Dynamic Early Warning System for Financial Crashes

Mokha Lerthsuwanroj

School of Computing Science

Sir Alwyn Williams Building

University of Glasgow

G12 8RZ

A dissertation presented in part fulfillment of the requirements of the Degree of Master of Science at the University of Glasgow

<Date of submission placed here>

Abstract

<Abstract goes here…>

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic form.

<Please note that you are under no obligation to sign this declaration, but doing so would help future students.>

Name: Signature:

Acknowledgements

<Acknowledgements go here>

Contents

<Update the table of contents by right-clicking on it and selecting Update Field… and then select page numbers only.>

Chapter 1 Introduction

1.1 Motivation

Financial market crashes have historically affected to the economic and social disruptions, leading to the downturn, unemployment, and loss of investor confidence. Early detection and intervention are critical for minimizing the impact of such crises on those affected.

In recent years, sentiment analysis has emerged as a powerful tool in financial risk monitoring, offering insights into investor expectations, fears, and behaviours. When sudden shifts in investor sentiment are combined with traditional volatility indicators, they can provide early signals of impending market instability (Liu et al., 2023). However, most existing early warning systems rely on static models that struggle to adapt to the fast-changing dynamics of the modern financial markets (Kustina et al., 2023). By integrating both market and sentiment volatility within a dynamic framework, this research aims to develop more responsive and accurate tool for crash prediction.

1.2 Purpose

The primary purpose of this research is to develop a dynamic early warning system (EWS) that integrates both market-based and sentiment-based volatility indicators to enhance the early detection of financial market crashes. Given the increasing complexity and unpredictability of financial markets, especially during periods of heightened uncertainty, there is a growing need for more adaptive and timely forecasting models.

This study seeks to address by analysing how time-varying patterns in market-based and sentiment-based volatility relate to the occurrence of past financial crises. Specifically, the research will make use of historical financial news headlines to extract relevant sentiment signals. These sentiment indicators will then be combined with traditional market-based volatility measures, using the S&P500 closing prices, within a dynamic modelling framework designed to capture the evolving relationships between these variables over time.

Chapter 2 Survey

2.1 Background Survey

2.1.1 Current Methods for Early Warning Systems (EWS)

Early warning systems (EWS) for financial crashes have evolved from simple statistical models to more advanced machine learning and nonlinear approaches. The traditional statistic models, such as logistic regression, have been widely used to detect early signals for financial crashes using predefined relationships between risk indicators and crash probabilities. Such models often suffer from rigid parameterization and lagging indicators, limiting their ability to capture regime shifts or sudden market changes (Kustina et al., 2023), in contrast with the more recent research that explored the nonlinear approach to overcome this limitation. Nonlinear algorithms, support vector machines (SVM), and neural networks have shown improved capacity for capturing the complex relationships in the real-world financial markets (Song et al., 2024), allowing more flexibility when modelling market risks and crash probabilities as new data becomes available. Empirical evidence supports that dynamic nonlinear methods outperform static models, providing better crisis prediction under changing market environments (Song et al., 2024).

Beside the market-based indicators, sentiment analysis has also gained attention in financial crash predictions. The rise of social media platforms such as Twitter (now known as X), along with financial news sources, has provided rich datasets for capturing investor mood and behaviours (Liu et al., 2023). However, extracting signals from this unstructured data often produces noisy which remains a challenge. Liu, Leu, and Holst (2023) proposed a method using FinBERT[1] combined with an ensemble SVM to reduce noise and filter out irrelevant content from social media discussions.

Huang et al. (2020) showed that FinBERT, which is specifically pre-trained on financial texts including earnings call transcripts, analyst reports, and financial news articles, significantly outperforms general-purpose language models like BERT and traditional approaches in various financial information extraction tasks, including the LM dictionary, NB, SVM, RF, CNN, and LSTM. The model's specialized training on domain-specific vocabulary and financial terminology enables it to better understand the context inherent in financial communications, resulting in improved accuracy for sentiment classification, named entity recognition, and relationship extraction from financial documents (Huang et al., 2020). However, it is important to note that FinBERT demonstrated superiority applies specifically to financial text analysis tasks, and not directly to modelling financial market volatility.

As noted by Parras-Gutiérrez et al. (2014), forecasting models usually designed for short-term or one-step-ahead predictions due to the increasing in difficulty and unreliability of medium- and long-term forecasts caused by error propagation over time. To complement this perspective, Allaj and Sanfelici (2023) introduced a time-

[1] https://huggingface.co/ProsusAI/finbert

varying window (e.g., $T = 22, 66, 132$ days) in the context of early warning systems for financial instability. This approach acknowledges the changing nature of financial markets and allows models to capture different temporal dynamics ranging within a unified structure. Together, these insights lead to a multi-horizon modelling method that balances predictive accuracy with a greater understanding of time.

2.1.2 Various Choices of Feature Extraction and Machine Learning Methods for Stock Price Prediction

Bonde and Khaled (2012) explores the effectiveness of various feature extraction strategies and machine learning algorithms in forecasting stock prices from financial markets, evaluates a combination of features and classifiers to identify the most predictive configurations. The study examines eight different feature sets that represent different levels of contextual and historical integration, ranging from minimalist (Company alone) to comprehensive market-informed (NASDAQ + S&P + Company). and evaluates four machine learning techniques to assess their effectiveness: neural networks, Sequential Minimal Optimization (SMO), bagging using SMO, and M5P.

Among these, SMO and bagging using SMO demonstrated superior performance, especially when combined with rich feature sets like Volume + Company and NASDAQ + S&P + Company. The ensemble approach of bagging helped improve generalization and robustness against overfitting.

In contrast, while neural networks are often favoured for financial modelling due to their flexibility, their performance in this study was subpar. The authors attributed this to insufficient tuning and a potential mismatch with the chosen features or architecture. They suggested that, with proper hyperparameter optimization, neural networks could perform competitively (Bonde et al., 2012).

2.1.3 Key Crash Indicator

Most studies operationalize a financial crash using a binary crash indicator equation, where a crash is identified based on a significant drop in asset prices or index returns over a specified time window. A common method involves calculating the log return of closing prices over a fixed period (e.g., 5-day or 10-day intervals) and labelling an observation as a "crash" if the return falls below a predefined threshold which often set at the 10th percentile of historical returns or a fixed percentage drop, such as −10% (Kaminsky et al., 1998). Nonetheless, early warning models also extend to other types of financial crises, such as currency and sovereign debt. Kaminsky and Reinhart (1999), for example, define currency crises based on a sharp depreciation of the exchange rate coupled with reserve losses, using an exchange market pressure index. Bussière and Fratzscher (2006) extend this framework to sovereign debt crises by incorporating a wide range of macroeconomic variables, flagging a crisis when key thresholds are breached.

In addition, volatility also remains one of the most important indicators in crash prediction research. To flag the potential financial instability, both realized volatility (observed historical price variability) and price-volatility feedback rate have been used (Allaj & Sanfelici, 2023). Pattern of increased volatility generally precede market downturns, making it useful for early warning system frameworks.

Mentioning the traditional risk measures, Value-at-Risk (VaR) and Expected Shortfall (ES) are widely used as quantitative measures to assess market risk and potential losses under various conditions. However, both VaR and ES forecasts often rely on models with specific distributional or structural assumptions (Allaj & Sanfelici, 2023), which may not capture sudden market regime shifts, nonlinear behaviours. This is especially true in emerging markets, where volatility is typically higher and market dynamics are less predictable.

A recent study by Le (2024) examined the effectiveness of combining multiple VaR and ES forecasting models in the context of the Vietnamese stock market. The research found that forecast combination techniques, such as weighted averaging of outputs from different models (e.g., GARCH (Bollerslev, 1986), and CAViaR (Engle et al., 2004), significantly improved the accuracy and reliability of risk forecasts, especially during periods of high market volatility. The combined models showed better back testing performance and greater compliance with regulatory risk thresholds, compared to any single model (Le, 2024).

2.1.4 Lags Selection

In time series analysis, lags involve using past data points to predict the future values. Specifically, a lagged variable is a prior value of the same variable, shifted backward in time by a specific number of time steps. The purpose of including lags is to capture the temporal dependencies, persistence, or self-correlation which commonly found in sequential data such as stock returns, volatility, or macroeconomic indicators (Box, Jenkins, & Reinsel, 2008).

The choice of how many lags to include directly impacts a model's ability to capture relevant temporal dependencies. Parras-Gutiérrez et al. (2014) addressed this issue in the context of short-, medium-, and long-term time series forecasting using the L-Co-R algorithm, which incorporates a cooperative-competitive evolutionary strategy to automatically select appropriate lags. Their approach revealed that lag structures reflect a broader challenge in time series modelling: too few lags may underfit, missing important dependencies, while too many lags may lead to overfitting or increased computational complexity. The study emphasizes that adaptive or data-driven lag selection methods, such as genetic algorithms or information-theoretic criteria (e.g., Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC)), can enhance model generalizability.

2.1.5 Rationale for Focusing on Index-Level Predictions

Index-level models offer several advantages, including aggregation benefits that help reduce noise and unexpected shocks from individual stocks. According to Park et al. (2024), the research has shown that top-down index forecasts tend to be more accurate and informative than bottom-up aggregation of individual stock predictions, particularly for systemic risk assessment. By concentrating on index-level sentiment and market volatility, the model can better capture macro-level signals that reflect wide-range market conditions (Park et al., 2024).

2.1.6 Limitations in Crash Prediction

As highlighted by Andreou et al. (2019), a major concern is the widespread reliance on annual distress risk measures, which may overlook short-term fluctuations that are more relevant to crash events. By using monthly data, the authors demonstrate that short-term increases in distress risk significantly predict future crashes, which earlier studies likely missed due to insufficient temporal resolution. Another key limitation is the lack of proper treatment for endogeneity, including reverse causality and missing variable bias, which the authors address through instrumental variable methods and a quasi-experimental design using the Sarbanes–Oxley Act of 2002 (Lander, 2002).

The literature also falls short in explaining the basic mechanisms of crash risk, particularly the role of managers hiding bad news during difficult periods. While financial opacity and information gaps have been recognized, their interaction with distress risk has rarely been tested in practice. Moreover, crash risk is often undervalued in real-world situations, despite being non-diversifiable, a crucial difference from volatility risk that poses significant threats especially to poorly diversified retail investors. The limited attention to earnings smoothing strategies and unclear financial reporting further weakens the explanatory power of many models (Andreou et al., 2019).

2.2 Research Objectives

The goal of this project is to assess the following objectives:

1. To investigate the contribution of both market-based and sentiment-based volatility indicators to financial crash prediction by evaluating their feature importance.
2. To develop a dynamic modelling framework to capture the evolving relationship between sentiment-driven and market-based volatility indicators.
3. To evaluate and compare the predictive performance of the proposed dynamic early warning system against traditional static statistical models, with a particular focus on assessing the added value of sentiment-based inputs.
4. To validate the robustness and generalizability of the developed model across different market environments and historical crisis periods.

2.3 Research Questions

Based on the stated objectives, this study seeks to answer the following key research questions:

1. How do market-based and sentiment-based volatility indicators individually and jointly relate to the timing and occurrence of past financial crashes at the index level?
2. Can a dynamic early warning system outperform traditional static models in predicting financial market crashes?
3. To what extent does sentiment volatility enhance the performance of early warning models compared to using market-based indicators alone?
4. Is the proposed dynamic early warning system model robust and generalizable across different market conditions and historical crisis periods?

Chapter 3 Design and Implementation

3.1 Data Collection and Preprocessing

For this study, the dataset S&P 500 with Financial News Headlines (2008-2024)[2] was utilized, which is publicly available on Kaggle. This dataset combines daily S&P 500 stock market data with corresponding financial news headlines, enabling the analysis of market behaviour alongside sentiment-driven news information. The dataset covers the period from August 2008 through 2024, providing a comprehensive view of the S&P 500's price movements along with market-relevant news during this period.

3.1.1 Dataset Structure and Columns

The dataset consists of multiple columns, which can be broadly categorized into market data and news headlines:

1. Title: This column includes the financial news headline(s) published on the respective trading date. These headlines reflect the key news events, market sentiments, or significant announcements that could potentially influence investor behaviour and market movements.
2. Date: This column records the trading date in the format YYYY-MM-DD. It corresponds to the actual days when the S&P 500 market was open, and trading took place.
3. CP: This column provides the closing price of the S&P 500 index on the given trading date, representing the final price at which the index traded on that day.

3.1.2 Labelling Crash Events

To identify potential future market crashes within the dataset, we introduce a labelling method that flags whether a significant drop in the S&P 500 closing price occurs within a defined future period. This approach allows us to create a binary target variable indicating the presence or absence of a market crash after a given trading day.

The key parameters in this labelling process are:

1. Look-ahead period: The number of trading days into the future over which we examine the price drop. Given approximately 132 trading days in a year, this value is set to 132 to analyse a half-year ahead horizon.
2. Drop threshold: The fractional threshold that defines a crash. If the future closing price drops below this fraction of the current closing price, a crash is labelled. For example, a threshold of 0.9 corresponds to a 10% decline.

3.1.3 Data Splitting and Class Imbalance Handling

The dataset was divided into training and testing portions using a time-based approach to maintain proper temporal sequence and prevent information leakage.

[2] https://www.kaggle.com/datasets/dyutidasmahaptra/s-and-p-500-with-financial-news-headlines-20082024

Data collected from 02/01/2008 to 31/12/2021 formed the training dataset, while data from 01/01/2022 to 04/03/2024 constituted the test dataset. Training data points comprised 2,969 instances, representing 84.66% of the total dataset, while 538 data points (15.34%) were recorded for testing.

In terms of event occurrence, there were 163 crashes before 2022 and 43 crashes after 2022. This corresponds to 79.13% of crashes occurring in the earlier subset and 20.87% in the later subset.

In addition, to mitigate the highly imbalanced issue with only 5.87% of crash instances, which is 206 crashes from all instances, we apply the Synthetic Minority Over-sampling Technique (SMOTE) from imbalanced learn package[3], a widely used resampling method that generates synthetic examples of the minority class based on feature space similarities between existing minority instances (Chawla et al., 2002). By interpolating new samples rather than simply duplicating existing ones, SMOTE improves the generalizability of the model and helps it learn decision boundaries that are more representative of both classes (Budhidharma et al., 2023).

3.1.4 Market Volatility

To capture the market's price variability over different time horizons, we calculate the n-day market volatility $MV_t^{(n)}$ based on the rolling standard deviation of daily returns scaled by the square root of the window length n. This scaling converts the volatility estimate to the n-day horizon, assuming returns are independent and identically distributed.

Given daily returns R_t calculated as the percentage change in closing prices CP_t:

$$R_t = \frac{CP_t - CP_{t-1}}{CP_{t-1}} \times 100$$

Market volatility is typically computed from returns, which are often modeled as a random walk or a Brownian motion (Osborne, 1959) process where daily returns are assumed to be independent and identically distributed (i.i.d.) (Ren et al., 2017). Under this assumption, the variance of returns over n days scales linearly with n:

$$\sigma^2\left(R_t^{(n)}\right) = n \times \sigma^2\left(R_t^{(1)}\right)$$

Since standard deviation is the square root of variance, the n-day standard deviation scales with $\sqrt{n}$. This scaling allows us to compare volatility estimates over different time frames on a consistent basis.

At this point, the n-day market volatility at day t defined as:

$$MV_t{}^{(n)} = \sqrt{n} \times \sqrt{\frac{1}{n}\sum_{j=t-n+1}^{t}(\bar{R}_j - \bar{R}_t^{(n)})^2}$$

where $\bar{R}_t^{(n)}$ is the mean of $\bar{R}$ over the n-day window ending at day t.

[3] https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html

3.1.5 Sentiment Volatility

To quantify the sentiment expressed in financial news headlines, we employ a custom sentiment score derived from the FinBERT model's output probabilities. FinBERT classifies each input text into three sentiment categories: negative, neutral, and positive, producing corresponding probabilities. Rather than relying solely on discrete class labels, we calculate a continuous sentiment score S defined as the difference between the positive and negative probabilities (Hiew et al., 2019):

$$S_t = \frac{T_t^{pos} - T_t^{neg}}{T_t^{pos} + T_t^{neu} + T_t^{neg}} = P_t^{pos} - P_t^{neg}$$

where $T_t^{pos}, T_t^{neu}, T_t^{neg}$ are the number of positive, neutral, and negative texts within the period, while P_t^{pos}, P_t^{neg} are the probability of the headline at time t being classified as positive and negative. This formulation captures the net sentiment polarity by balancing positive and negative signals while effectively ignoring the neutral component.

To capture the temporal dynamics and fluctuations in market sentiment, we compute the daily average sentiment score and its rolling volatility over various time horizons. This is formalized as:

$$\bar{S_t} = \frac{1}{N_t} \sum_{i=1}^{N_t} S_{i,t}$$

where $\bar{S_t}$ represents the average sentiment score on the day t, $S_{i,t}$ is the sentiment score of the i-th headline, and N_t is the total number of headlines on that day.

Then compute the rolling standard deviation across multiple window lengths, corresponding to different market time frames: one week (5 trading days), one month (22 days), one quarter (66 days), and half a year (132 days). For each window n, the sentiment volatility $SV_t{}^{(n)}$ is given by:

$$SV_t{}^{(n)} = \sqrt{\frac{1}{n} \sum_{j=t-n+1}^{t} (\bar{S_J} - \bar{S}_t^{(n)})^2}$$

where $\bar{S}_t^{(n)}$ is the mean of $\bar{S}$ over the n-day window ending at day t.

Sentiment volatility is usually calculated as the rolling standard deviation of a sentiment score or index, which is not necessarily a return or increment-like variable. The sentimental data often reflects an aggregate or smooth measure of market mood or perception and may have different statistical properties than returns (e.g., not i.i.d., possibly autocorrelated).

Thus, when computing sentiment volatility, just take the rolling standard deviation over the window without multiplying by n because the interpretation of sentiment volatility is often relative variability within that window. Moreover, it is treated more as a descriptive statistic rather than a time-scaled measure like market volatility.

3.1.6 VaR and ES Features

To enrich the model with forward-looking risk measures, we incorporate parametric Value-at-Risk (VaR) and Expected Shortfall (ES) as additional features. These measures are computed under the assumption of normally distributed returns using a rolling window approach. For each window length n, and standard deviation of σ_t daily returns are calculated. The one-day VaR at confidence level α is given by:

$$VaR_t^{(n)} = -(\mu_t + z_\alpha \cdot \sigma_t)$$

where z_α is the z-score corresponding to the confidence level α (e.g., $z_{0.05} \approx 1.64$ for 95% confidence). The corresponding ES, which estimates the expected loss conditional on the loss exceeding the VaR threshold, is computed as:

$$ES_t^{(n)} = -(\mu_t + \sigma_t \cdot \frac{\phi(z_\alpha)}{\alpha})$$

where $\phi(z_\alpha)$ is the standard normal probability density function evaluated at z_α, and α is the tail probability. These risk measures provide a theoretically grounded way to quantify downside risk and help capture volatility dynamics that may precede extreme market events.

When estimating VaR and ES, the rolling standard deviation is typically used without the n factor because the goal is to estimate 1-day risk based on the most recent n-day window (Hällman, 2017). In this context, the standard deviation represents the forecast of next-day volatility, not an aggregated risk over n days.

3.1.7 Preprocessing and Scaling

The feature columns with missing data were imputed using mean values calculated from the training dataset, followed by feature scaling using the StandardScaler from sklearn package[4]. This scaler standardizes features by removing the mean and scaling to unit variance, ensuring that each feature contributes equally to the model training process and preventing features with larger scales from dominating. Additionally, SMOTE was implemented solely on the training set to address class imbalance issues.

3.2 Model Implementation

3.2.1 Static Logistic Regression

The model estimates the conditional probability of a future crash $Y_t = 1$ at time t, given the features, by modelling the probability as a function:

$$P(Y_t = 1|X_t) = \sigma(\beta_0 + \beta^T \cdot X_t)$$

[4] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

where $\sigma(*)$ is the sigmoid function, β are the coefficients learned from the data, and β_0 is the intercept. The log-odds (logit) form of model shown as:

$$\log\left(\frac{P(Y_t = 1|X_t)}{1 - P(Y_t = 1|X_t)}\right) = \beta_0 + \beta_1 \cdot MV_t^{(n)} + \beta_2 \cdot SV_t^{(n)} + \beta_3 \cdot VaR_t^{(n)} + \beta_4 \cdot ES_t^{(n)}$$

$$X_t = [MV_t^{(n)}, SV_t^{(n)}, VaR_t^{(n)}, ES_t^{(n)}]$$

$$\beta = [\beta_0, \beta_1, \beta_2, \beta_3, \beta_4]$$

where X_t is the feature vector, and β are set of the model coefficients learned during training.

Logistic regression with elastic net regularization was fitted to the resampled and scaled training data. The elastic net combines L1 and L2 penalties to encourage both sparsity and stability in the model coefficients. The model was optimized using the 'SAGA' solver (Defazio et al., 2014) with a maximum of 1000 iterations, allowing effective handling of regularization and class imbalance. The penalty term in elastic net regularization defined as:

$$\lambda(\alpha\left||\beta|\right|_1 + \frac{1-\alpha}{2}\left||\beta|\right|_2^2)$$

where $\left||\beta|\right|_1 = \sum_j |\beta|_j$ is the L1 norm, $\left||\beta|\right|_2^2 = \sum_j |\beta|_j^2$ is the squared L2 norm, $\alpha \epsilon [0, 1]$ controls the L1 vs L2 trade-off, and $\lambda > 0$ controls overall strength of regularization.

As for this model, a static threshold of 0.5 was initially used to convert predicted probabilities into class labels, where probabilities above or equal to 0.5 were classified as positive cases.

3.2.2 Dynamic Logistic Regression + Dynamic Threshold

To capture the temporal dependencies and improve the predictive performance of the crash classification model, we extend the baseline logistic regression by incorporating lagged features of the original market and sentiment volatility indicators, VaR, and ES. Specifically, for each volatility window n, the model uses both the current-day features and their lagged values at t, where the lag is a fixed hyperparameter (e.g., 10 days). The extended feature vector and coefficients are written as:

$$X_t = [MV_t^{(n)}, SV_t^{(n)}, VaR_t^{(n)}, ES_t^{(n)}, MV_{t-lag}^{(n)}, SV_{t-lag}^{(n)}, VaR_{t-lag}^{(n)}, ES_{t-lag}^{(n)}]$$

$$\beta = [\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8]$$

Furthermore, a dynamic threshold is applied on the predicted probabilities to convert them into binary predictions, rather than using a fixed threshold like 0.5 in our static logistic regression model. The optimal threshold is chosen by maximizing the F1-score, calculated from the precision-recall curve on the test data:

$$Threshold = argmax_\theta(\frac{2 \cdot Precision(\theta) \cdot Recall(\theta)}{Precision(\theta) + Recall(\theta)})$$

This approach accounts for class imbalance and different cost sensitivities in prediction errors, yielding a threshold tuned specifically to the model's performance on each test set.

3.2.3 CNN

For each volatility horizon n, we constructed the input feature set using the same variables as the first static regression model, then standardized using StandardScaler, and transformed into rolling sequences of length $w = 22$ (approximately one trading month) to serve as inputs to the CNN. The target label Y_t was aligned with the endpoint of each sequence, maintaining proper temporal order.

The CNN architecture consisted of two 1D convolutional layers with ReLU activations: the first with 64 filters and a kernel size of 3, followed by max pooling, and a second convolutional layer with 128 filters. A global max pooling layer was applied to extract the most salient features, followed by dropout regularization and dense layers. The final output layer used a sigmoid activation function to produce probability estimates for binary classification.

To handle class imbalance without disrupting the temporal structure of sequences, we applied class weighting during training. These weights were calculated using class weight computing from sklearn[5] to penalize misclassification of the minority crash class. Lastly, a dynamic thresholding method was also applied to the CNN outputs.

3.2.4 LSTM

The LSTM architecture comprised a single LSTM layer with 64 units, followed by a dropout layer (rate = 0.3) to reduce overfitting, and a fully connected dense output layer with a sigmoid activation function for binary classification. The model was compiled with the binary cross-entropy loss function and optimized using the Adam optimizer with a learning rate of 0.001. Training was performed for 10 epochs using a batch size of 32, with 20% of the training data reserved for validation.

As with the dynamic logistic regression and CNN models, a dynamic thresholding approach was applied to convert predicted probabilities into binary crash predictions.

3.3 Evaluation Metrics

In evaluating Early Warning Systems (EWS) for financial crash prediction, selecting appropriate performance metrics is essential due to the highly imbalanced nature of crash events and the asymmetric cost of misclassification. In this context, we weighted evaluation metrics based on their practical importance and alignment with the goals of a crash-detection system.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

3.3.1 Confusion Matrix

The confusion matrix provides a tabular representation of classification outcomes by comparing predicted labels with actual labels. It includes four components:

- True Positives (TP): Correctly predicted crash instances.
- True Negatives (TN): Correctly predicted non-crash instances.
- False Positives (FP): Non-crash instances incorrectly predicted as crashes.
- False Negatives (FN): Crash instances missed by the model.

Confusion Matrix–Derived Metrics

Several performance metrics are derived from the confusion matrix to provide a comprehensive evaluation of the classification model, especially in the context of Early Warning Systems (EWS):

1. True Positive Rate (TPR) / Sensitivity / Recall: Measures the proportion of actual crash events correctly identified. In EWS applications, high sensitivity is crucial to ensure that warnings are triggered for most potential crashes.

$$TPR = \frac{TP}{TP + FN}$$

2. True Negative Rate (TNR) / Specificity: Indicates the proportion of non-crash events correctly predicted.

$$TNR = \frac{TN}{TN + FP}$$

3. False Positive Rate (FPR): Reflects the likelihood of the model incorrectly raising a crash alert when none occurred.

$$FPR = \frac{FP}{FP + TN}$$

4. False Negative Rate (FNR): Indicates how often the model fails to detect an actual crash, especially critical in risk-sensitive domains.

$$FNR = \frac{FN}{TP + FN}$$

5. Positive Predictive Value (PPV) / Precision: Shows the proportion of predicted crashes that were actual crashes.

$$PPV = \frac{TP}{TP + FP}$$

6. False Omission Rate (FOR): Measures the probability of missing a crash among the instances predicted as non-crash.

$$FOR = \frac{FN}{FN + TN}$$

7. Noise-to-Signal Ratio (NSR): Quantifies the level of false alarms relative to correct crash predictions. A lower NSR indicates a more reliable early warning system.

$$NSR = \frac{FPR}{TPR}$$

According to Kaminsky (1998), an NSR below 0.34 is considered a reliable threshold for identifying real banking crises. In contrast, an NSR equal to or exceeding 1.0 suggests that the model generates more noise than useful signals, rendering it ineffective for practical crisis prediction.

8. Accuracy (ACC): Represents the overall proportion of correct predictions.

$$ACC = \frac{TP + TN}{Total}$$

3.3.2 Area Under the ROC Curve (AUC)

The AUC measures the ability of a classifier to distinguish between the positive and negative classes across all possible classification thresholds. It is derived from the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate against the false positive rate. AUC values range from 0 to 1, where 0.5 indicates random guessing and 1.0 indicates perfect classification. While useful, it does not directly account for the practical cost of false negatives versus false positives, so it is used as a complementary indicator rather than a sole decision criterion.

3.3.3 Weighted Scoring Function

To evaluate model performance more holistically, a custom weighted scoring function was defined. This function assigns greater positive weight to metrics that are more critical for early warning systems, such as TPR, TNR and PPV, while penalizing undesirable outcomes FPR and FNR by giving a negative weight.

Specifically, the weighted score is computed as:

$$Score = \sum_{i=1}^{n} w_i \cdot m_i$$

where $m_i \in \{TPR, TNR, FPR, FNR, PPV, FOR, NSR, ACC\}$ are the metric values, and $w_i \in \{w_{TPR}, w_{TNR}, w_{FPR}, w_{FNR}, w_{PPV}, w_{FOR}, w_{NSR}, w_{ACC}\}$ are the weights for each corresponding metric.

3.4 Results

3.4.1 Volatility Clustering Across Time Windows

To assess the predictive utility of sentiment and market volatility in anticipating financial crashes, we examine their joint behaviour over multiple forecast horizons. Figures 1 visualize the relationships between n-day market volatility and n-day sentiment volatility, labelled by future crash occurrence (crash = 1, no crash = 0), across five rolling windows: 5, 22, 66, and 132 days.

Figure 1 reveals horizon-dependent structural differences in how volatility metrics associate with future crash outcomes. For all observed horizons, crash points (orange) are diffusely scattered across the volatility space and exhibit substantial overlap with non-crash points (blue). Nevertheless, a significant observation is the shorter window (5-day and 22-day) show increased crash density in high sentiment volatility regimes, suggesting that shorter-term financial distress is more likely to be preceded by sustained turbulence in investor sentiment. The result is consistent with behavioural finance theories suggesting that investor mood and narrative instability often precede tangible price-based dislocations (Gaies et al., 2022).

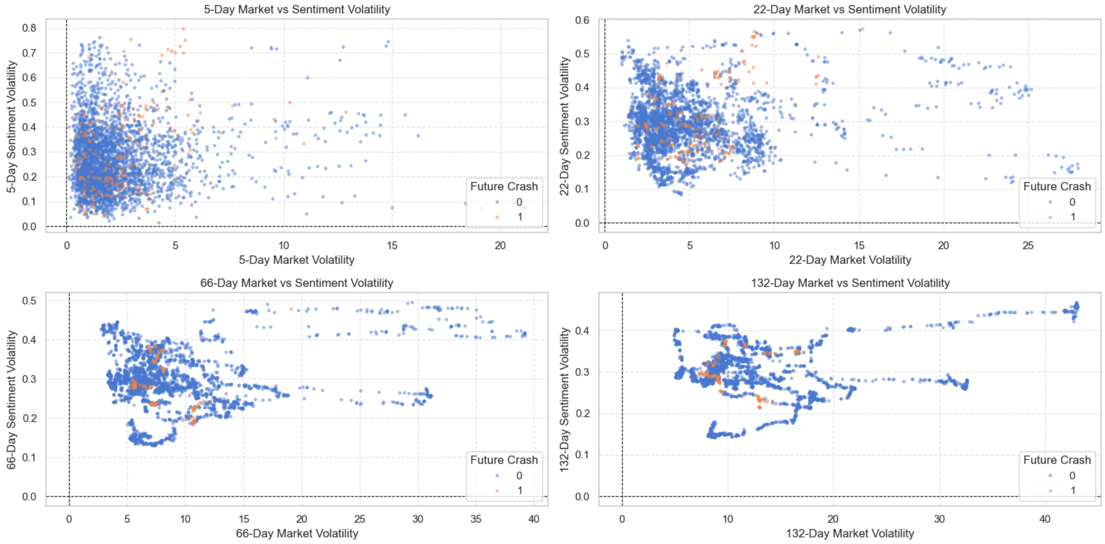

Figure 1: Relationship between n-day market volatility and n-day sentiment volatility for each volatility period

3.4.2 Window Size Analysis

Examination in Appendix A reveals that for window sizes 66 and 132, model performance significantly deteriorates. The ROC curves for these configurations are generally close to or below the diagonal, especially for the Market features (Figures A1, A4, A7, A10) and all LSTM models (Figures A10-12). Such results indicate that these window sizes fail to capture meaningful predictive patterns and may even introduce noise that degrades performance.

By contrast, window sizes 5 and 22 show consistently higher ROC curve separation from the diagonal and smoother crash probability patterns that respond more clearly to actual market crashes. This suggests that shorter historical windows retain more relevant and timely predictive information for crash detection.

3.4.3 Models and Features Comparison

Figure A13 shows the actual crash periods from March to May 2022, as well as near-crash period from August to September 2022, which serve as key reference points for evaluating model predictions.

To streamline model identification, we adopt the naming convention “A_F_T”, where A refers to the model architecture (Static Logit, Dynamic Logit, CNN, LSTM), F specifies the feature type used (Market, Sentiment, and Combined), and T represents the window length in days (5, 22, 66, and 132). For example, "Dynamic_Logit_Combined_22" refers to a Dynamic Logistic Regression model using both market and sentiment features over a 22-day window.

Static Logistic Regression models (Figure 2) utilizing sentiment and combined feature sets demonstrate robust performance with AUC scores of 0.81 and 0.83, respectively. The temporal analysis reveals that predicted crash probabilities exhibit notable increases that coincide with observed market distress periods in April, May and August 2022, with the combined feature model (Static_Logit_Combined_22) showing marginally superior temporal accuracy in crash prediction timing.

The results for Dynamic Logistic Regression models (Figure 3) indicate comparable discriminatory performance with AUC scores of 0.81 and 0.83 for sentiment and combined features, nearly identical to their static counterparts. While both approaches show a reasonable crash probability estimates, the dynamic models exhibit slightly higher probability estimates during the August 2022 crash period, suggesting enhanced sensitivity to temporal market dynamics during this particularly volatile episode. However, the performance trade-offs between static and dynamic approaches reveal distinct characteristics (Table 1). The static model achieves higher sensitivity (TPR=0.86) compared to the dynamic model (TPR=0.60), but the dynamic approach demonstrates superior specificity (TNR=0.88 vs. 0.75) and overall accuracy (ACC=0.86 vs. 0.76). Notably, the dynamic model exhibits a significantly lower noise-to-signal ratio (NSR=0.20 vs. 0.29), indicating more reliable crash predictions with fewer false alarms, albeit at the cost of missing more actual crash events.

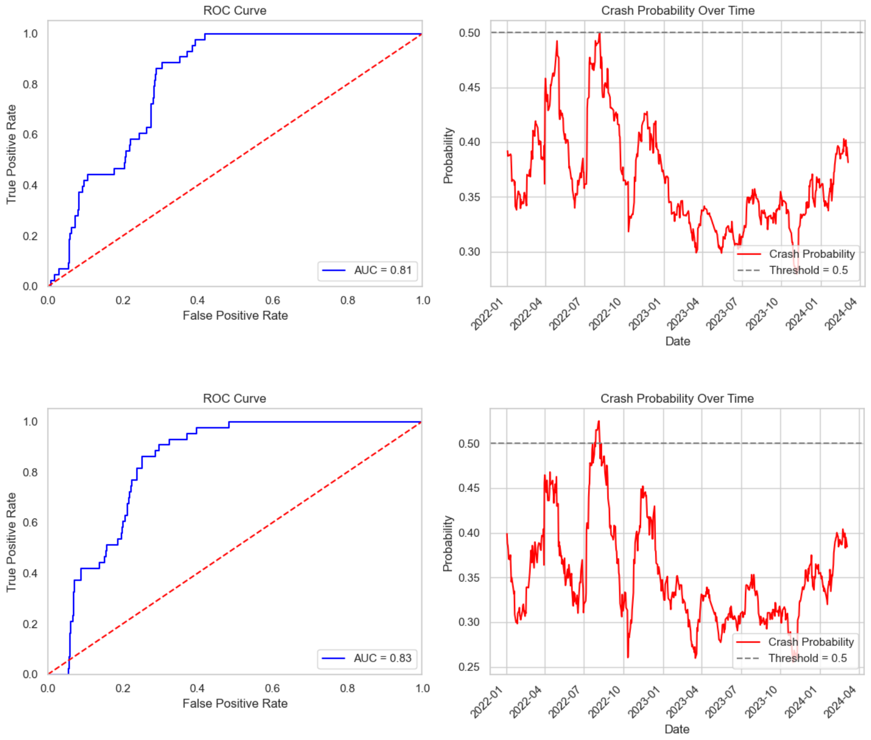

Figure 2: ROC curves (left) and crash probability plots (right) for Static_Logit_Sentiment_22 (top) and Static_Logit_Combined_22 (bottom).

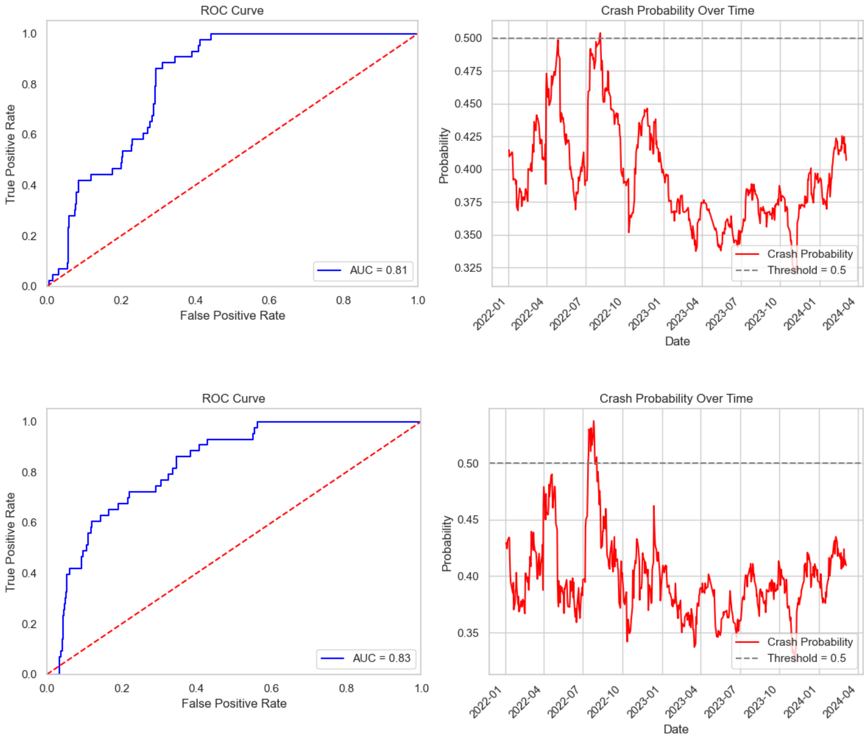

Figure 3: ROC curves (left) and crash probability plots (right) for Dynamic_Logit_Sentiment_22 (top) and Dynamic_Logit_Combined_22 (bottom).

Table 1: Performance Comparison of Logit Models

Model	**Evaluation Metrics**			
Static_Logit_Combined_22	TPR=0.8605	TNR=0.7474	FPR=0.2525	FNR=0.1395
	PPV=0.2284	FOR=0.0160	NSR=0.2935	ACC=0.7565
Dynamic Logit_Combined_22	TPR=0.6047	TNR=0.8808	FPR=0.1192	FNR=0.3953
	PPV=0.3059	FOR=0.0375	NSR=0.1971	ACC=0.8587

As shown in Figure 4, CNN_Market_22 utilizing exclusively market-based features achieve an AUC of 0.79. The temporal analysis exhibits significant crash probability elevations during March, May, and August 2022, as well as early 2023, demonstrating consistent predictive alignment across multiple market distress episodes. While sentiment-based features achieve the highest discriminatory performance with an AUC of 0.83 and pronounced crash probability peaks during May and September 2022, the combined feature model yields a reduced AUC of 0.71 but demonstrates superior crash probability discrimination with minimal false positive rates outside identified crash windows.

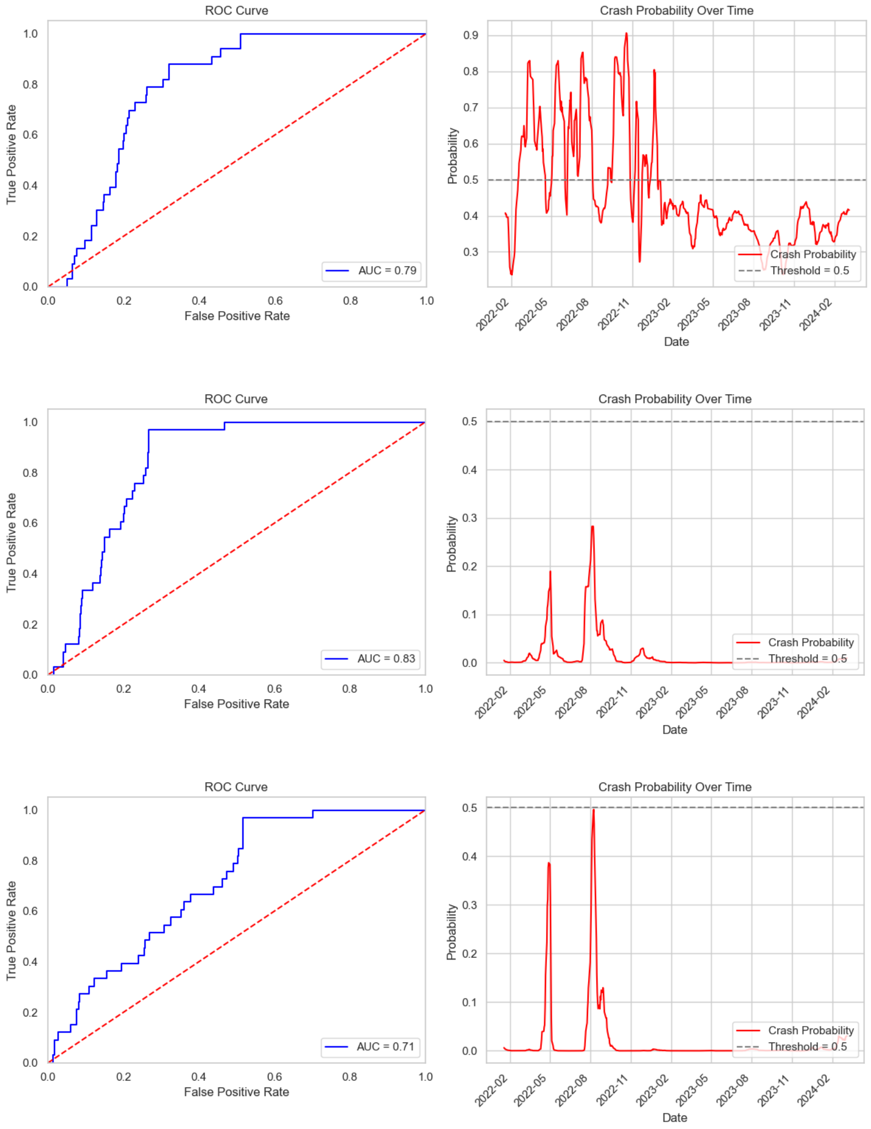

Figure 4: ROC curves (left) and crash probability plots (right) for CNN_Market_22 (top), CNN_Sentiment_22 (middle) and CNN_Combined_22 (bottom).

Figure 5 reveals that LSTM models achieve relatively elevated AUC values of 0.86 and 0.84 for market and sentiment features, respectively. However, the temporal crash probability distributions demonstrate limited interpretability, with probability estimates consistently remaining below 0.1 threshold levels, reducing their practical utility for crash prediction timing.

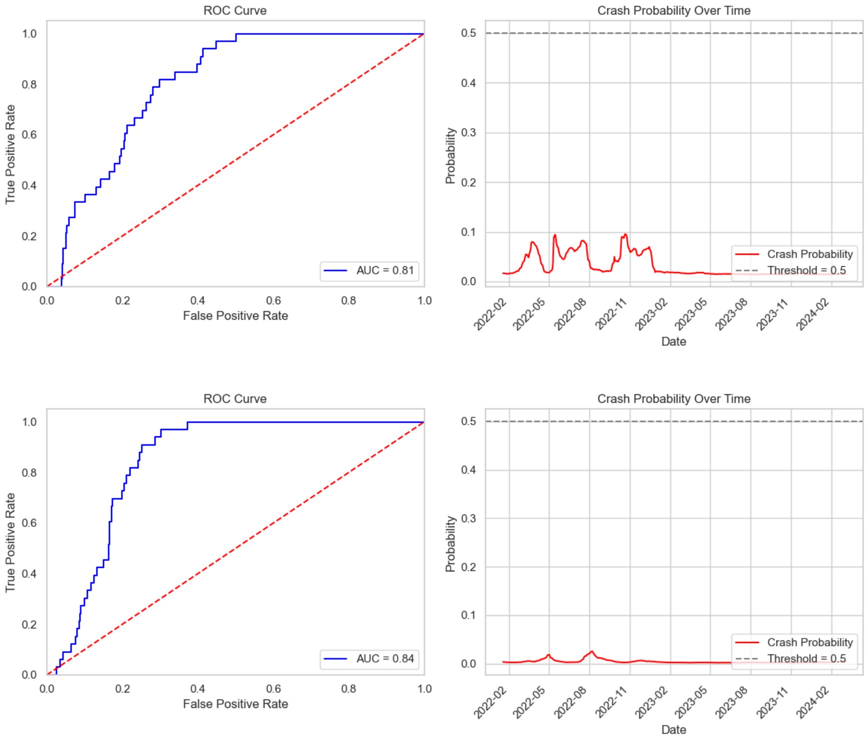

Figure 5: ROC curves (left) and crash probability plots (right) for LSTM_Market_22 (top) and LSTM_Sentiment_22 (bottom).

Despite exhibiting reduced discriminatory performance with an AUC of 0.73, the LSTM model incorporating combined features with a 5-day temporal window configuration shows promise (Figure 6). The temporal crash probability analysis reveals distinctive probability elevations coinciding with primary market distress periods, indicating underlying predictive potential despite diminished overall classification performance.

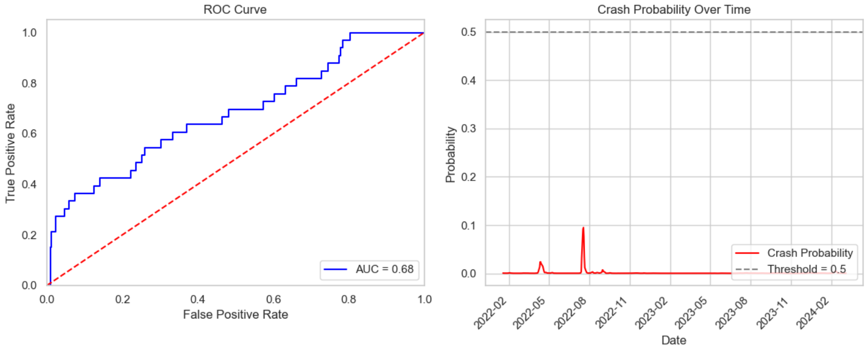

Figure 6: ROC curves (left) and crash probability plots (right) for LSTM_Combined_5.

3.4.4 Ranking the Models

The performance metrics of the compared models are summarized in Table 1, where T denotes the window length (in days) used for feature aggregation and prediction.

Table 2: Performance of Models Filtered by NSR and FNR Thresholds

Model	Evaluation Metrics				Score
CNN_Sentiment_22	TPR=0.9394	TNR=0.7232	FPR=0.2768	FNR=0.0606	0.7223
	PPV=0.1845	FOR=0.0056	NSR=0.2946	ACC=0.7367	
Static_Logit_Combined_22	TPR=0.8605	TNR=0.7474	FPR=0.2525	FNR=0.1395	0.6779
	PPV=0.2284	FOR=0.0160	NSR=0.2935	ACC=0.7565	
LSTM_Sentiment_22	TPR=0.6667	TNR=	FPR=	FNR=	0.5482
	PPV=	FOR=	NSR=	ACC=	
Dynamic Logit_Combined_22	TPR=0.6047	TNR=	FPR=	FNR=	0.5361
	PPV=	FOR=	NSR=	ACC=	
LSTM_Market_22	TPR=	TNR=	FPR=	FNR=	0.4728
	PPV=	FOR=	NSR=	ACC=	

The models were retained only if they met two thresholds: NSR of 0.34 or less, as recommended by Kaminsky (1998), and FNR of 0.5 or less, ensuring that the model correctly captures the majority of actual crash events. These constraints are critical for maintaining both practical utility and predictive reliability, especially in financial or policy-sensitive applications where missed crises and excessive false alarms carry significant costs.

After applying these criteria, only 5 out of 48 models remained. The highest-ranked model by the custom weighted scoring function was the CNN_Sentiment_22 with a score of 0.7223, a high TPR at 0.9394, and an NSR of 0.2946, indicating strong sensitivity with controlled false alarms. Following closely was the Static_Logit_Combined_22 with a score of 0.6779, combining strong TPR (0.8605) and balanced specificity. Other retained models included LSTM_Sentiment_22, Dynamic_Logit_Combined_22, and LSTM_Market_22 models, all demonstrating acceptable FNRs and low NSRs while maintaining competitive scores (ranging from 0.4728 to 0.5482).

3.4.5 Results and Empirical interpretation

Answer research questions

3.5 Out of Sample Analysis?

- Test the goodness of the previous model

<Figure below is in style “figure” which continues to style “figure caption” when you press Enter and then back to “Normal” when you press Enter again.>

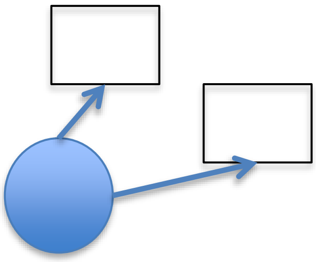

Figure 1: Some important shapes.

<If you wanted to show any code fragments, you could use the following style called code, which could then be followed by figure caption..>

```
# This is a little bit of Python
for i in range( 10 ):
  for j in range( 10 ):
    print i*j,
  print
```

Figure 2: A crucial algorithm for the project.

Chapter 4 Conclusion

Main conclusions of your project. Here you should also include suggestions for future work.

Appendix A Graphical Evaluation of Models Across Window Sizes

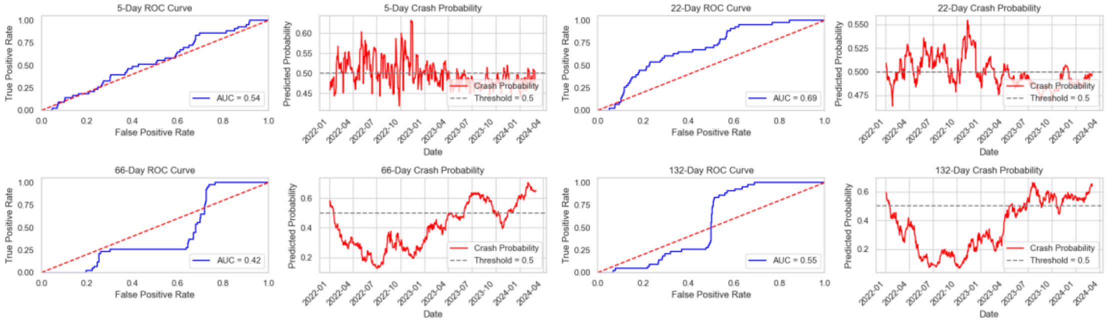

Figure A1: ROC curves and crash probability plots for Static Logit Regression with Market features across window sizes 5, 22, 66, and 132.

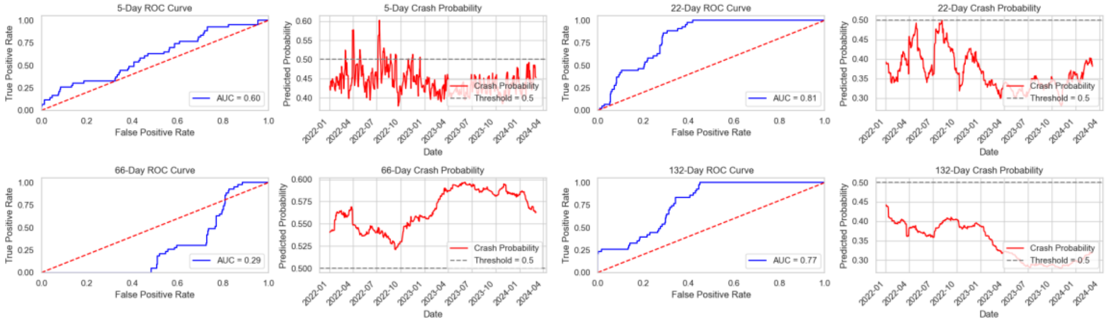

Figure A2: ROC curves and crash probability plots for Static Logit Regression with Sentiment features across window sizes 5, 22, 66, and 132.

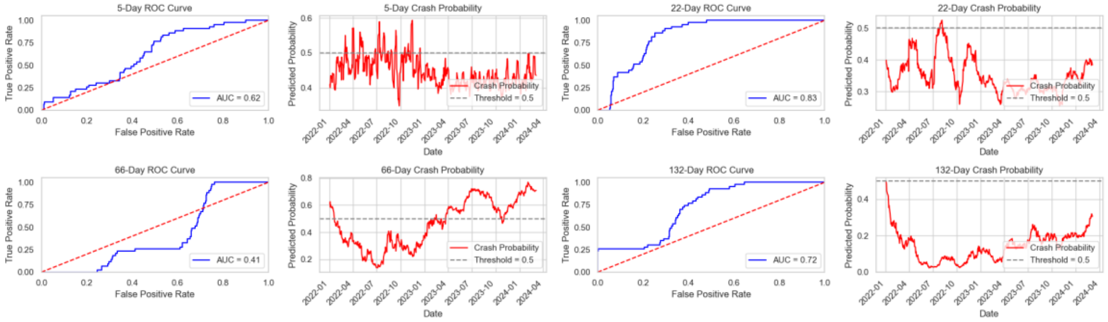

Figure A3: ROC curves and crash probability plots for Static Logit Regression with Combined features across window sizes 5, 22, 66, and 132.

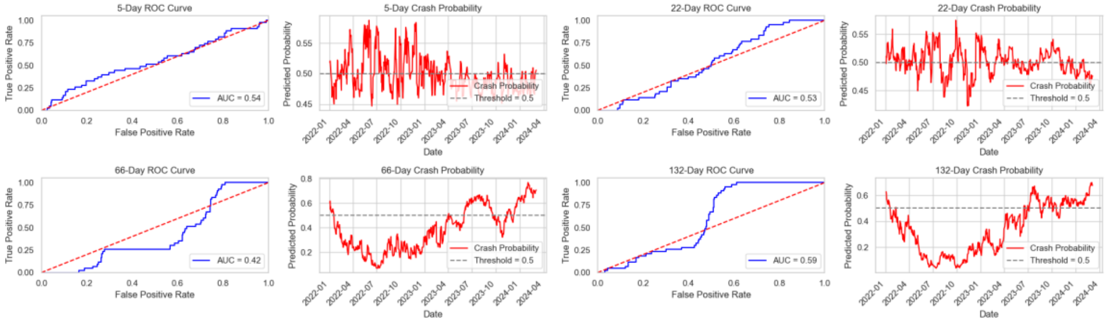

Figure A4: ROC curves and crash probability plots for Dynamic Logit Regression with Market features across window sizes 5, 22, 66, and 132.

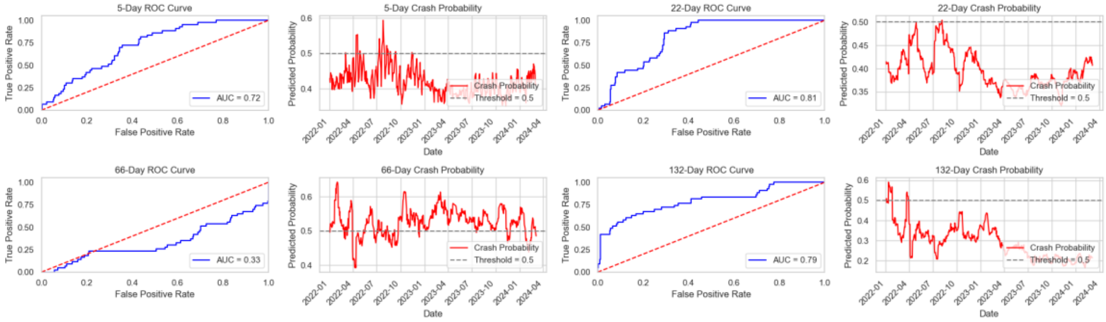

Figure A5: ROC curves and crash probability plots for Dynamic Logit Regression with Sentiment features across window sizes 5, 22, 66, and 132.

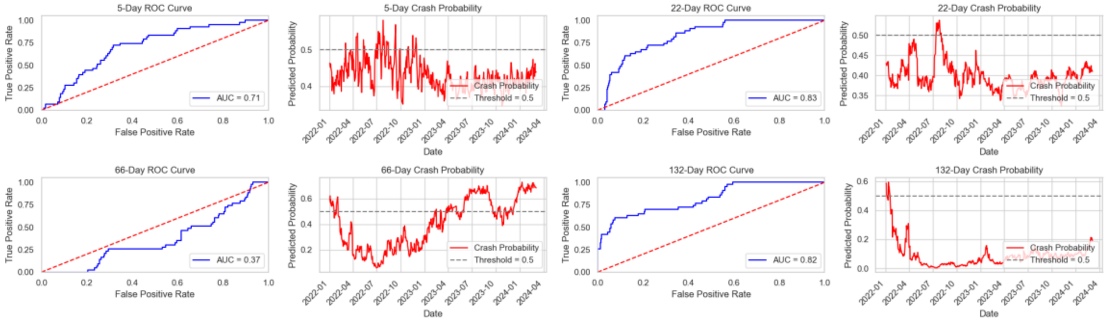

Figure A6: ROC curves and crash probability plots for Dynamic Logit Regression with Combined features across window sizes 5, 22, 66, and 132.

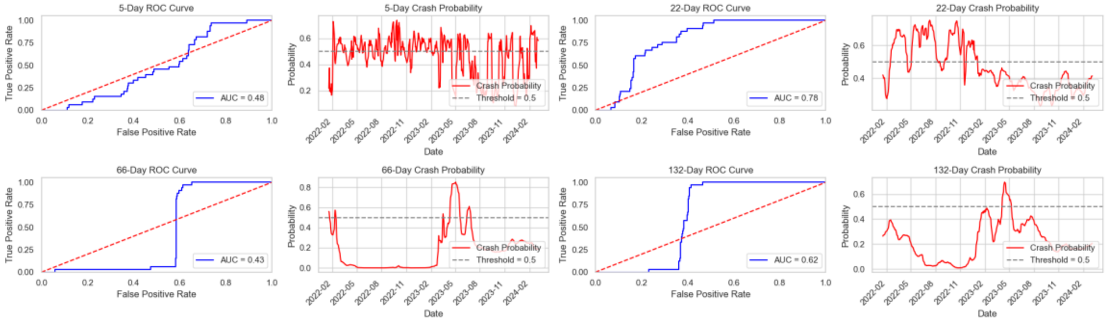

Figure A7: ROC curves and crash probability plots for CNN with Market features across window sizes 5, 22, 66, and 132.

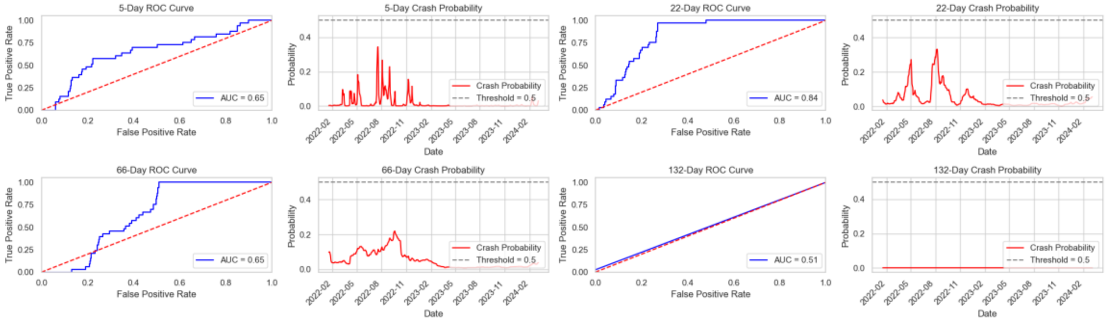

Figure A8: ROC curves and crash probability plots for CNN with Sentiment features across window sizes 5, 22, 66, and 132.

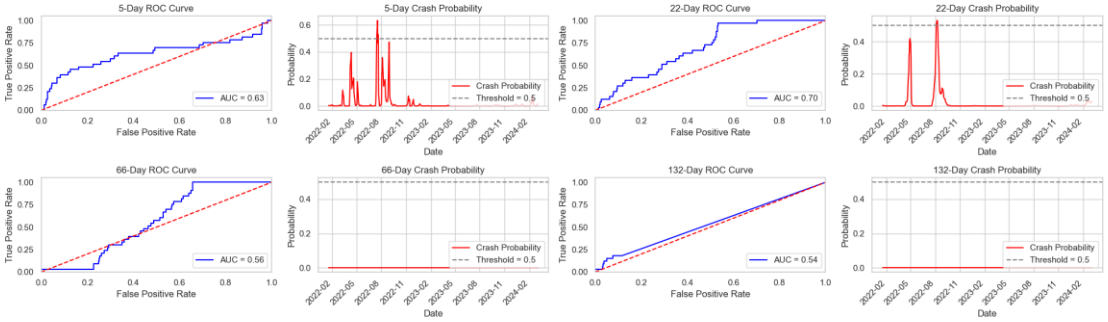

Figure A9: ROC curves and crash probability plots for CNN with Combined features across window sizes 5, 22, 66, and 132.

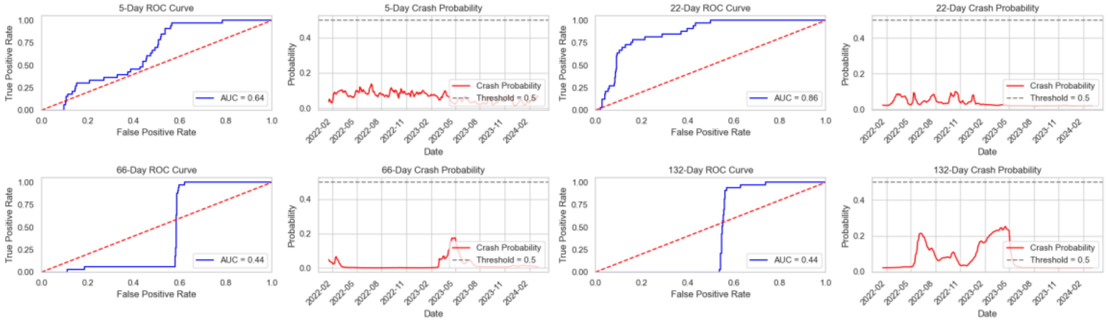

Figure A10: ROC curves and crash probability plots for LSTM with Market features across window sizes 5, 22, 66, and 132.

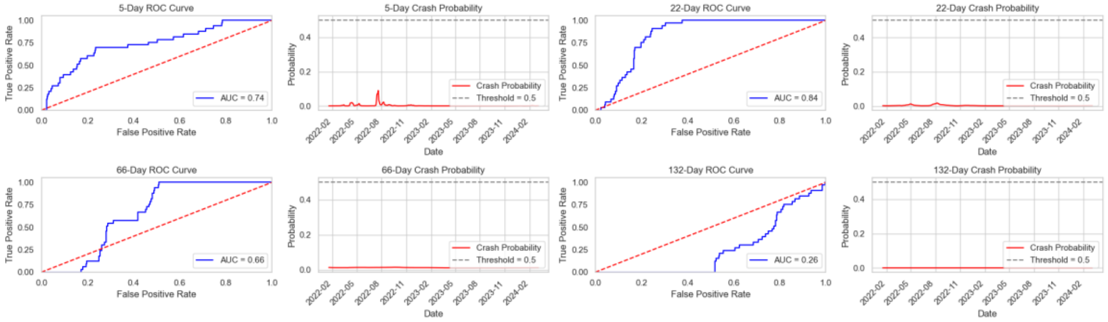

Figure A11: ROC curves and crash probability plots for LSTM with Sentiment features across window sizes 5, 22, 66, and 132.

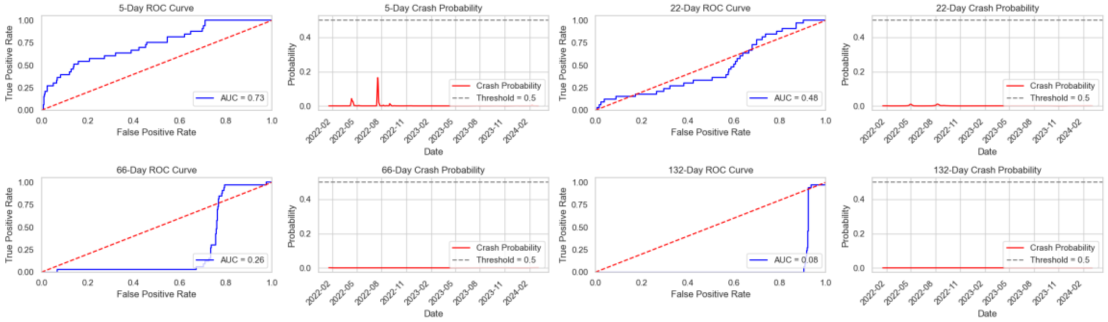

Figure A12: ROC curves and crash probability plots for LSTM with Combined features across window sizes 5, 22, 66, and 132.

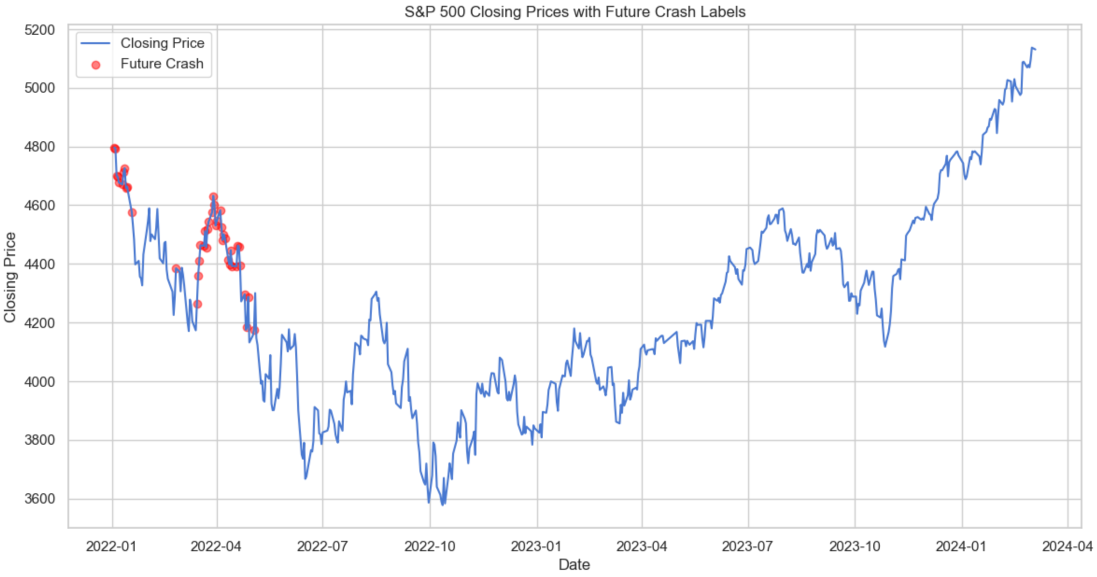

Figure A13: S&P500 closing prices with future crash labels from 01/01/2022 to 04/03/2024

Table A1: Performance of all models

Model	**Evaluation Metrics**				**Score**
CNN_Sentiment_22	TPR=0.9394	TNR=0.7232	FPR=0.2768	FNR=0.0606	0.7223
	PPV=0.1845	FOR=0.0056	NSR=0.2946	ACC=0.7367	
Static_Logit_Combined_22	TPR=0.8605	TNR=0.7474	FPR=0.2525	FNR=0.1395	0.6779
	PPV=0.2284	FOR=0.0160	NSR=0.2935	ACC=0.7565	
LSTM_Sentiment_22	TPR=0.6667	TNR=	FPR=	FNR=	0.5482
	PPV=	FOR=	NSR=	ACC=	
Dynamic Logit_Combined_22	TPR=0.6047	TNR=	FPR=	FNR=	0.5361
	PPV=	FOR=	NSR=	ACC=	
LSTM_Market_22	TPR=	TNR=	FPR=	FNR=	0.4728
	PPV=	FOR=	NSR=	ACC=	

Appendix B Evaluation Metrics

Bibliography

[1] Kustina, L., Sudarsono, R., & Effendi, N. (2023). Market crash factors and developing an early warning system: Evidence from Asia. *Investment Management and Financial Innovations, 20*(3), 116–125. https://doi.org/10.21511/imfi.20(3).2023.10

[2] Song, S., & Li, H. (2024). Early warning signals for stock market crashes: Empirical and analytical insights utilizing nonlinear methods. *EPJ Data Science, 13*, 16. https://doi.org/10.1140/epjds/s13688-024-00457-2

[3] Liu, J., Leu, J., & Holst, S. (2023). Stock price movement prediction based on Stocktwits investor sentiment using FinBERT and ensemble SVM. *PeerJ Computer Science, 9*, e1403. https://doi.org/10.7717/peerj-cs.1403

[4] Allaj, E., & Sanfelici, S. (2023). Early warning systems for identifying financial instability. *International Journal of Forecasting, 39*(4), 1777–1803. https://doi.org/10.1016/j.ijforecast.2022.08.004

[5] Park, M., Peterson, M., & Weisbrod, E.H. (2024). Top-Down vs. Bottom-Up Index Forecasts: Are Market Strategists Strategically Pessimistic? *SSRN Electronic Journal*. http://dx.doi.org/10.2139/ssrn.4695279

[6] Le, T.H. (2024). Forecasting value-at-risk and expected shortfall in emerging market: does forecast combination help? *Journal of Risk Finance, 25*(1), 160-177. https://doi.org/10.1108/JRF-06-2023-0137

[7] Huang, A. H., Wang, H., & Yang, Y. (2020). FinBERT – A large language model for extracting information from financial text. *SSRN Electronic Journal*. http://dx.doi.org/10.2139/ssrn.3910214

[8] Kaminsky, G., Lizondo, S., & Reinhart, C. M. (1998). Leading indicators of currency crises. *IMF Staff Papers, 45*(1), 1–48 https://EconPapers.repec.org/RePEc:pal:imfstp:v:45:y:1998:i:1:p:1-48

[9] Kaminsky, G., & Reinhart, C. M. (1999). The twin crises: The causes of banking and balance-of-payments problems. *American Economic Review, 89*(3), 473–500. https://doi.org/10.1257/aer.89.3.473

[10] Jemović, M., & Marinković, S. (2019). Determinants of financial crises—An early warning system based on panel, logit regression. *International Journal of Finance & Economics, 24*(4), 1–15. https://doi.org/10.1002/ijfe.1779

[11] Bussière, M., & Fratzscher, M. (2006). Towards a new early warning system of financial crises. *Journal of International Money and Finance, 25*(6), 953–973. https://doi.org/10.1016/j.jimonfin.2006.07.007.

[12] Parras-Gutiérrez, E., Rivas, V. M., García-Arenas, M., & del Jesús, M. J. (2014). Short-, medium- and long-term forecasting of time series using the L-Co-R algorithm. *Neurocomputing, 128*, 433–446. https://doi.org/10.1016/j.neucom.2013.08.023

[13] Budhidharma, Valentino & Sembel, Roy & Hulu, Edison & Ugut, Gracia. (2023). Early warning signs of financial distress using random forest and logit model. *Corporate and Business Strategy Review, 4*, 69-88. https://doi.org/10.22495/cbsrv4i4art8

[14] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

[15] Box, G.E.P., Jenkins, G.M. and Reinsel, G.C. (2008). Time Series Analysis: Forecasting and Control. 4th Edition, *Wiley, Oxford*. https://doi.org/10.1002/9781118619193

[16] Hiew J. Z. G., Huang X., Mou H., Li D., Wu Q., Xu Y. (2019). BERT-based financial sentiment index and LSTM-based stock return predictability. *Statistical Finance [q-fin.ST]*, ArXiv preprint. arXiv:1906.09024. https://doi.org/10.48550/arXiv.1906.09024

[17] Bonde, G. Khaled R. (2012). Extracting the best features for predicting stock prices using machine learning. *Institute of Artificial Intelligence, University Of Georgia Athens, GA-30601*, https://api.semanticscholar.org/CorpusID:252608319

[18] Bollerslev T. (1986), Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics, 31*(3), 307-327, https://doi.org/10.1016/0304-4076(86)90063-1

[19] Engle R. F., & Manganelli, S. (2004). CAViaR: Conditional Autoregressive Value at Risk by Regression Quantiles. *Journal of Business & Economic Statistics, 22*(4), 367–381. https://doi.org/10.1198/073500104000000370

[20] Andreou C., Andreou P., Lambertides N. (2019). Financial Distress Risk and Stock Price Crashes. *Journal of Corporate Finance*, https://doi.org/10.2139/ssrn.3450075

[21] Lander G. P., The Sarbanes ‐ Oxley Act of 2002 (2002). *Journal of Investment Compliance*, *3*(1), 44‑53. https://doi.org/10.1108/joic.2002.3.1.44

[22] Hällman, L. (2017). The Rolling Window Method: Precisions of Financial Forecasting. *Matematisk statistic, KTH*. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-205595

[23] Defazio, A., Bach, F.R., & Lacoste-Julien, S. (2014). SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. *Neural Information Processing Systems*. https://doi.org/10.48550/arXiv.1407.0202

[24] Osborne M. F. M. (1959). Brownian Motion in the Stock Market. *Operations Research 7*(2), 145-173. https://doi.org/10.1287/opre.7.2.145

[25] Ren L., Ren P. (2017). Testing the market efficiency by mean absolute deviation. Benchmarking, *An International Journal, 24*(7), 2049–2062. https://doi.org/10.1108/BIJ-06-2016-0096

[26] Gaies B., Nakhli M. S., Ayadi R., Sahut J. (2022) Exploring the causal links between investor sentiment and financial instability: A dynamic macro-financial analysis, *Journal of Economic Behavior & Organization, 204*, 290-303, https://doi.org/10.1016/j.jebo.2022.10.013.