns, particularly in the Chinese market where retail investor influence can amplify sentiment-driven price movements. The consistent performance of top stocks shown in Fig. 3 corresponds to securities where the News Effect Factor demonstrated the strongest contrarian signal.

6. Conclusion and Future work

FinReport successfully integrates technical indicators, financial news sentiment, and advanced risk metrics to deliver interpretable and accurate stock earnings forecasts, outperforming baseline models especially in large-cap and consumer stocks. Future enhancements will focus on incorporating advanced Large Language Models for improved news analysis, dynamic factor weighting, and intelligent report generation, alongside scalability and regulatory compliance improvements. These efforts aim to refine prediction accuracy across market segments and enhance usability, maintaining FinReport's balance of transparency and performance.

References

[1] S.-H. Poon, C. W. J. Granger, Forecasting volatility in financial markets: A review, Journal of Economic Literature 41 (2) (2003) 478–539.
[2] K. Chen, L.-A. Zhou, Y. Wang, Stock return predictability and the adaptive markets hypothesis: Evidence from century-long us data, Journal of Empirical Finance 31 (2015) 94–108.
[3] G. E. P. Box, G. M. Jenkins, Time series analysis: Forecasting and control, San Francisco: Holden-Day (1970).
[4] E. F. Fama, K. R. French, Common risk factors in the returns on stocks and bonds, Journal of Financial Economics 33 (1) (1993) 3–56.
[5] B. G. Malkiel, The efficient market hypothesis and its critics, Journal of Economic Perspectives 17 (1) (2003) 59–82.
[6] W. F. Sharpe, Capital asset prices: A theory of market equilibrium under conditions of risk, The Journal of Finance 19 (3) (1964) 425–442.
[7] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
[8] W. Bao, J. Yue, Y. Rao, A deep learning framework for financial time series using stacked autoencoders and long-short term memory, PLoS ONE 12 (7) (2017) e0180944.
[9] R. P. Schumaker, H. Chen, Textual analysis of stock market prediction using breaking financial news: The azfin text system, ACM Transactions on Information Systems 27 (2) (2009) 1–19.
[10] F. Z. Xing, E. Cambria, R. E. Welsch, Natural language based financial forecasting: a survey, Artificial Intelligence Review 50 (1) (2018) 49–73.
[11] D. B. Nelson, Conditional heteroskedasticity in asset returns: A new approach, Econometrica 59 (2) (1991) 347–370.
[12] R. F. Engle, Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation, Econometrica 50 (4) (1982) 987–1007.
[13] Kanishk1420, Finreport: Explainable stock earnings forecasting via news factor - chinese a-share stock dataset, Available at: https://github.com/Kanishk1420/FinReport-Explainable-Stock-Earnings-Forecasting-via-News-Factor, dataset contains historical stock data (2018-2021), financial news, and pre-computed technical indicators for Chinese A-share stocks. Primary dataset file: `src/stock_data.csv` (2025).
[14] T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions, European Journal of Operational Research 270 (2) (2018) 654–669.
[15] P. C. Tetlock, Giving content to investor sentiment: The role of media in the stock market, The Journal of Finance 62 (3) (2007) 1139–1168.
[16] M. T. Ribeiro, S. Singh, C. Guestrin, Why should i trust you?: Explaining the predictions of any classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016) 1135–1144.
[17] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, CoRR abs/1908.10063 (2019).
[18] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics 31 (3) (1986) 307–327.

[19] X. Ding, Y. Zhang, T. Liu, J. Duan, Deep learning for event-driven stock prediction, Proceedings of the 24th International Conference on Artificial Intelligence (2015) 2327–2333.

[20] T. Loughran, B. McDonald, When is a liability not a liability? textual analysis, dictionaries, and 10-ks, The Journal of Finance 66 (1) (2011) 35–65.