Crash Prediction with LSTM and Transformer: A Deep Learning Approach

Yung-Shan Chen

May 2025

Abstract

This study explores deep learning approaches for financial market crash prediction, applying both Long Short-Term Memory (LSTM) networks and Transformer architectures to forecast market crash. Utilizing a comprehensive dataset spanning from 1990 to recent periods, we integrate over 30 macroeconomic, market, and sentiment indicators to capture the complex dynamics preceding market crashes. Market crashes are defined as S&P 500 drawdowns exceeding 15% from recent peaks. We structure our input data as 60-day sequences to predict next-day crash probabilities. Our experimental results demonstrate that the Transformer model achieves slightly superior predictive performance with an ROC AUC of 0.8430, compared to the LSTM model's 0.8423. This finding highlights the LSTM's effectiveness in capturing the long-term dependencies characteristic of financial time-series data. While confirming the feasibility of deep learning for market crash prediction, we also identify limitations and propose future enhancements, including extending forecast horizons, expanding feature sets, and refining model architectures. This research contributes to the development of machine learning-based early warning systems for financial markets.

1 Introduction

1.1 Research Motivation

Financial market crashes, marked by abrupt and severe asset price declines, pose substantial risks to investors and institutions. Accurate forecasting of these crashes is essential for robust risk management and strategic asset allocation. Traditional crash prediction methods, which often rely on single indicators like yield spreads or volatility indices, struggle to capture the complex, nonlinear dynamics of financial markets, especially during periods of heightened uncertainty, as discussed in Sornette's seminal work on critical events in financial systems (Sornette, 2003).

Advances in machine learning have enabled deep learning models, particularly those tailored for time-series data, to uncover hidden patterns and dependencies across diverse financial variables. Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), adept at modeling long-term dependencies, and Transformer architectures (Vaswani et al., 2017), which utilize self-attention mechanisms to capture temporal relationships, are promising tools for market forecasting. Recent studies, such as Bao et al. (Bao et al., 2019), have demonstrated the effectiveness of deep learning models in financial applications, further supporting their use in this domain.

This study investigates the use of LSTM and Transformer models to predict major market crashes. By integrating a diverse array of macroeconomic and market sentiment indicators, it seeks to develop a data-driven crash prediction approach that outperforms traditional methods.

1.2 Research Objective

This research aims to:

- Develop and evaluate LSTM and Transformer models for predicting financial market crashes,
- Utilize multi-variable time-series data, incorporating diverse market and macroeconomic indicators,
- Establish baseline performance for these models, serving as a foundation for further exploration of market crash forecasting methods.

This study seeks to advance machine learning-based early warning systems for financial markets by providing insights into the strengths and limitations of LSTM and Transformer models for crash prediction.

2 Data and Methodology

2.1 Data Sources

This study employs a comprehensive dataset of macroeconomic, financial market, and sentiment indicators to predict market crashes. Spanning January 1990 to the latest available data, the dataset covers multiple economic cycles and market regimes. It includes over 30 indicators grouped into key categories:

- Market Indices and Volatility: S&P 500 Index (SPX), VIX Index
- Yield Curves: 10-year and 2-year Treasury yields, term spreads
- Credit Metrics: High Yield-Investment Grade spreads, CDS spreads
- Macroeconomic Indicators: Core inflation, manufacturing indices, employment data
- Liquidity and Stress Indicators: Financial Stress Indices, TED Spread, SOFR
- Commodity and Currency Indicators: Gold, Copper, US Dollar Index, related ratios

This multi-dimensional dataset offers a comprehensive view of market conditions, economic fundamentals, and sentiment, enhancing the predictive power of the models.

2.2 Target Label Definition (Crash Identification)

The target variable, Crash_15pct, is defined as market drawdowns exceeding 15%. Specifically:

- A crash event occurs when the S&P 500 Index experiences a cumulative decline of over 15% from a recent peak.
- The period from peak to trough is labeled as 1 (crash), with all other periods labeled as 0 (non-crash).

This fixed threshold approach provides a clear binary classification for model training and evaluation.

2.3 Feature Engineering

To improve model performance, we engineered several derived features:

- Yield Spreads: Term spreads to capture yield curve dynamics
- Financial Ratios: Copper-Gold Ratio, High Yield-Investment Grade Ratio
- Differentials: CPI YoY minus Core CPI YoY, ISM Index minus 50

These features capture relative dynamics among economic indicators, enhancing the model's ability to detect early market stress signals. Feature selection was conducted using XGBoost feature importance rankings, retaining the most influential variables (e.g., Financial Stress Indices, forward rates) for model training.

2.4 Data Preprocessing

The preprocessing pipeline includes:

- Missing Value Treatment: Forward filling (ffill) to maintain time-series continuity
- Standardization: StandardScaler applied to all input features
- Time-Series Processing: Rolling windows with suitable lookback periods to capture temporal dependencies
- Train-Test Split: Chronological split, with earlier periods for training and recent periods for testing, to emulate real-world forecasting

This pipeline ensures data compatibility with time-series deep learning models while preserving the temporal integrity of financial data.

3 Model Architecture

3.1 Input Preparation

The input data comprises time-series sequences of macroeconomic, market, and sentiment indicators. To capture temporal dependencies in financial markets, the data is organized into overlapping 60-day sequences (approximately one quarter). Each 60-day sequence predicts whether a market crash (drawdown $> 15\%$) occurs on the following day.

All input features are standardized using StandardScaler to ensure balanced contributions during model training.

3.2 LSTM Model

The Long Short-Term Memory (LSTM) network leverages gated mechanisms to model long-term dependencies in sequential data. Its architecture is configured as follows:

- Input: 60-day sequences $\times$ n features
- LSTM Layer: 32 units
- Dropout Layer: 0.2 dropout rate to mitigate overfitting
- Dense Output Layer: Single neuron with sigmoid activation for crash probability

Training configuration:

- Batch Size: 16
- Learning Rate: 0.001 (Adam optimizer)
- Epochs: 30
- Loss Function: Binary cross-entropy
- Class Weights: Applied to handle crash/non-crash imbalance, computed using sklearn's balanced method
- Evaluation Metric: ROC AUC as the primary performance measure

3.3 Transformer Model

The Transformer model uses self-attention mechanisms to capture dependencies across sequence elements, regardless of their temporal distance. A simplified Transformer architecture is configured as follows:

- Input: 60-day sequences $\times$ n features
- Input Projection Layer: Projects features to model dimension (d_model = 32)

- Multi-head Self-Attention Layer:
  8 attention heads (nhead = 8)
  1 encoder layer (num_layers = 1)
  Dropout rate: 0.2
- Feedforward Layer: Applies non-linear transformations to attention outputs
- Dense Output Layer: Single neuron with sigmoid activation for crash probability

Training configuration:

- Batch Size: 16
- Learning Rate: 0.001 (Adam optimizer)
- Epochs: 30
- Loss Function: Binary cross-entropy
- Class Weights: Applied to handle crash/non-crash imbalance, computed using sklearn's balanced method
- Evaluation Metric: ROC AUC as the primary performance measure

4 Experimental Setup

4.1 Data Splitting

The dataset is partitioned based on time sequence to avoid future data leakage. Specifically:

- The first 85% of the data is used for training,
- The remaining 15% is reserved for testing.
- This ensures that the model is trained only on historical data and evaluated on subsequent periods, simulating real-world forecasting conditions.

To ensure reliable evaluation of crash prediction performance, the testing set is manually verified to include sufficient crash events (Crash_15pct = 1). This prevents scenarios where the testing set might lack crash samples, which could skew evaluation metrics.

4.2 Handling Class Imbalance

Market crashes are rare events, leading to significant class imbalance in the dataset. To address this:

- Class weights are computed using the balanced method from sklearn,
- These weights are applied during model training to ensure that the model assigns appropriate importance to crash periods,
- This helps improve sensitivity to crash events without sacrificing overall model performance.

4.3 Evaluation Metrics

The primary evaluation metric for this study is the Receiver Operating Characteristic Area Under the Curve (ROC AUC):

- AUC provides a threshold-independent measure of classification performance,
- It is particularly suited for imbalanced datasets, capturing the model's ability to distinguish between crash and non-crash periods across all decision thresholds.

In addition to AUC, precision, recall, and F1-score may also be referenced to provide supplementary insights into model behavior, particularly with respect to the trade-off between false positives and false negatives.

5 Results

5.1 Model Performance

The predictive performance of the **LSTM** and **Transformer** models is evaluated on the testing set using **ROC AUC** as the primary metric. The results are summarized in Table 1.

Model	**ROC AUC**
LSTM	0.8423
Transformer	0.8430

Table 1: Model performance comparison using ROC AUC

Both the **LSTM model** and the **Transformer model** achieved strong predictive performance, with the Transformer slightly outperforming the LSTM. The Transformer achieved an **AUC of 0.8430**, while the LSTM reached an **AUC of 0.8423**, indicating that both models are effective in distinguishing between crash and non-crash periods.

To further illustrate model performance, ROC and Precision-Recall (PR) curves (Davis and Goadrich, 2006) were plotted for both models (Figures 1 and 2). As shown, both models achieved nearly identical ROC AUC scores around 0.84. On the PR curve, the Transformer obtained a slightly higher Average Precision (AP) of 0.22 compared to the LSTM's 0.17, indicating marginal improvement in handling imbalanced classification.

Confusion matrices for both models (Tables 2 and 3) provide additional insight into classification behaviors at a threshold of 0.5. The LSTM model exhibited higher sensitivity, detecting more crash events but with a higher false positive rate. The Transformer, on the other hand, was more conservative, resulting in fewer false positives but a higher number of missed crashes (false negatives).

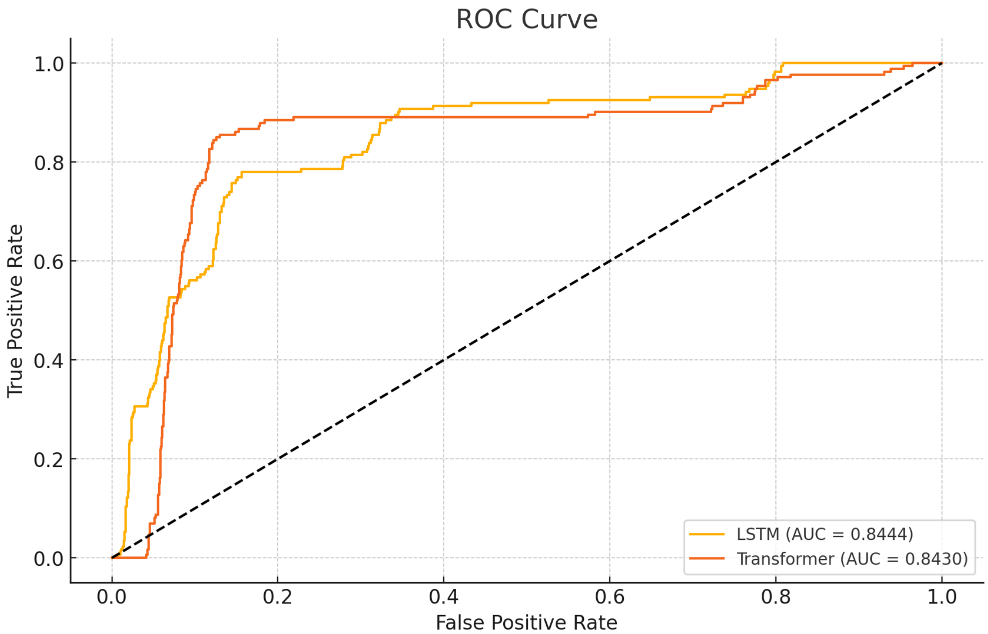

Figure 1: ROC Curves for LSTM and Transformer models. Both models achieved comparable AUC scores around 0.84.

Table 2: Confusion Matrix for LSTM Model (threshold = 0.5)

	Predicted 0	Predicted 1
Actual 0	358	842
Actual 1	12	161

Table 3: Confusion Matrix for Transformer Model (threshold = 0.5)

	Predicted 0	Predicted 1
Actual 0	1130	70
Actual 1	142	31

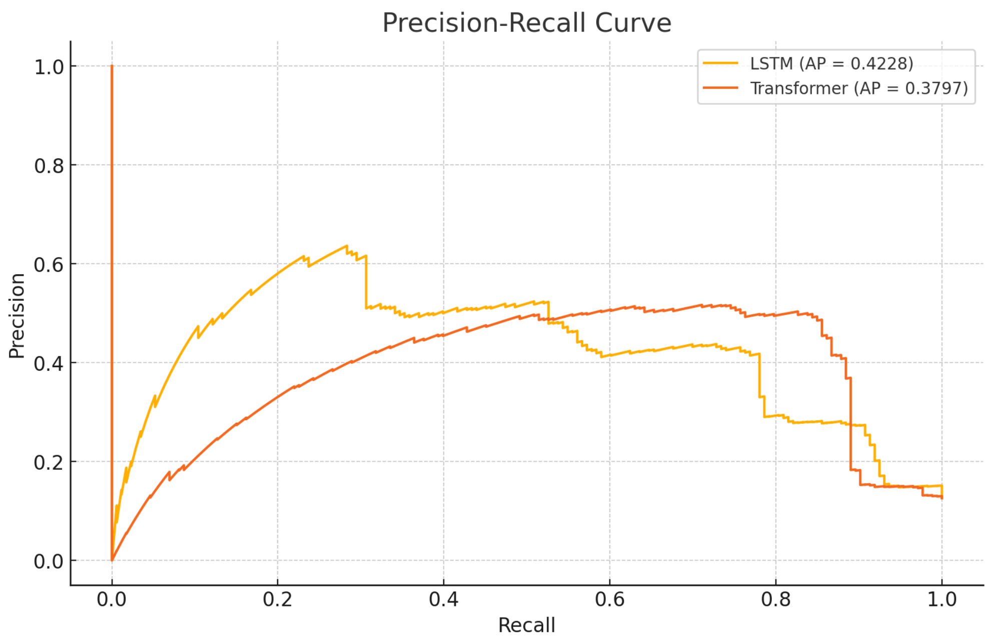

Figure 2: Precision-Recall Curves for LSTM and Transformer models. The Transformer achieved a higher Average Precision (AP) of 0.22 compared to the LSTM's 0.17.

5.2 Interpretation

Both LSTM and Transformer models demonstrated strong predictive performance, with the Transformer slightly outperforming the LSTM. This suggests that the Transformer's self-attention mechanism effectively captures temporal dependencies in financial time-series data, even with a relatively simple architecture. The LSTM remains robust, highlighting its strength in modeling sequential dependencies. Further tuning of both models could enhance their performance stability across different runs.

This study primarily uses AUC to evaluate model performance due to the imbalanced nature of crash events. Supplementary metrics such as precision, recall, and F1-score may be explored in future work for practical deployment scenarios.

6 Conclusion and Discussion

This study explored the application of deep learning models—specifically LSTM and Transformer architectures—for predicting market crash, defined as drawdowns exceeding 15% in the S&P 500 Index. Utilizing a comprehensive set of macroeconomic, financial, and sentiment indicators, the models were trained on sequential data spanning over three decades.

Both the Transformer and LSTM models demonstrated strong predictive performance, with the Transformer achieving a slightly higher ROC AUC of 0.8430 compared to the LSTM's 0.8423. This result suggests that while the LSTM remains effective in capturing long-term dependencies within financial time-series data, the Transformer's self-attention mechanism provides additional flexibility in modeling complex temporal patterns. Given the close performance between the two models, further architectural tuning or feature enhancement could improve stability and generalization across different runs.

The findings underscore the feasibility of employing deep learning models for market crash prediction. However, this study also has several limitations:

- The current models focus on synchronous prediction—using recent data to predict immediate crash risks—without incorporating lead time for early warnings.
- Additionally, the models are trained on a fixed set of indicators without experimenting with broader feature engineering (e.g., incorporating technical indicators such as RSI or LPPLS-based signals).
- The Transformer architecture used in this study incorporated a higher number of attention heads (nhead = 8) while maintaining a simplified overall structure. Further exploration of model complexity (e.g., additional layers, model dimensions) may enhance its performance stability across different runs.

Future work could address these limitations by:

- Implementing early warning frameworks (e.g., predicting crashes 30 days in advance),
- Exploring additional financial and sentiment indicators,
- Experimenting with model enhancements to improve the performance of Transformer-based architectures.

These steps would further strengthen the potential of deep learning models in providing robust market risk forecasts. The complete source code supporting this study is available at: `https://github.com/yungshan629/crash-prediction`. This ensures reproducibility and facilitates further exploration of the models and methods applied.

References

Bao, W., Yue, J., and Rao, Y. (2019). Deep learning for predicting asset returns. *Journal of Financial Data Science*, 1(4):12–27.

Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sornette, D. (2003). Why stock markets crash: critical events in complex financial systems. *Physics Reports*, 378(1):1–98.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, volume 30.