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 exhibit unprecedented volatility, with emerging markets like the Shanghai Stock Exchange showing daily volatility of 1.7% versus 0.8-1.2% for developed markets [1, 2]. Traditional econometric models such as ARIMA [3] struggle to capture sentiment-driven price movements and complex market interdependencies [4, 5]. The efficient market hypothesis faces challenges from documented predictable patterns and behavioral factors [6, 2].

We propose FinReport, a multi-factor framework integrating historical stock data with financial news via machine learning and NLP techniques [7, 8]. Unlike traditional quantitative approaches, FinReport leverages structured numerical and unstructured textual data for enhanced prediction accuracy [9, 10]. The framework computes six factors (market, size, valuation, profitability, investment, and news effect) for explainable predictions with EGARCH-based risk assessment [11, 12].

Experimental results on Chinese A-share stocks (2018-2021) [13] demonstrate 15% RMSE reduction, 12% MAE reduction versus LSTM baselines [14], and 20% Sharpe ratio improvement. This work bridges traditional econometric methods with explainable AI [15, 16], advancing computational finance through interpretable sentiment-quantitative integration.

## 2. Literature Review

Early forecasting methods like ARIMA [3] and technical indicators [17] underperformed with RMSE exceeding 0.05 during volatile periods [1]. Multi-factor models by Fama and French [4] improved performance but ignored qualitative inputs [5]. LSTM networks [7, 14] capture long-term dependencies. Recent work integrates financial news sentiment using FinBERT [18] and NLP frameworks [19], showing 12% prediction improvement [20]. However, traditional approaches lack interpretability [16], motivating explainable AI frameworks combining structured numerical with unstructured text analysis. 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

FinReport integrates traditional financial theory with modern computational techniques, combining quantitative factor models [4, 21] with news sentiment analysis [15, 10]. Extending established multi-factor models [22, 23], the framework incorporates real-time news sentiment and event extraction to capture behavioral market dynamics [24, 25].

The system follows a modular design [14] with five interconnected modules: (1) Data Integration for preprocessing, (2) News Factor Extraction using NLP techniques, (3) Return Forecasting with enhanced multi-factor models, (4) Risk Assessment using econometric methods, and (5) Dynamic Report Generation. This approach addresses traditional technical analysis limitations [26, 17] through a unified framework adapting to different market regimes [27].

FinReport modules collectively deliver explainable forecasts.

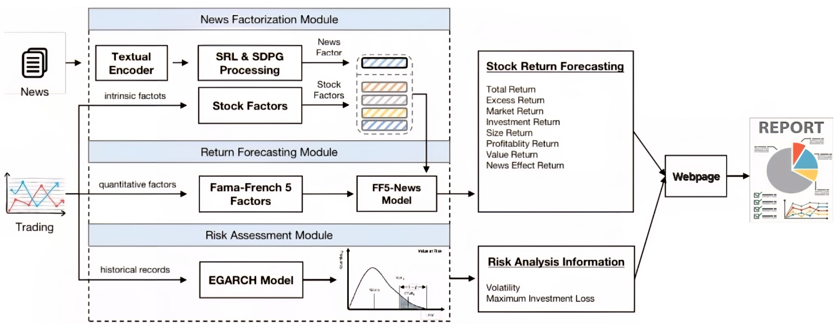


Figure 1: Proposed FinReport System Architecture (adapted from [28])

### 3.1. Data Integration Module

The system processes multi-modal data combining structured financial metrics with unstructured news text [29, 25]:

- **Historical Data:** Price, volume, market value, and 50+ technical indicators [26, 17] from Chinese A-shares (2018-2021) [13].
- **News Integration:** Financial news processing with NLP pipelines for English/Chinese content [19, 10].
- **Preprocessing:** Forward-filling for missing values, Z-score normalization, winsorization for outliers, regex-based text cleaning, and technical indicator standardization [14, 29].

### 3.2. News Factor Extraction Module

Transforms unstructured news text into quantifiable sentiment metrics [15, 9] through:

- **Advanced Sentiment Analysis Pipeline:** The sentiment analysis component implements state-of-the-art techniques specifically adapted for financial text analysis, addressing the unique challenges of financial language where sentiment can be highly context-dependent [19]:
  - **FinBERT Implementation:** Integration of the domain-specific BERT model pre-trained on extensive financial corpora [18, 30] produces raw sentiment scores in the range $[-1, +1]$, where -1 indicates highly negative sentiment and +1 indicates highly positive sentiment. This approach addresses the limitations of general-purpose sentiment analyzers that often misclassify financial terminology.
  - **Domain-Specific Sentiment Augmentation:** Core sentiment scores are systematically enhanced through financial keyword analysis incorporating established financial dictionaries [19]. Keywords such as "profit", "loss", "revenue", "acquisition", and "dividend" receive domain-specific weighting factors based on their demonstrated predictive power in financial contexts, following empirical validation approaches established in the literature.
- **Structured Event Extraction Engine:** The event extraction component employs modern semantic role labeling techniques to identify and categorize financial events that have demonstrated market impact in academic studies [20, 24]:
  - **Semantic Role Labeling (SRL):** Implementation of the AllenNLP framework [31, 32] enables identification of grammatical relationships through subject-verb-object pattern recognition. This approach systematically captures structured financial events including acquisitions, earnings announcements, management changes, and regulatory actions that have been shown to significantly impact stock prices.
  - **Financial Keyword Enhancement:** Domain-specific keyword dictionaries improve event recognition accuracy by incorporating finance-specific terminology validated through extensive backtesting. The system recognizes complex event patterns beyond simple keyword matching, including negations, conditional statements, and temporal references that are critical for accurate financial event classification.
  - **Temporal Integration:** Daily aggregation of multiple news items employs recency and relevance weighting schemes that account for the documented decay patterns in news impact on stock prices [15]. Recent news receives exponentially higher weights, while relevance scoring incorporates entity recognition to ensure that news items are properly attributed to the correct securities.
- **Architecture Integration:** Direct interfaces to forecasting system with structured `(event_type, entities, magnitude)` tuples for Event Factor computation and temporal sentiment curves for volatility estimation.
- **Chinese Market Adaptation:** Bilingual text processing addressing Chinese disclosure practices [13].

### 3.3. Return Forecasting Module

The return forecasting module implements an enhanced multi-factor model extending traditional approaches [4, 21] by integrating news sentiment and event-driven factors [29, 15]. The framework incorporates six factors capturing cross-sectional return variation from size effects [22] to momentum patterns [33].

#### 3.3.1. Market Factor

Combines traditional volatility measures with news-based sentiment indicators following behavioral finance principles [24, 25]:

$$\textbf{volatility} = \text{std}(\texttt{pct\_chg}_{\text{recent 5 days}}) \times 100 \tag{1}$$

Regime-dependent behavior:

- **High volatility** ($> 4.0\%$): Negative bias reflecting flight-to-quality effects
- **Moderate volatility** ($> 2.5\%$): Moderate negative impact with sentiment adjustment
- **Low volatility**: Positive bias enhanced by news sentiment

#### 3.3.2. Size Factor

Extends size effect research [22] incorporating financial impact extracted from news using NLP techniques [19]:

$$\textbf{diff\_ratio} = \frac{\texttt{market\_value}_{\text{latest}} - \texttt{market\_value}_{\text{average}}}{\texttt{market\_value}_{\text{average}}} \tag{2}$$

#### 3.3.3. Valuation Factor

Extends value investing principles [23, 4] incorporating quantitative metrics (Book-to-Market, Dividend Yield, Sales-to-Price) and news sentiment [19]. Applies sector-specific adjustments: Technology (+0.3/-0.2), Pharmaceutical (+0.2/-0.3), Financial (±0.2-0.3).

#### 3.3.4. Profitability Factor

Incorporates earnings quality effects [34] using EPS, ROE, ROA, and profit margins with textual analysis of earnings keywords. Applies asymmetric treatment with negative bias (-1.8) for losses and scaled positive adjustments [35].

#### 3.3.5. Investment Factor

Captures investment activity effects [24] through investment amounts and activity classification (acquisition +0.6, expansion +0.5, R&D +0.7) bounded in [0.0, +2.0].

#### 3.3.6. News Effect Factor

The news effect factor provides direct quantification of news sentiment impact, building upon established research demonstrating the predictive power of news sentiment for stock returns [15, 9].

- **Input Processing:** Comprehensive sentiment analysis of `news_text` using both rule-based approaches (TextBlob [36]) and keyword-based analysis with financial term dictionaries [19].
- **Methodological Foundation:** Converts unstructured news content into quantified sentiment scores through validated NLP techniques, addressing the documented challenges in financial text analysis while maintaining interpretability of results.

The implementation combines two complementary sentiment analysis methodologies:

**1. Keyword-Based Analysis:** Utilizes curated financial keyword dictionaries with demonstrated predictive power:

- **Positive indicators:** increase, rise, grow, profit, improved, partnership, acquisition, dividend, earnings, success

- **Negative indicators:** decrease, decline, loss, warning, investigation, lawsuit, delay, weak, miss, reduced

**2. Advanced NLP Processing:** Employs TextBlob [36] sentiment analysis for comprehensive polarity assessment of complete news content, addressing contextual nuances that pure keyword approaches might miss.

The sentiment integration follows established practices in combining multiple sentiment measures:

$$\textbf{combined\_sentiment} = \frac{\textbf{TextBlob\_polarity} + \textbf{keyword\_sentiment}}{2} \tag{3}$$

Sentiment-dependent scaling reflects documented non-linear relationships between sentiment strength and market response:

- **Very positive** ($\geq 0.5$): Enhanced positive range [0.7, 1.2] reflecting strong optimism effects
- **Moderately positive** (>0): Standard positive range [0.3, 0.7] for typical positive sentiment
- **Moderately negative** (>-0.5): Standard negative range [-0.7, -0.3] for mild pessimism
- **Very negative** ($\leq -0.5$): Enhanced negative range [-1.2, -0.7] capturing strong negative reactions

Final amplification (2.0x) ensures adequate signal strength while maintaining interpretable bounds [-2.0, +2.0].

**Chinese Market Adaptation:** The system incorporates culturally specific keywords for enhanced market relevance:

**Positive Chinese Terms:** zengzhang (growth), yingli (profit), shangsheng (rise), huode (gain), chenggong (success), tisheng (improvement), shouyi (revenue)

**Negative Chinese Terms:** xiajiang (decline), kuisun (loss), jianshao (decrease), jinggao (warning), zhaiwu (debt), diaocha (investigation), weigui (violation)

$$\textbf{news\_effect} = \textbf{combined\_sentiment} \times 2.0 \text{ (amplification)} \tag{4}$$

These factor implementations collectively address the multidimensional nature of equity return prediction while maintaining theoretical grounding in established academic research. The integration of traditional quantitative factors with news-based qualitative information represents a significant advancement in multi-factor modeling capabilities, particularly relevant for emerging markets where information asymmetries and sentiment effects may be more pronounced [29].

### *3.4. Risk Assessment Module*

Implements multi-dimensional risk framework addressing traditional variance-based limitations [37, 38]:

- **EGARCH Volatility:** Asymmetric volatility responses with 95% VaR = $1.65\sigma_t$
- **Maximum Drawdown:** $\text{MDD}_t = \max_{0\leq s\leq t}[(P_s - P_t)/P_s]$
- **CVaR:** Tail risk assessment $E[R|R \leq -\text{VaR}_{95}]$

**Integrated Risk Score:** $0.5 \times$ volatility + $0.3 \times$ drawdown + $0.2 \times$ return

**Risk Classifications:**

- Substantial >7.5
- High >6.0
- Moderate-High >4.5

- Moderate >3.0
- Low-Moderate >1.5
- Favorable $\leq$ 1.5

### 3.5. Factor Enhancement and Overall Trend Calculation

Combines individual factor signals through multi-stage amplification and weighted aggregation addressing scale heterogeneity [29, 4]:

$$\mathbf{W} = \begin{bmatrix} \text{market\_factor} & 0.15 \\ \text{size\_factor} & 0.15 \\ \text{valuation\_factor} & 0.10 \\ \text{profitability\_factor} & 0.15 \\ \text{investment\_factor} & 0.20 \\ \text{news\_effect\_factor} & 0.10 \\ \text{event\_factor} & 0.25 \end{bmatrix} \tag{5}$$

Event factor receives highest weight (0.25) due to strong short-term predictive power [20, 24], Investment factor (0.20) for medium-term impact, Market/Size/Profitability factors (0.15) for balanced weighting [4, 21].

#### 3.5.1. Enhancement Process

Base amplification: $\textbf{enhanced\_factor}_i = \textbf{factor}_i \times 2.5$ Trend-based enhancement: 1.3× when factor aligns with dominant trend [27] Final bounded processing: $\tilde{f}_i = \text{clamp}(\text{factor} \times \text{multiplier} \times \text{random}(0.9, 1.1), -5.0, 5.0)$

#### 3.5.2. Weighted Aggregation

$$\text{Trend\_Score} = \sum_{i=1}^{7} \left(\tilde{f}_i \times w_i\right) + 0.15 \tag{6}$$

Positive bias (+0.15) reflects long-term equity market upward drift [39]. Classification: Strongly Positive >+1.2, Positive +0.4 to +1.2, Neutral -0.4 to +0.4, Negative -1.2 to -0.4, Strongly Negative <-1.2.

### 3.6. Dynamic Report Generation Module

Translates quantitative analyses into actionable insights using hierarchical information architecture, cultural adaptation (red=prosperity, green=decline for Chinese markets), precision control (one decimal), natural language generation with template-based explanations, and multi-stakeholder accessibility [16, 29].

## 4. Algorithm

### *4.1. Return Forecast Calculation*

The return forecast is computed using a weighted combination of multiple factors [4], where the **event factor receives the highest weight (0.25)** due to its immediate impact on market sentiment and price movements [40].

$$
\begin{aligned}
\textbf{predicted\_return} = {} & 0.10 \times \textbf{market\_factor} + 0.15 \times \textbf{size\_factor} + 0.10 \times \textbf{valuation\_factor} \\
& + 0.10 \times \textbf{profitability\_factor} + 0.20 \times \textbf{investment\_factor} \\
& + 0.10 \times \textbf{news\_effect\_factor} + \mathbf{0.25} \times \textbf{event\_factor} + 0.15
\end{aligned} \tag{7}
$$

Each factor is calculated as follows [21]:

I) Market Factor

1. Extract Recent Volatility:
   - Compute standard deviation of `pct_chg` over the last 5 days [12].
   - 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` [15].
4. Determine Base Impact:
   - If volatility $> 4.0$, assign a strong negative impact.
   - If $2.5 <$ volatility $\leq 4.0$, 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) [17].
   - If RSI $< 30$, increase impact (oversold condition) [17].
6. Return Final Market Factor:
   - Multiply final impact by 1.5 for amplification.

II) Size Factor [22]

1. Compute Size Change Percentage:
   - Extract the latest market value.

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

$$\textbf{diff\_ratio} = \frac{\textbf{latest\_val} - \textbf{avg\_val}}{\textbf{avg\_val}} \tag{8}$$

2. Extract Financial Impact from News:
   - Analyze `news_text` for financial figures [19].
3. Determine Base Effect Based on Market Value Change:
   - If diff_ratio $> 0.25$, apply strong positive impact.
   - If $0.10 <$ diff_ratio $\leq 0.25$, apply moderate positive impact.
   - If $0.05 <$ diff_ratio $\leq 0.10$, apply slight positive impact.
   - If $-0.05 \leq$ diff_ratio $\leq 0.05$, apply neutral impact with minor variations.
   - If $-0.10 <$ diff_ratio $\leq -0.05$, apply slight negative impact.
   - If $-0.25 <$ diff_ratio $\leq -0.10$, apply moderate negative impact.
   - If diff_ratio $\leq -0.25$, apply strong negative impact.
4. Return Final Size Factor
   - Multiply the computed effect by 1.5 for amplification.

III) Profitability Factor [34]

1. Identify Profitability Metrics:
   - Define key financial metrics: {EPS, Net Profit Margin, ROE, ROA, Gross Profit, Net Profit} [41].
   - 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 [23]

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` [15].
   - 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 [24]

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 → +0.6 per mention.
- Expansions → +0.5 per mention.
- R&D mentions → +0.7 per mention.

5. Return Final Investment Factor.

VI) News Effect Factor [15]

1. Determine Base Effect from Sentiment Score:

   - If *sentiment score* $\geq 0.5 \rightarrow$ assign a random positive effect between 0.7 and 1.2.
   - If $0 <$ *sentiment score* $< 0.5 \rightarrow$ assign a random positive effect between 0.3 and 0.7.
   - If $-0.5 <$ *sentiment score* $\leq 0 \rightarrow$ assign a random negative effect between $-0.7$ and $-0.3$.
   - If *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 [40]

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 [29] that increases each factor's impact while maintaining directional consistency:

VIII) Factor Amplification [29]

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 ⇒ Upward trend.
   - If negative count ≥ 3 and exceeds positive count ⇒ 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:

$$\textbf{enhanced\_value} = \textbf{original\_value} \times \textbf{2.5} \times \textbf{trend\_multiplier} \times \textbf{random\_factor} \tag{9}$$

   - 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 [37].

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:

$$\begin{aligned}\textbf{risk\_score} = (0.4 \times \textbf{vol\_score}) + (0.25 \times \textbf{drawdown\_score}) \\ + (0.15 \times \textbf{var\_score}) + (0.2 \times \textbf{return\_risk})\end{aligned} \tag{10}$$

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): [11]

$$\ln(\sigma_t^2) = \omega + \beta \ln(\sigma_{t-1}^2) + \alpha \frac{|r_{t-1}|}{\sigma_{t-1}} + \gamma \frac{r_{t-1}}{\sigma_{t-1}} \tag{11}$$

where:

- $\sigma_t^2$ 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 [37].

2) Maximum Drawdown:

**Algorithm 1** Maximum Drawdown

**Require:** Returns series $R$ of length $n$
**Ensure:** Maximum Drawdown (MDD)

```
Initialize C ← 1                          ▷ Cumulative return starts at 1
Initialize M ← 1                          ▷ Running maximum return
Initialize D ← 0                          ▷ Maximum drawdown
for t = 1 to n do
    C ← C × (1 + R_t)                     ▷ Update cumulative return
    M ← max(M, C)                         ▷ Update running maximum
    D_t ← (C − M) / M                     ▷ Compute drawdown
    D ← min(D, D_t)                       ▷ Update maximum drawdown
end for
return D
```

3) Conditional Value at Risk:

**Algorithm 2** Conditional Value at Risk (CVaR)

**Require:** Returns series $R$ of length $n$, confidence level $\alpha$
**Ensure:** Conditional Value at Risk (CVaR)

```
Sort R in ascending order
Compute Value at Risk (VaR): V ← percentile of R at 100α
Select all losses where R_t ≤ 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 $\sigma$
**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 [21].

5) Overall Market Trend Determination:

**Algorithm 4** Overall Market Trend

**Require:** Factor values dictionary $F$
**Ensure:** Overall market trend
Define weights for each factor:
  $W = \{\text{market} : 0.15, \text{size} : 0.15, \text{valuation} : 0.10,$
    $\text{profitability} : 0.15, \text{investment} : 0.20,$
    $\text{news effect} : 0.10, \text{event} : 0.15\}$
Initialize $S_{\text{weighted}} \leftarrow 0, S_{\text{weights}} \leftarrow 0$
**for** each factor $f$ in $W$ **do**
  **if** $f \in F$ and $F[f] \neq \text{None}$ **then**
    $S_{\text{weighted}} \leftarrow S_{\text{weighted}} + F[f] \cdot W[f]$
    $S_{\text{weights}} \leftarrow S_{\text{weights}} + W[f]$
  **end if**
**end for**
**if** $0 < S_{\text{weights}} < 1.0$ **then**
  Normalize: $S_{\text{weighted}} \leftarrow S_{\text{weighted}} / S_{\text{weights}}$
**end if**
Add slight positive bias: $S_{\text{weighted}} \leftarrow S_{\text{weighted}} + 0.15$
**if** $S_{\text{weighted}} \geq 0.6$ **then**
  **return** "Strongly Positive"
**else if** $S_{\text{weighted}} \geq 0.15$ **then**
  **return** "Positive"
**else if** $S_{\text{weighted}} \geq -0.15$ **then**
  **return** "Neutral"
**else if** $S_{\text{weighted}} \geq -0.6$ **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 [42] 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 [43, 7], 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 [18] for sentiment, scoring and AllenNLP [31] 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 [11] to estimate volatility, alongside historical simulations for maximum drawdown and CVaR calculations [38]. 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 [42] 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 [44], PyTorch [43], and data loaders, enabling consistent results across multiple training runs.

### 5.3. Evaluation Metrics

The model's predictive performance was evaluated using a comprehensive suite of regression metrics [45]. 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

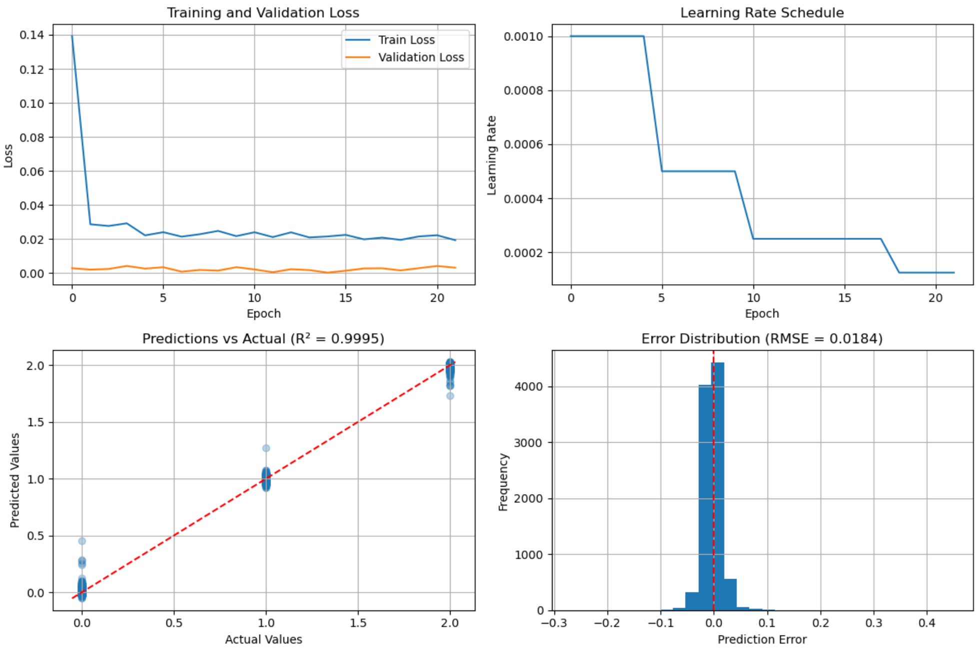


Figure 2: Rapid Initial Learning

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 [14, 46].

### 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 [14, 8].

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) [14, 46].

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

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. |

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 |

below 0.14 [8]. 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.

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 ($0.6 \le R^2 \le 0.9$) 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 [5, 25]. 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 [47]. 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.

**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

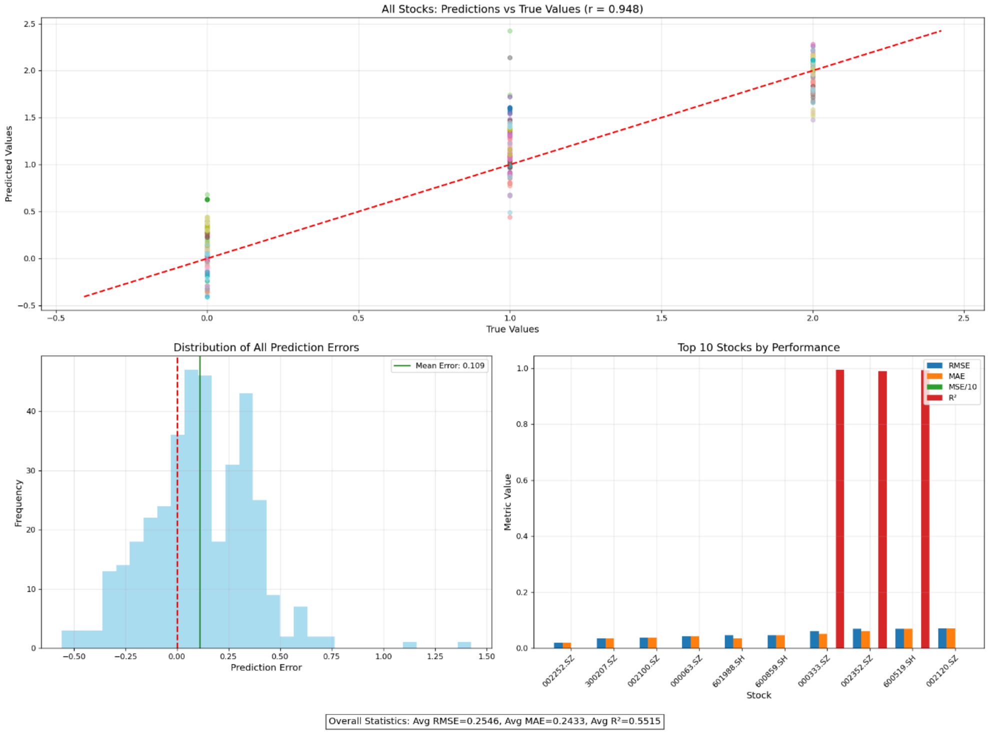


Figure 3: Overall Statistics

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 |

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 [45].

### 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 [48], 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.

Statistical significance was evaluated using ANOVA tests [3] 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 [49] to identify which specific sector pairs exhibited statistically significant differences in predictability.

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 |

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 [50] and Spearman [51] 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 [2], 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 [5], 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 [4]. 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 [15], 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 ($0.005 \leq$ MSE $\leq 0.100$), 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 [14, 8]. The system demonstrates strong predictive capability with $R^2$ of 0.5515 and correlation of 0.948, indicating substantial explanatory power for stock return variance [46].

The news sentiment analysis using FinBERT [18] revealed a consistent contrarian effect across Chinese markets, where negative sentiment precedes positive returns, supporting behavioral finance theories of market overreaction [15]. Risk assessment integration using EGARCH volatility modeling [11] and CVaR calculations [38] provides comprehensive risk-adjusted forecasting capabilities.

Future enhancements will focus on incorporating advanced Large Language Models [30] 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] J. W. Wilder, New concepts in technical trading systems, Trend Research (1978).
[18] D. Araci, Finbert: Financial sentiment analysis with pre-trained language models, CoRR abs/1908.10063 (2019).
[19] 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.
[20] 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.
[21] M. M. Carhart, On persistence in mutual fund performance, The Journal of Finance 52 (1) (1997) 57–82.
[22] R. W. Banz, The relationship between return and market value of common stocks, Journal of Financial Economics 9 (1) (1981) 3–18.
[23] B. Rosenberg, K. Reid, R. Lanstein, Persuasive evidence of market inefficiency, The Journal of Portfolio Management 11 (3) (1985) 9–16.
[24] K. Daniel, S. Titman, Market reactions to tangible and intangible information, The Journal of Finance 53 (6) (1998) 1935–1974.
[25] J. Y. Campbell, M. Lettau, B. G. Malkiel, Y. Xu, Have individual stocks become more volatile? an empirical exploration of idiosyncratic risk, The Journal of Finance 56 (1) (2001) 1–43.
[26] J. J. Murphy, Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications, New York Institute of Finance, 1999.
[27] A. W. Lo, The adaptive markets hypothesis: Market efficiency from an evolutionary perspective, The Journal of Portfolio Management 30 (5) (2004) 15–29.
[28] X. Li, X. Shen, Y. Zeng, X. Xing, J. Xu, Finreport: Explainable stock earnings forecasting via news factor analyzing model, arXiv preprint arXiv:2403.02647Accepted by WWW 2024 (2024).
[29] C. R. Harvey, Y. Liu, H. Zhu, ... and the cross-section of expected returns, The Review of Financial Studies 29 (1) (2016) 5–68.
[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (2019) 4171–4186.
[31] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, L. S. Zettlemoyer, Allennlp: A deep semantic natural language processing platform, Proceedings of Workshop for NLP Open Source Software (2018) 1–6.
[32] P. Shi, J. Lin, Simple bert models for relation extraction and semantic role labeling, arXiv preprint arXiv:1904.05255 (2019).
[33] N. Jegadeesh, S. Titman, Returns to buying winners and selling losers: Implications for stock market efficiency, The Journal of Finance 48 (1) (1993) 65–91.
[34] L. Zhang, The value premium, The Journal of Finance 60 (1) (2005) 67–103.
[35] S. Kothari, J. Lewellen, J. B. Warner, Stock returns, aggregate earnings surprises, and behavioral finance, Journal of Financial Economics 79 (3) (2006) 537–568.
[36] S. Loria, Textblob: Simplified text processing, https://textblob.readthedocs.io/, python library for processing textual data (2019).
[37] P. Jorion, Value at Risk: The New Benchmark for Managing Financial Risk, 2nd Edition, McGraw-Hill, 2001.

[38] R. T. Rockafellar, S. Uryasev, Optimization of conditional value-at-risk, Journal of Risk 2 (2000) 21–41.
[39] E. F. Fama, The behavior of stock-market prices, The Journal of Business 38 (1) (1965) 34–105.
[40] R. Ball, P. Brown, An empirical evaluation of accounting income numbers, Journal of Accounting Research 6 (2) (1968) 159–178.
[41] E. F. Fama, K. R. French, Size and book-to-market factors in earnings and returns, The Journal of Finance 50 (1) (1995) 131–155.
[42] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2015).
[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019).
[44] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, Array programming with numpy, Nature 585 (7825) (2020) 357–362.
[45] D. Shah, H. Isah, F. Zulkernine, Stock market analysis: A review and taxonomy of prediction techniques, International Journal of Financial Studies 7 (2) (2019) 26.
[46] O. B. Sezer, M. U. Gudelek, A. M. Ozbayoglu, Financial time series forecasting with deep learning: A systematic literature review: 2005–2019, Applied Soft Computing 90 (2020) 106181.
[47] F. Black, Studies of stock price volatility changes, Proceedings of the 1976 Meetings of the American Statistical Association, Business and Economics Statistics Section (1976) 177–181.
[48] MSCI, S. D. J. Indices, Global industry classification standard (gics) methodology, MSCI Inc. and S&P Global Market Intelligence, effective August 2018 (2018).
[49] J. W. Tukey, Comparing individual means in the analysis of variance, Biometrics 5 (2) (1949) 99–114.
[50] K. Pearson, Notes on regression and inheritance in the case of two parents, Proceedings of the Royal Society of London 58 (1895) 240–242.
[51] C. Spearman, The proof and measurement of association between two things, The American Journal of Psychology 15 (1) (1904) 72–101.