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

1st September 2025

Abstract

This study investigates the role of market-based and sentiment-based volatility indicators in predicting financial crashes and develops a dynamic early warning system (EWS) to improve crisis detection. The research evaluates 48 models, comparing traditional static logit approaches with dynamic architectures including dynamic logit, CNN, and LSTM, across multiple volatility windows. Models were filtered based on false negative rate ($FNR \leq 0.5$) and noise-to-signal ratio ($NSR \leq 0.34$), resulting in 6 models that met both criteria. The CNN model using sentiment features along with 22-day volatility window achieved the highest weighted score ($Score = 0.8971$) and true positive rate ($TPR = 0.9394$), demonstrating strong sensitivity while maintaining reasonable specificity. The analysis shows that sentiment-based indicators provide anticipatory signals that complement market-based volatility, enhancing early crash detection. While some models with higher specificity exhibited lower sensitivity, prioritizing models with high sensitivity is preferable due to the asymmetric cost of missed crises. An additional finding is that shorter volatility windows (5-day and 22-day) outperform longer windows (66-day and 132-day), suggesting the importance of timely information.

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.

Name: Mokha Lerthsuwanroj Signature: Mokha Lerthsuwanroj

Acknowledgements

I would like to express my gratitude to all those who contributed to this project. Special thanks to my supervisor, Dr. Chris McCaig for guidance, constructive feedback, and encouragement throughout the research. I also appreciate the support of School of Computing Science for providing the necessary resources and research environment.

Contents

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 system (EWS) 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-varying window (e.g., T = 22, 66, 132 days) in the context of EWS for financial

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

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 EWS 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-based and market-based volatility indicators.
3. To evaluate and compare the predictive performance of the proposed dynamic EWS 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:

RQ1: 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?

RQ2: Can a dynamic EWS outperform traditional static models in predicting financial market crashes?

RQ3: To what extent does sentiment volatility enhance the performance of early warning models compared to using market-based indicators alone?

RQ4: Is the proposed dynamic EWS 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. Close: 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. Data collected from 02/01/2008 to 31/12/2021 formed the training dataset, while

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

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 Selected Time Horizons for Volatility Analysis

In this study, both volatility is evaluated using four distinct rolling window lengths, each representing a different market time horizon: one week (5 trading days), one month (22 trading days), one quarter (66 trading days), and half a year (132 trading days).

3.1.5 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 over an n-day window, scaled by the square root of per-year horizon h. This scaling converts the daily volatility estimate to an annualized figure, under the assumption that daily 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 a given time horizon scales linearly with the length of that horizon. Since standard deviation is the square root of variance, the standard deviation over h days scales with $\sqrt{h}$. In the case of annualized volatility, where the horizon is taken to be one trading year with approximately 252 trading days ($h = 252$), the annualized n-day market volatility at day t is defined as:

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

$$MV_t^{(n)} = \sqrt{252} \times \sqrt{\frac{1}{n}\sum_{j=t-n+1}^{t}(\bar{R}_j - \bar{R}_t^{(n)})^2}$$

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

3.1.6 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, 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 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.

3.1.7 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.

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.

3.1.8 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:

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

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:

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

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

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

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

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:

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

where $||\beta||_1 = \sum_j |\beta|_j$ is the L1 norm, $||\beta||_2^2 = \sum_j |\beta|_j^2$ is the squared L2 norm, $\alpha \in [0, 1]$ 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

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 as provided in Appendix B, Table B.1

3.2.3 Convolutional Neural Network (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. This model was implemented using the TensorFlow framework[5].

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[6] to penalize misclassification of the minority crash class. Lastly, a dynamic thresholding method was also applied to the CNN outputs.

3.2.4 Long Short-Term Memory (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. This model was implemented using the TensorFlow framework[7].

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 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.

3.3.1 Confusion Matrix

The confusion matrix provides a tabular representation of classification outcomes by comparing predicted labels with actual labels, consist of True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN). Most of performance metrics derived from the confusion matrix, these include True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Precision (PPV), False Omission Rate (FOR), Noise-to-Signal Ratio (NSR), and Accuracy (ACC). A detailed explanation of each metric, along with

[5] https://www.tensorflow.org/tutorials/images/cnn
[6] https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html
[7] https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

its formula and relevance in the EWS context, is provided in Appendix B, Table B.1.

Among these, NSR is particularly crucial in the context of EWS. It measures the level of false alarms relative to correct crash predictions, offering a direct signal quality indicator. According to Kaminsky (1998), an NSR below 0.34 indicates that the model generates meaningful signals with relatively few false alarms. Conversely, an NSR above 1.0 suggests the model produces more noise than signal, making it unreliable for practical warning systems.

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 EWSs, 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. The weight values used in the scoring are listed in Appendix B2.

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).

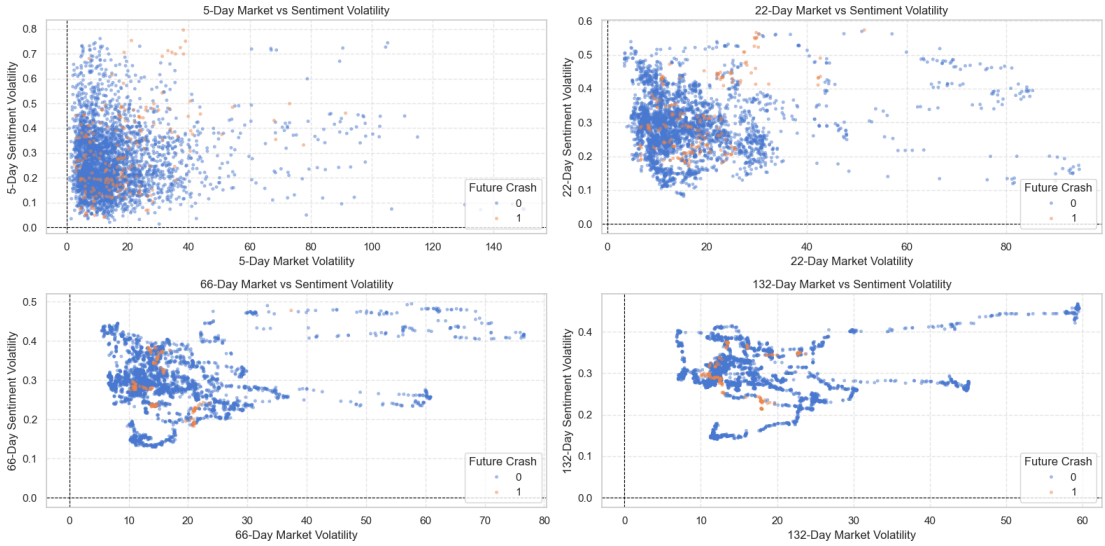


Figure 1: Relationship between n-day market volatility and n-day sentiment volatility for each volatility period

3.4.2 Window Size Analysis

Examination in Appendix A reveals that for window sizes 66 and 132, model performance significantly declines. The ROC curves for these configurations are generally close to or below the diagonal, especially for the market features (Figure A.1, A.4, A.7, A.10) and all LSTM models (Figure A.10-12). 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.

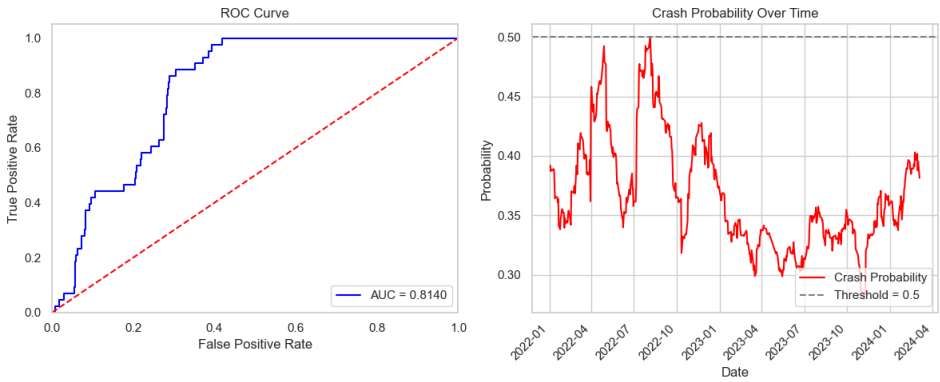
3.4.3 Performance Comparison

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 0.8140 and 0.8283, 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.

(a)



(b)

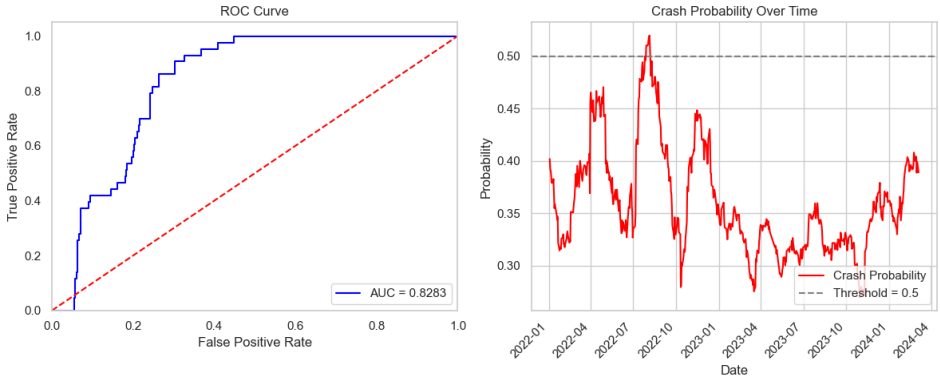
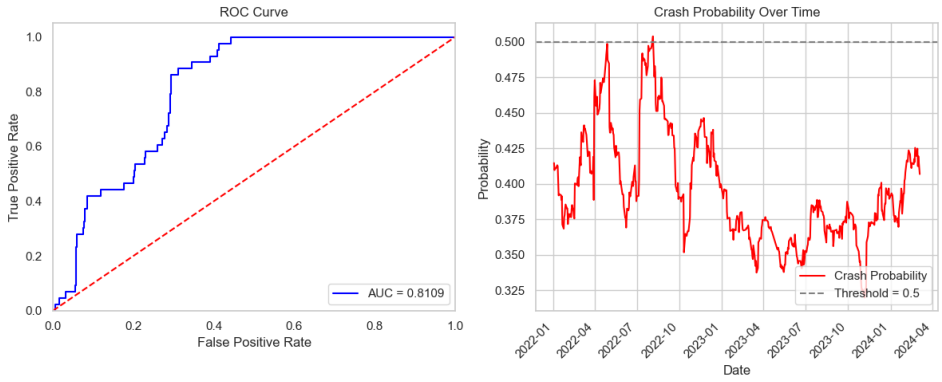


Figure 2: ROC curves and crash probability plots for (a) Static_Logit_Sentiment_22 and (b) Static_Logit_Combined_22.

The results for Dynamic Logistic Regression models (Figure 3) indicate comparable discriminatory performance with AUC scores of 0.8109 and 0.8339 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 compared to the dynamic model (TPR = 0.8605 vs. 0.6279), but the dynamic approach demonstrates superior specificity (TNR = 0.7374 vs. 0.8828) and overall accuracy (ACC = 0.7472 vs. 0.8625). Notably, the dynamic model exhibits a significantly lower noise-to-signal ratio (NSR = 0.3052 vs. 0.1866), indicating more reliable crash predictions with fewer false alarms, albeit at the cost of missing more actual crash events.

(a)



(b)

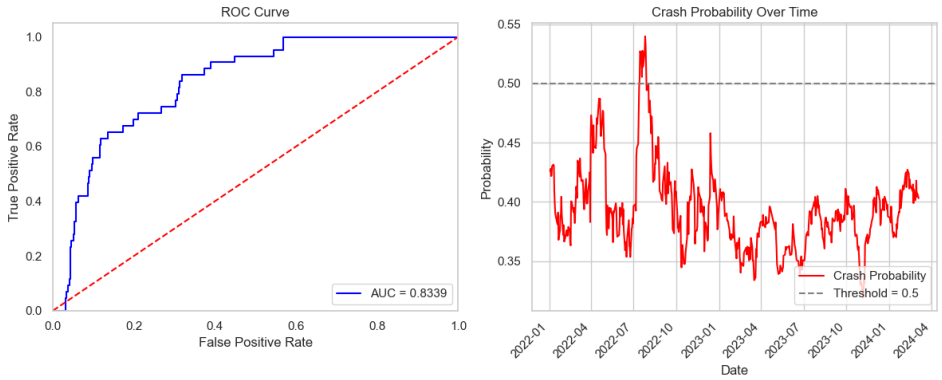


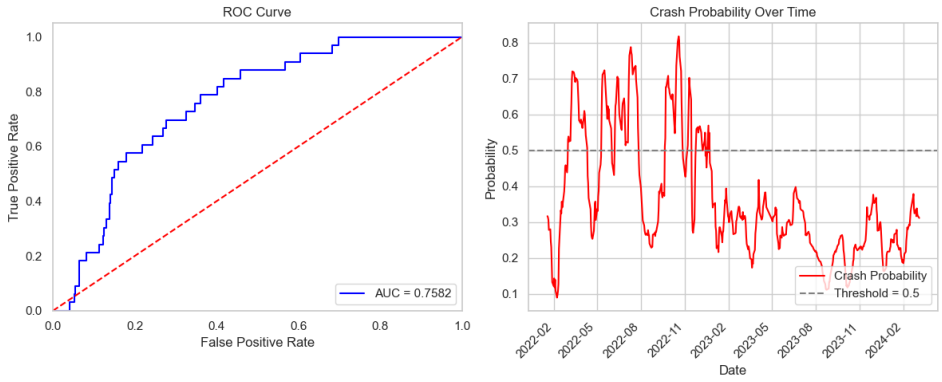
Figure 3: ROC curves and crash probability plots for (a) Dynamic_Logit_Sentiment_22 and (b) Dynamic_Logit_Combined_22.

Table 1: Performance comparison of Static_Logit_Combined_22 and Dynamic_Logit_Combined_22

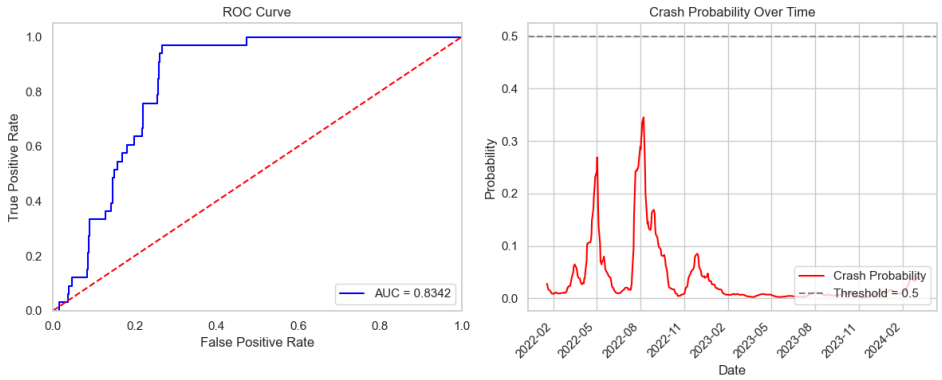
Model	**Evaluation Metrics**			
Static_Logit_Combined_22	TPR=0.8605	TNR=0.7374	FPR=0.2626	FNR=0.1395
	PPV=0.2216	FOR=0.0162	NSR=0.3052	ACC=0.7472
Dynamic_Logit_Combined_22	TPR=0.6279	TNR=0.8828	FPR=0.1172	FNR=0.3721
	PPV=0.3176	FOR=0.0352	NSR=0.1866	ACC=0.8625

As shown in Figure 4, CNN_Market_22 utilizing exclusively market-based features achieve an *AUC* of 0.7582. The temporal analysis exhibits significant crash probability elevations during March to July 2022, as well as around October 2022 to early 2023, shows consistent predictive alignment across multiple market distress episodes. While sentiment-based features achieve the highest discriminatory performance with an *AUC* of 0.8342 and pronounced crash probability peaks during May and September 2022, the combined feature model yields a reduced *AUC* of 0.6578 but demonstrates superior crash probability discrimination with minimal false positive rates outside identified crash windows.

(a)



(b)



(c)

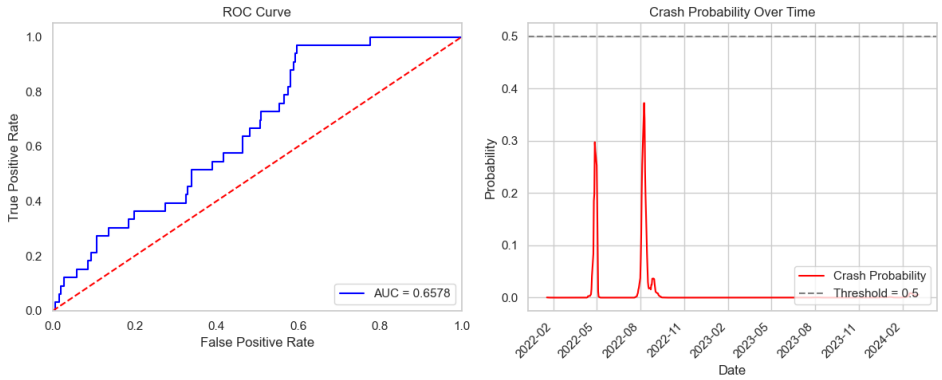


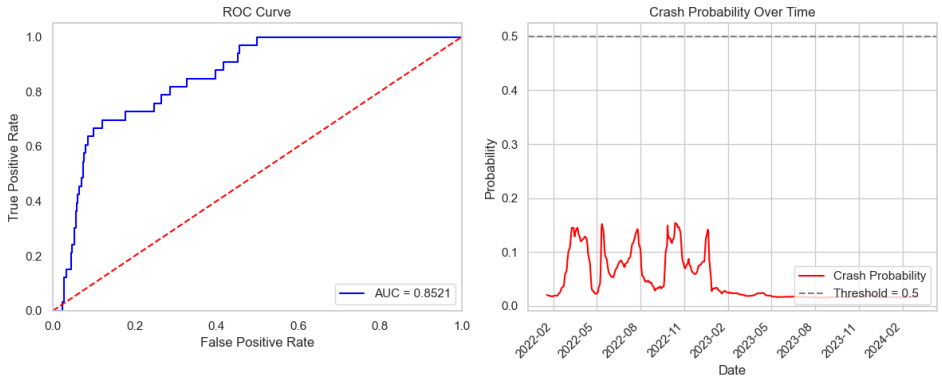
Figure 4: ROC curves and crash probability plots for (a) CNN_Market_22, (b) CNN_Sentiment_22 and (c) CNN_Combined_22

Figure 5 reveals that LSTM models achieve relatively elevated AUC values of 0.8521 and 0.8443 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.

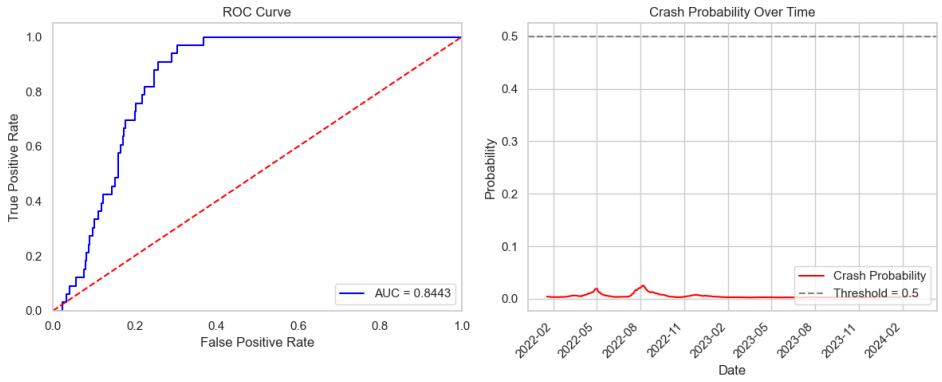
Despite exhibiting reduced discriminatory performance with an AUC of 0.7414, the LSTM model incorporating combined features with a 5-day temporal window

configuration shows promise. The temporal crash probability analysis reveals distinctive probability elevations coinciding with primary market distress periods, indicating underlying predictive potential despite diminished overall classification performance.

(a)



(b)



(c)

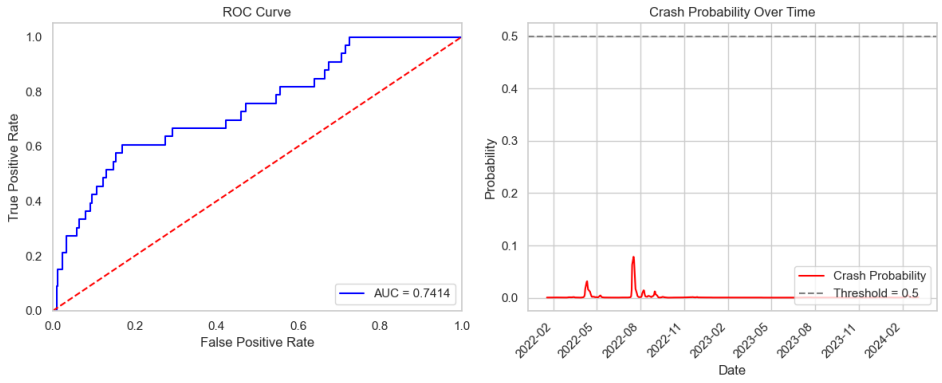


Figure 5: ROC curves and crash probability plots for (a) LSTM_Market_22, (b) LSTM_Sentiment_22, and (c) LSTM_Combined_5.

3.4.4 Features Comparison

Figure 7-10 present the feature importance rankings across four model architectures. Sentiment volatility emerges as a highly influential feature in short to medium time windows (especially 5 and 22 days), particularly in models capable of capturing nonlinear or temporal relationships such as CNN and LSTM. Conversely, as the window size increases to 66 and 132 days, market volatility and related risk metrics gain greater importance, particularly in more traditional models like Static and Dynamic Logit Regression.

Additionally, the dynamic models (those incorporating lags) highlight the usefulness of lagged volatility indicators, especially lagged market volatility, in improving predictive accuracy during rapidly evolving market conditions.

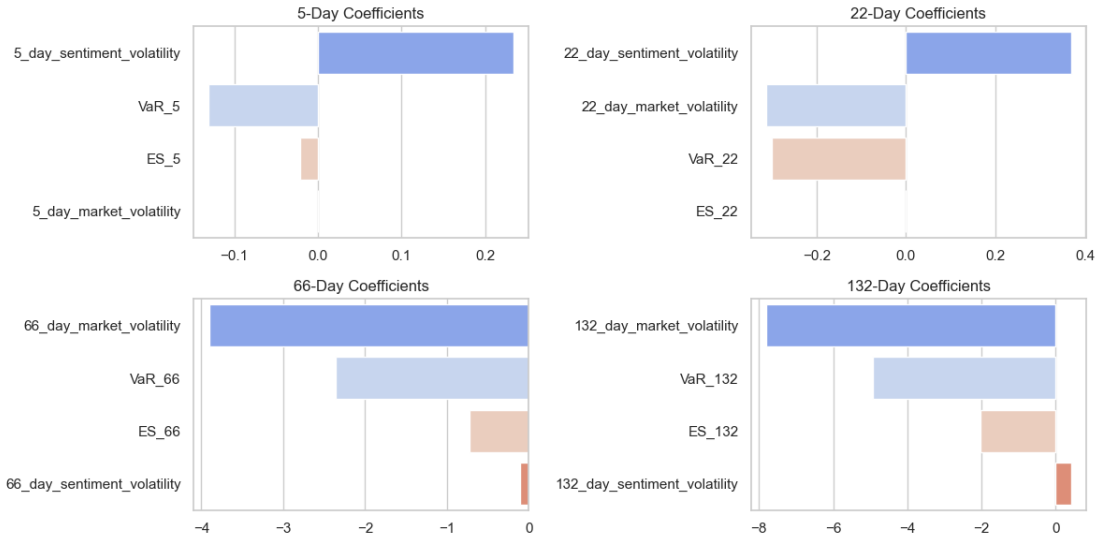


Figure 7: Feature importance plots for Static Logit Regression with Combined features across window sizes 5, 22, 66, and 132.

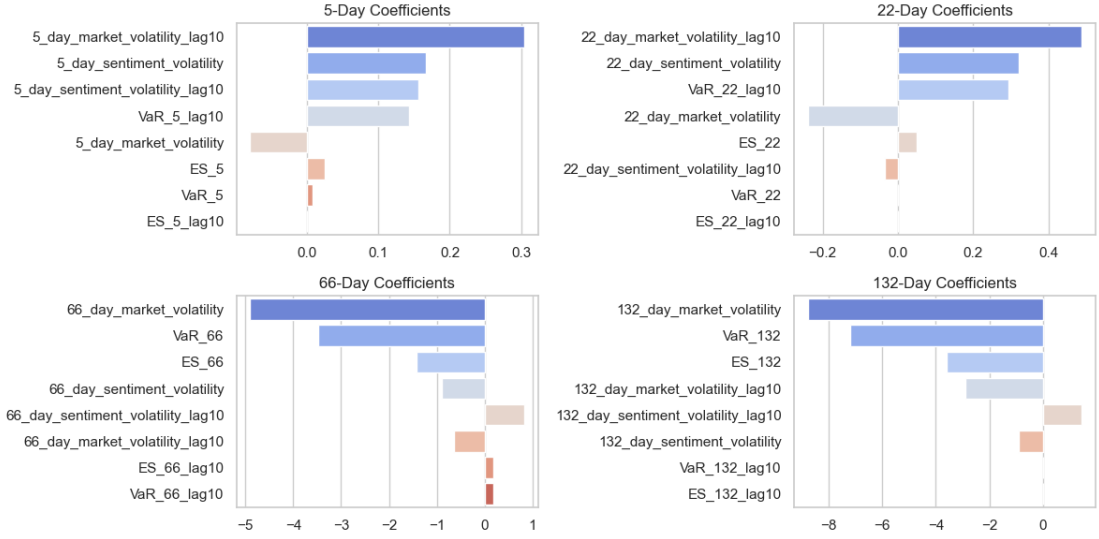


Figure 8: Feature importance plots for Dynamic Logit Regression with Combined features across window sizes 5, 22, 66, and 132.

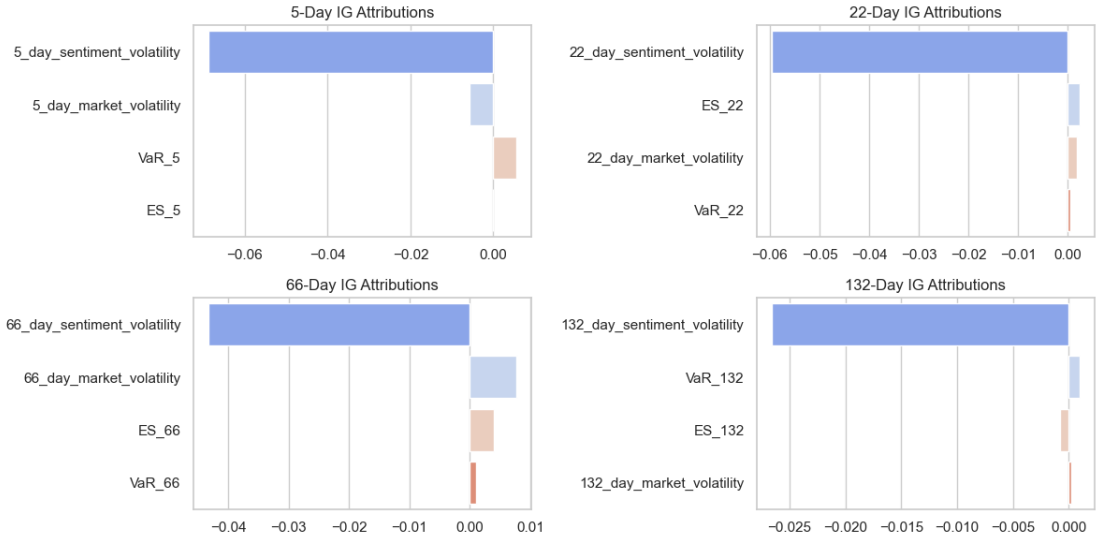


Figure 9: Feature importance plots for CNN with Combined features across window sizes 5, 22, 66, and 132.

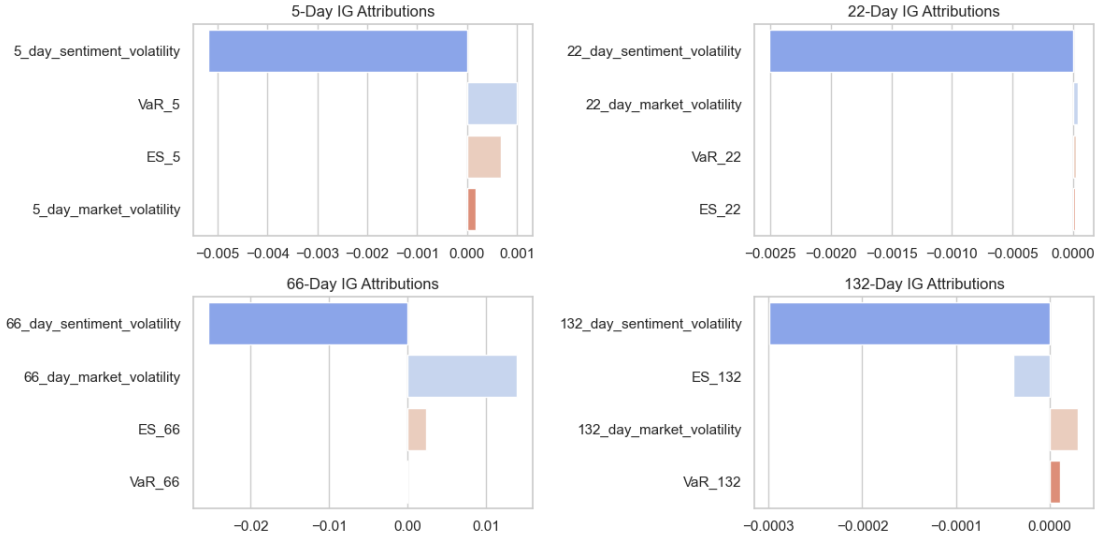


Figure 10: Feature importance plots for LSTM with Combined features across window sizes 5, 22, 66, and 132.

3.4.5 Ranking the Models

The performance metrics of the compared models are summarized in Table 2. 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.

After applying both thresholds, 6 out of 48 models were retained. The highest-ranked model by the custom weighted scoring function was CNN_Sentiment_22 with a score of 0.8971, demonstrating very high *TPR* (0.9394) and moderate *TNR* (0.7373), along with a controlled *NSR* (0.2839). Close behind, The Static_Logit_Combined_22 scored 0.8310, showing solid *TPR* (0.8605) with moderate *TNR* (0.7374) and *NSR* (0.3052).

Other retained models included LSTM_Market_22, Dynamic_Logit_Combined_22, LSTM_Sentiment_22, and CNN_Market_22 exhibited acceptable FNR and low NSR, with competitive accuracy, although their sensitivity was lower compared to the top models.

An important observation is that models with higher TPR generally had slightly higher NSR, while models with lower NSR tended to trade off sensitivity for specificity. Lastly, all retained models were based on a 22-period window size, suggesting that shorter windows are generally more effective for meeting the NSR and FNR thresholds.

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

Model	Evaluation Metrics				Score
CNN_Sentiment_22	$TPR = 0.9394$	$TNR = 0.7373$	$FPR = 0.2667$	$FNR = 0.0606$	0.8971
	$PPV = 0.1902$	$FOR = 0.0055$	$NSR = 0.2839$	$ACC = 0.7462$	
Static_Logit_Combined_22	$TPR = 0.8605$	$TNR = 0.7374$	$FPR = 0.2216$	$FNR = 0.1395$	0.8310
	$PPV = 0.2216$	$FOR = 0.0162$	$NSR = 0.3052$	$ACC = 0.7472$	
LSTM_Market_22	$TPR = 0.6061$	$TNR = 0.9152$	$FPR = 0.0848$	$FNR = 0.3939$	0.7385
	$PPV = 0.3226$	$FOR = 0.0279$	$NSR = 0.1400$	$ACC = 0.8958$	
Dynamic_Logit_Combined_22	$TPR = 0.6279$	$TNR = 0.8828$	$FPR = 0.1172$	$FNR = 0.3721$	0.7321
	$PPV = 0.3176$	$FOR = 0.0353$	$NSR = 0.1866$	$ACC = 0.8625$	
LSTM_Sentiment_22	$TPR = 0.6667$	$TNR = 0.8222$	$FPR = 0.1778$	$FNR = 0.3333$	0.7021
	$PPV = 0.2000$	$FOR = 0.0263$	$NSR = 0.2667$	$ACC = 0.8125$	
CNN_Market_22	$TPR = 0.5152$	$TNR = 0.8404$	$FPR = 0.1596$	$FNR = 0.4848$	0.5646
	$PPV = 0.1771$	$FOR = 0.0370$	$NSR = 0.3098$	$ACC = 0.8201$	

The performance of CNN_Sentiment_22 further demonstrates the effectiveness of sentiment-based features in early crash detection. As shown in Figure 11, the model's predicted crash probabilities (red line) align closely with the actual crash periods (grey areas). Importantly, the model not only captured the possible future crash at the beginning of 2022 but also signalled elevated probabilities during the period from August 2022 to November 2022, when no official future crash label occurred according to our labelling criterion. However, as seen in Figure 12, this period coincided with a sharp market decline, where the S&P 500 dropped significantly before recovering. The data shows that the model was sensitive to heightened systemic risk and market stress, even in near-crash scenarios.

Bringing up this observation, the performance of most models was negatively affected by the strict crash labelling criterion. Since periods like August 2022 to November 2022 were labelled as non-crash despite sharp market declines, models that correctly signalled elevated risk during this time were penalized. We can infer that the binary definition of crashes may limit the ability of models to fully demonstrate their predictive value, particularly in capturing near-crash episodes that are highly relevant from a risk management perspective.

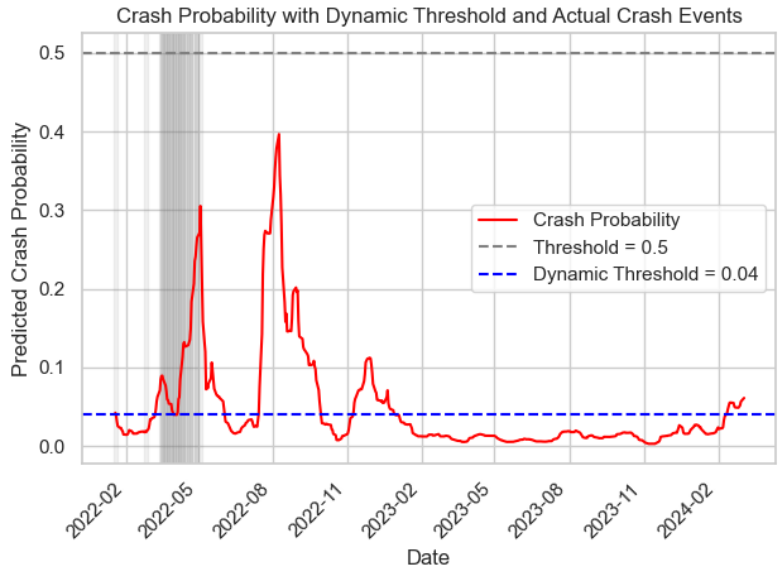


Figure 11: Crash probability with dynamic threshold and actual future crash labels for CNN_Sentiment_22.

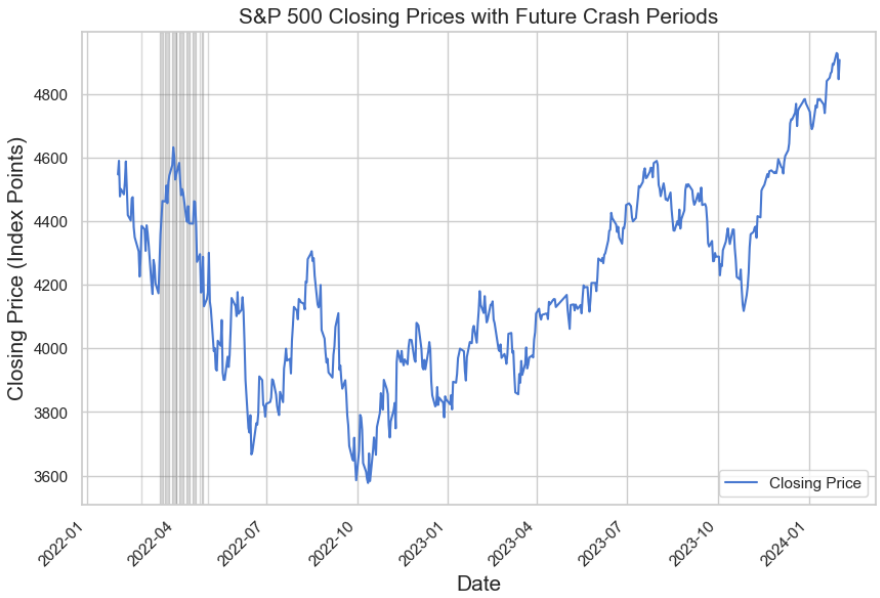


Figure 12: S&P500 closing prices with actual future crash labels.

3.4.6 Research Question Evaluation

RQ1

The analysis of feature importance across models indicates that both market-based and sentiment-based volatility indicators contribute meaningfully to predicting financial crashes (see Figure 7-10). However, their predictive power varies by context and model architecture. Market volatility often showed strong negative importance in static and dynamic models, while lagged market volatility and sentiment volatility exhibited consistently positive contributions. This suggests that while market-based signals capture reactive market behaviour, sentiment-based indicators provide complementary anticipatory signals.

RQ2

The dynamic EWS generally outperforms traditional static logit models in capturing early crash signals, supported by both graphical observations (see Figure 2-6) and a detailed evaluation (see Table 2). The CNN model using sentiment features achieved the highest TPR of 0.9394 and weighted score of 0.8971, reflecting its strong ability to detect crashes while maintaining reasonable specificity. The static logit model with combined features also performed well with a TPR of 0.8605 and a competitive weighted score of 0.8310, suggesting that even traditional models benefit from feature combination. In contrast, although the remain models demonstrated higher specificity than those CNN and static logit models with $TNR > 0.8000$, they suffered from lower TPR, indicating a tendency to miss crash events. Given the context of EWS, where missing a crisis can be more costly than issuing a false alarm, models like CNN with high sensitivity are preferable.

RQ3

Sentiment volatility significantly enhances early warning model performance when compared to market volatility alone. Across various window sizes and model types, sentiment-based features often yielded higher scores and better-aligned crash probability spikes with actual crisis periods. In several cases, models using only sentiment volatility performed comparably or even better than those using combined features (see Figure 2-6 and Table 2), indicating that sentiment captures unique signals not fully reflected in market prices.

RQ4

The proposed dynamic models demonstrate robustness and generalizability across different conditions and periods. This is evident through consistent performance in ROC and crash probability plots across the 2022–2023 timeline (see Figure 2-6). However, performance degrades with longer volatility windows (66 and 132 days), especially in models relying on market-based features, suggesting the importance of choosing appropriate temporal parameters.

Chapter 4 Conclusion

4.1 Discussions

The objective of this project is to investigate the role of market-based and sentiment-based volatility indicators in predicting financial crashes, develop a dynamic EWS, and evaluate its performance against traditional static approaches. The findings demonstrate several important insights.

Market and sentiment volatility are both useful, but they work differently. Market-based volatility tends to capture reactive market behaviour, whereas sentiment-based provide anticipatory signals that often improve early detection. Sentiment indicators pick up on shifts in investor mood and media coverage before those changes fully show up in stock prices, which makes them especially valuable for spotting crashes early.

Model complexity versus simplicity revealed interesting trade-offs. The dynamic framework proved generally more effective than the static ones, with the best results coming from a neural network. However, the fact that a static logit model with combined features still ranked among the top-performing models highlights that model simplicity should not be dismissed. A possible explanation is that crashes often arise from relatively abrupt shifts, and static frameworks can still detect these when provided with the right feature sets. While fancy models can be helpful, having quality inputs might be more important than we initially thought. The performance evaluation also highlighted another trade-off inherent in EWS. Some models exhibited higher specificity, they suffered from lower sensitivity, reflecting a trade-off between avoiding false alarms and missing actual crises. Given that the costs of failing to detect a crash typically outweigh the costs of false alarms, models prioritizing sensitivity are particularly valuable in the EWS context.

Lastly, an important temporal pattern emerged consistently across the performance analysis. Nearly all retained models were based on a 22-day volatility window, with longer windows generally underperforming. This finding aligns with our initial exploratory analysis, where shorter windows (5-day and 22-day) showed 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 consistent superiority of the 22-day window confirms that shorter temporal horizons are more effective for capturing the evolving signals that preceding financial crashes, supporting behavioral finance theories that investor mood and narrative instability often precede actual market crashes.

4.2 Challenges

A key challenge in this study lies in the rarity of crash events. According to the applied criteria, crashes account for only 5.87% of the dataset, making the prediction task inherently imbalanced. This imbalance not only increases the risk of model's overfitting to non-crash periods but also makes it difficult to assess robustness under limited crisis observations.

Another challenge arises from the use of more complex architectures such as CNN and LSTM. While these models captured patterns more effectively than static frameworks, they often produced very low probability outputs with limited confidence levels.

4.3 Future Works

Future research could explore models of intermediate complexity that strike a balance between flexibility and interpretability. While CNN and LSTM architectures are powerful, they often produce very low probability outputs and limited confidence under rare crash conditions. Tree-based ensembles such as XGBoost, or shallower neural networks, may capture nonlinear relationships more effectively than static models without suffering from under confidence.

Expanding the set of input features is another direction. Additional volatility measures or alternative choices, such as implied volatility or dispersion, could provide better signals of market stress and improve early warning performance. Moreover, using more sophisticated volatility estimators, such as Parkinson, Garman-Klass, or Rogers-Satchell, may also enhance model accuracy by better capturing market variability than simple variance-based measures.

Finally, there is significant potential in rethinking how we score and evaluate these models. The reality is that missing an actual crisis costs much more than issuing a false warning, so our evaluation methods should reflect this imbalance. By improving the custom evaluation formula combined with additional relevant metrics and applying more appropriate weightings, we can create assessment tools that better align with the practical realities of early warning systems.

Appendix A Graphical Evaluation of Models

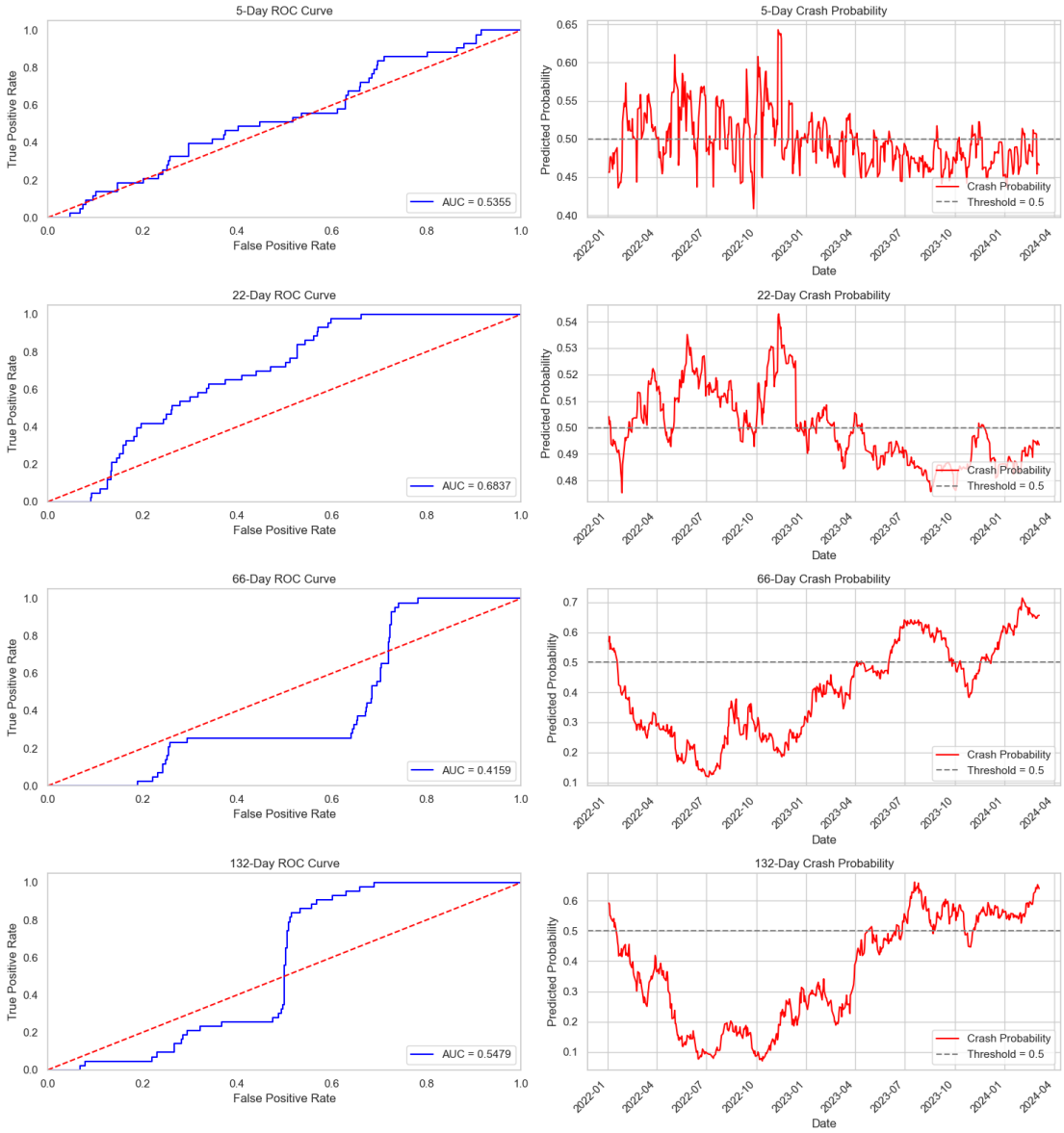


Figure A.1: ROC curves and crash probability plots for Static Logit Regression with Market features across window sizes 5, 22, 66, and 132.

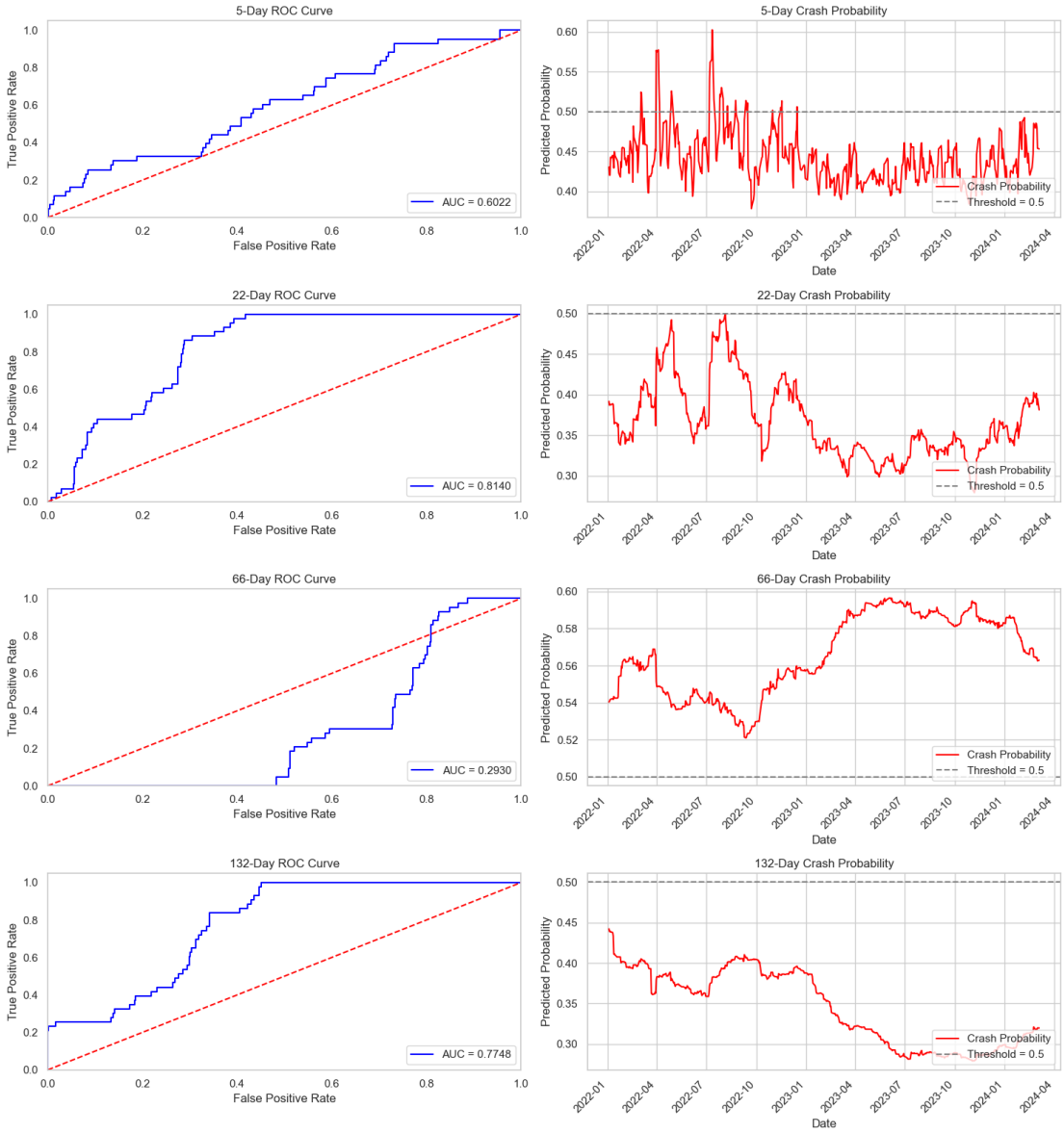


Figure A.2: ROC curves and crash probability plots for Static Logit Regression with Sentiment features across window sizes 5, 22, 66, and 132.

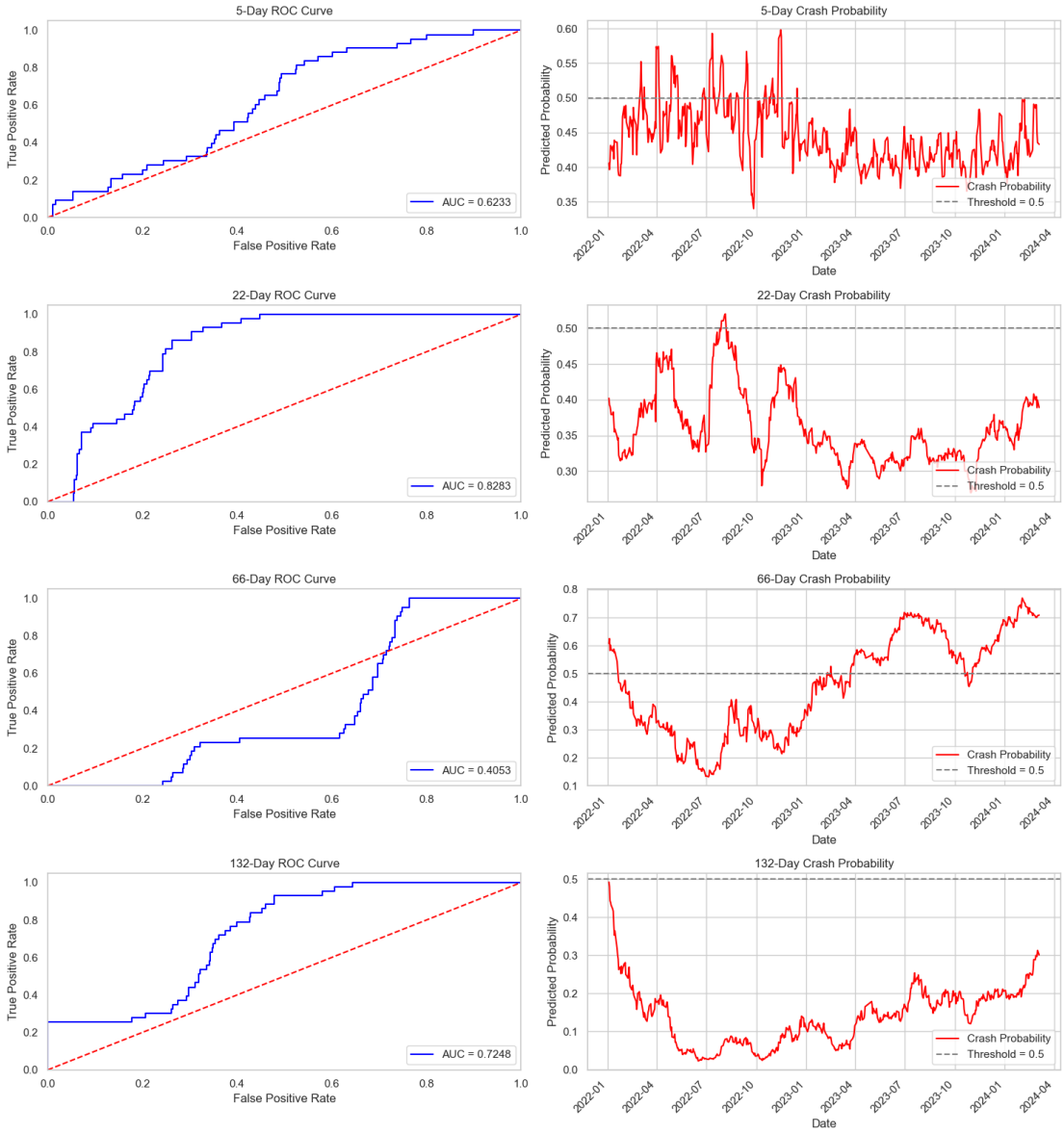


Figure A.3: ROC curves and crash probability plots for Static Logit Regression with Combined features across window sizes 5, 22, 66, and 132.

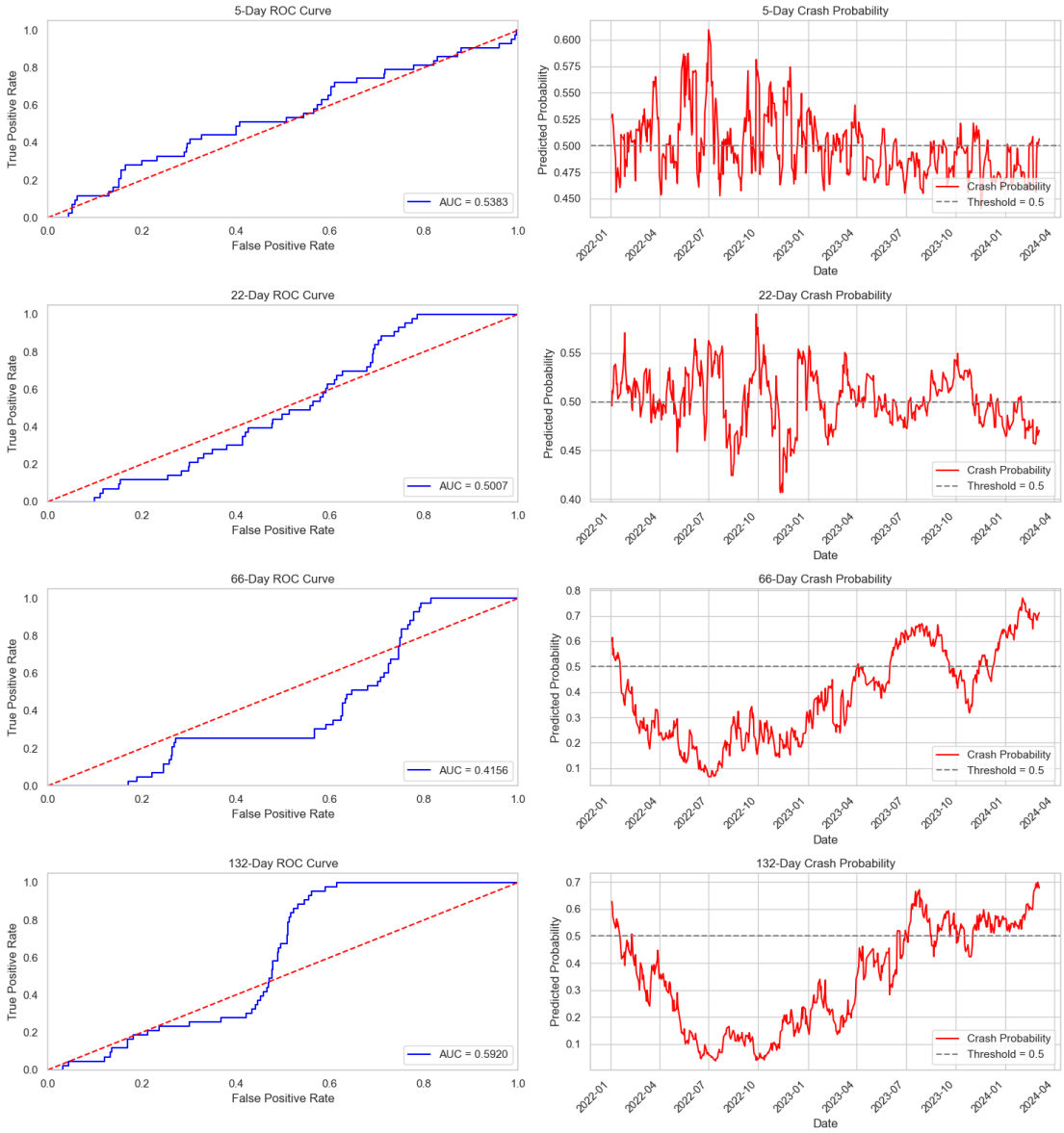


Figure A.4: ROC curves and crash probability plots for Dynamic Logit Regression with Market features across window sizes 5, 22, 66, and 132.

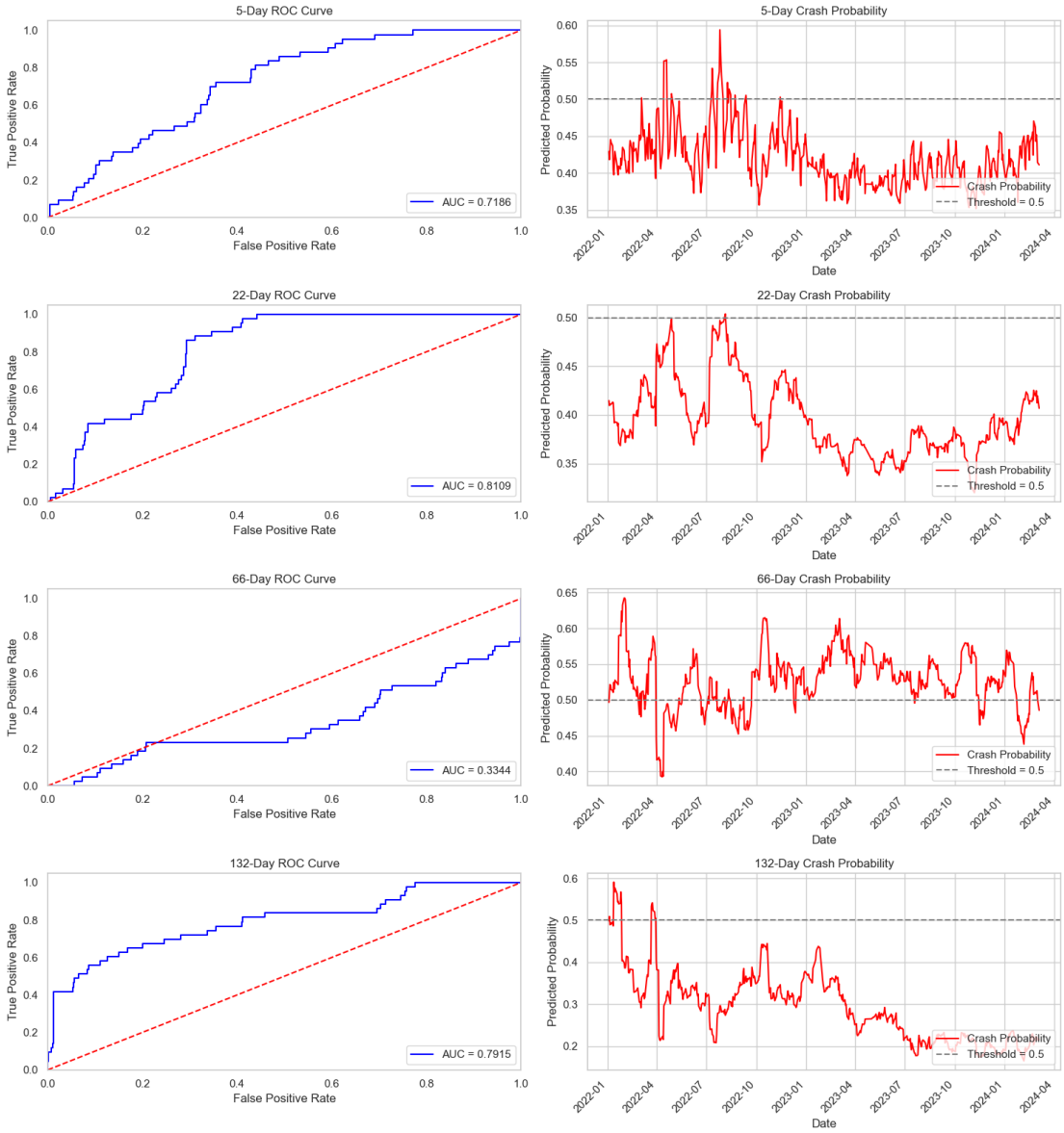


Figure A.5: ROC curves and crash probability plots for Dynamic Logit Regression with Sentiment features across window sizes 5, 22, 66, and 132.

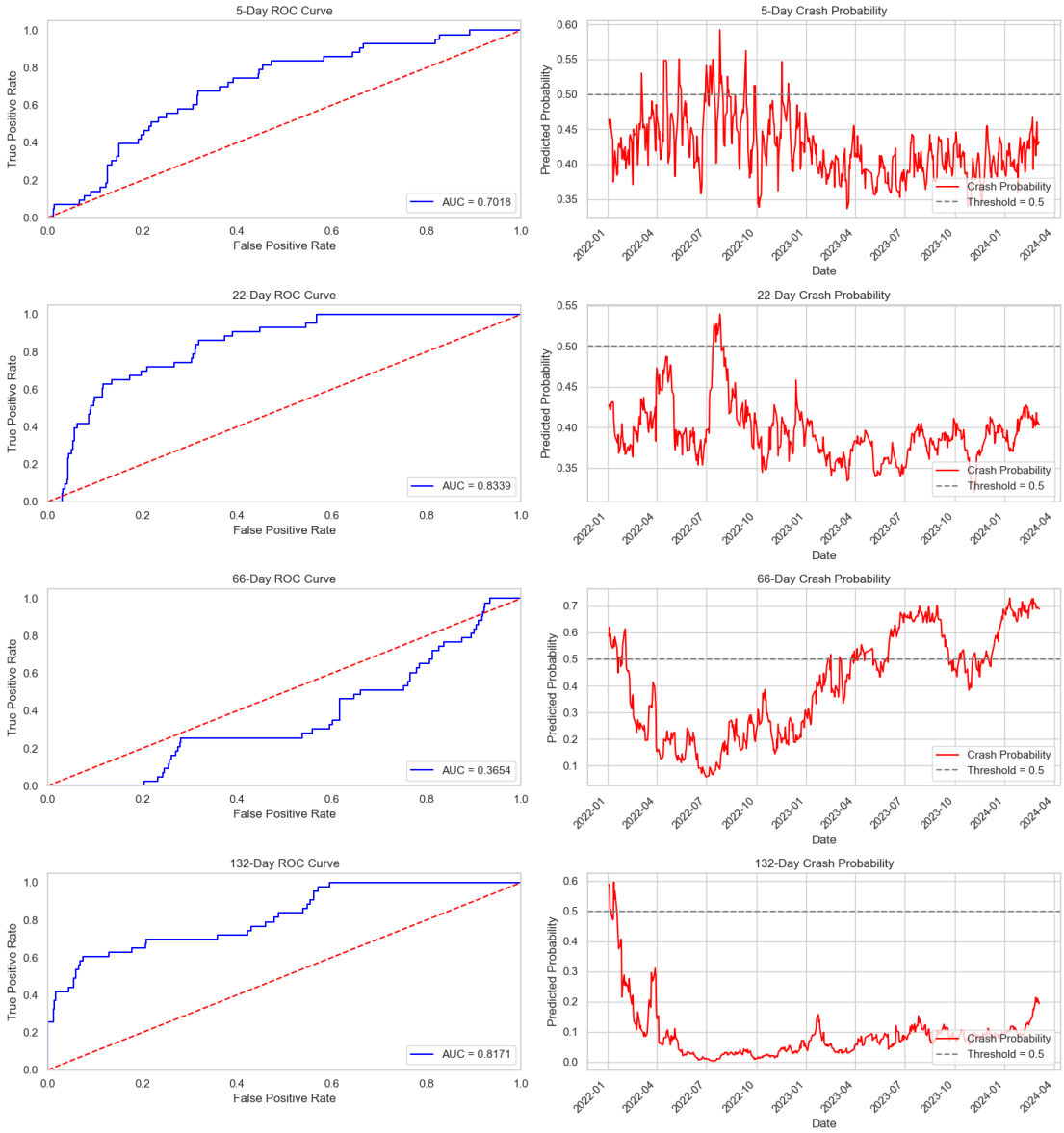


Figure A.6: ROC curves and crash probability plots for Dynamic Logit Regression with Combined features across window sizes 5, 22, 66, and 132.

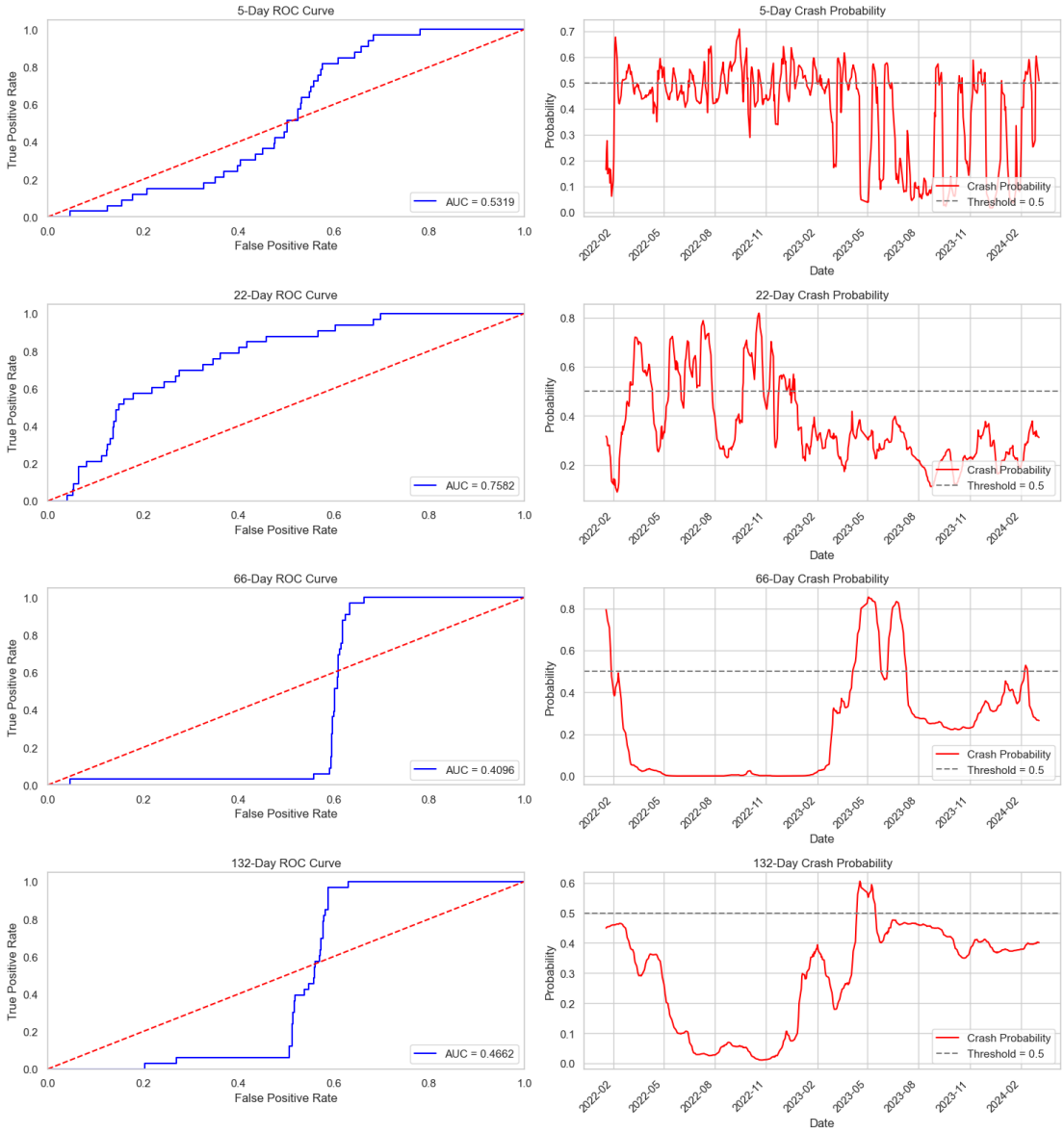


Figure A.7: ROC curves and crash probability plots for CNN with Market features across window sizes 5, 22, 66, and 132.

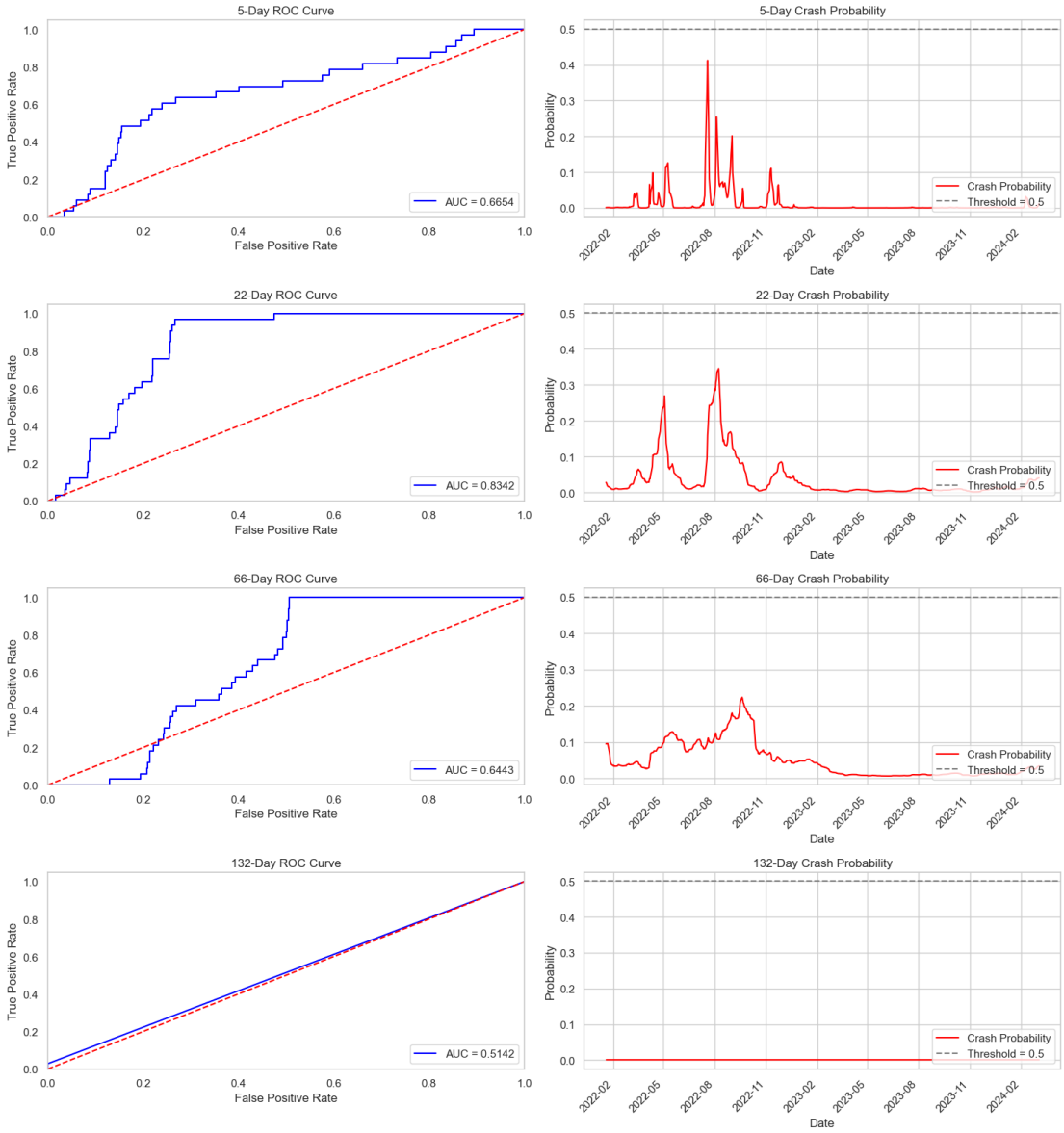


Figure A.8: ROC curves and crash probability plots for CNN with Sentiment features across window sizes 5, 22, 66, and 132.

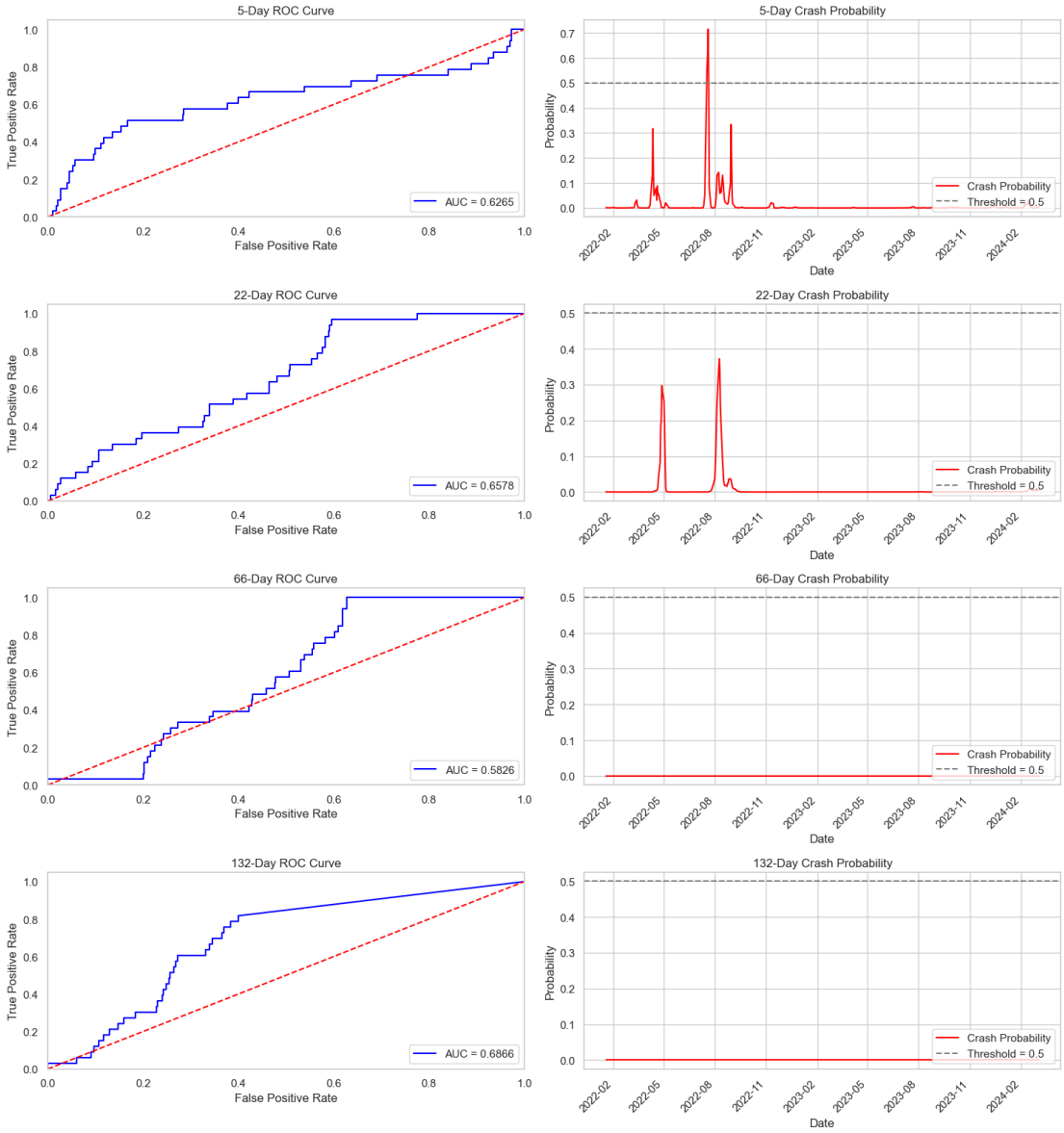


Figure A.9: ROC curves and crash probability plots for CNN with Combined features across window sizes 5, 22, 66, and 132.

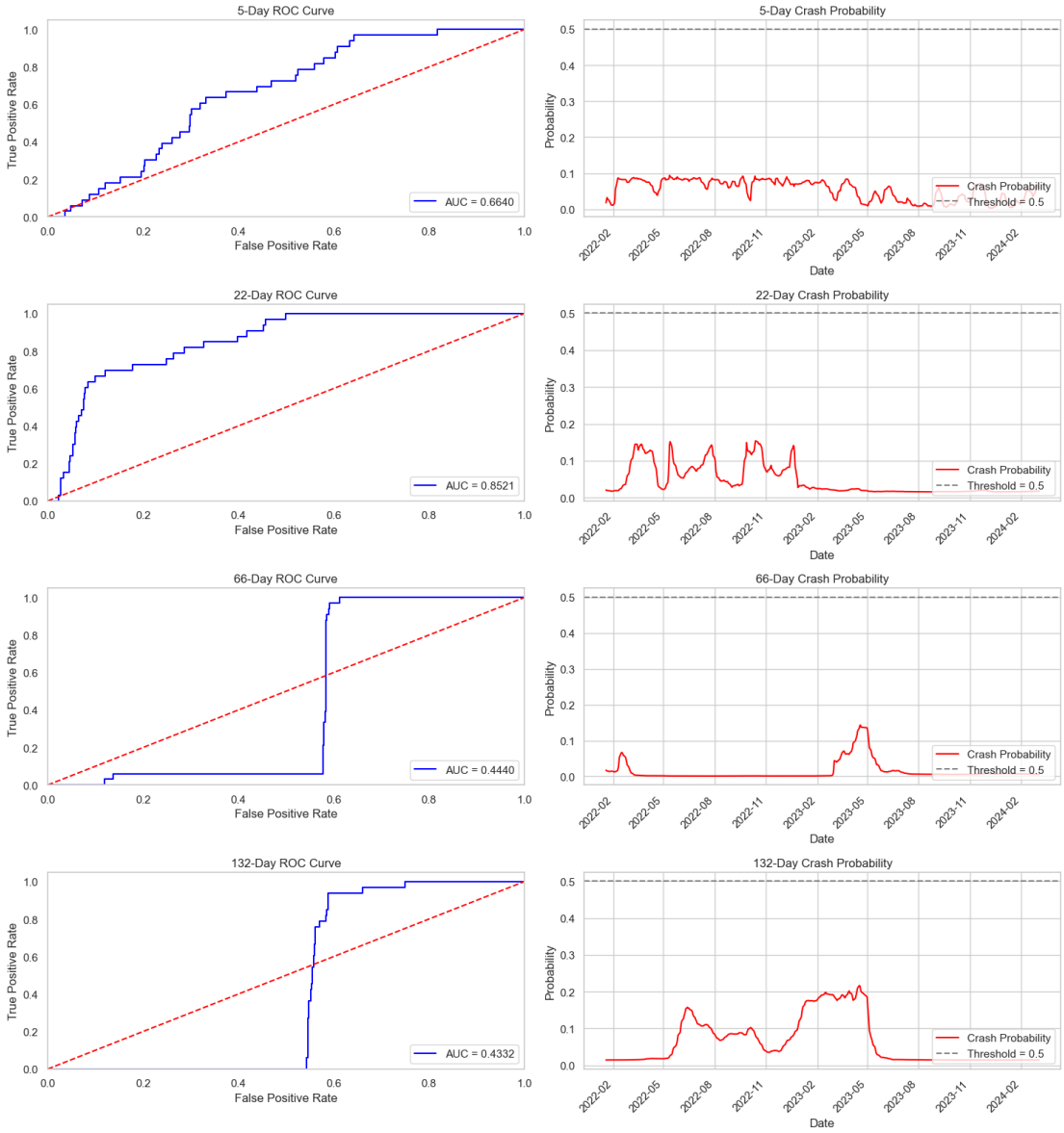


Figure A.10: ROC curves and crash probability plots for LSTM with Market features across window sizes 5, 22, 66, and 132.

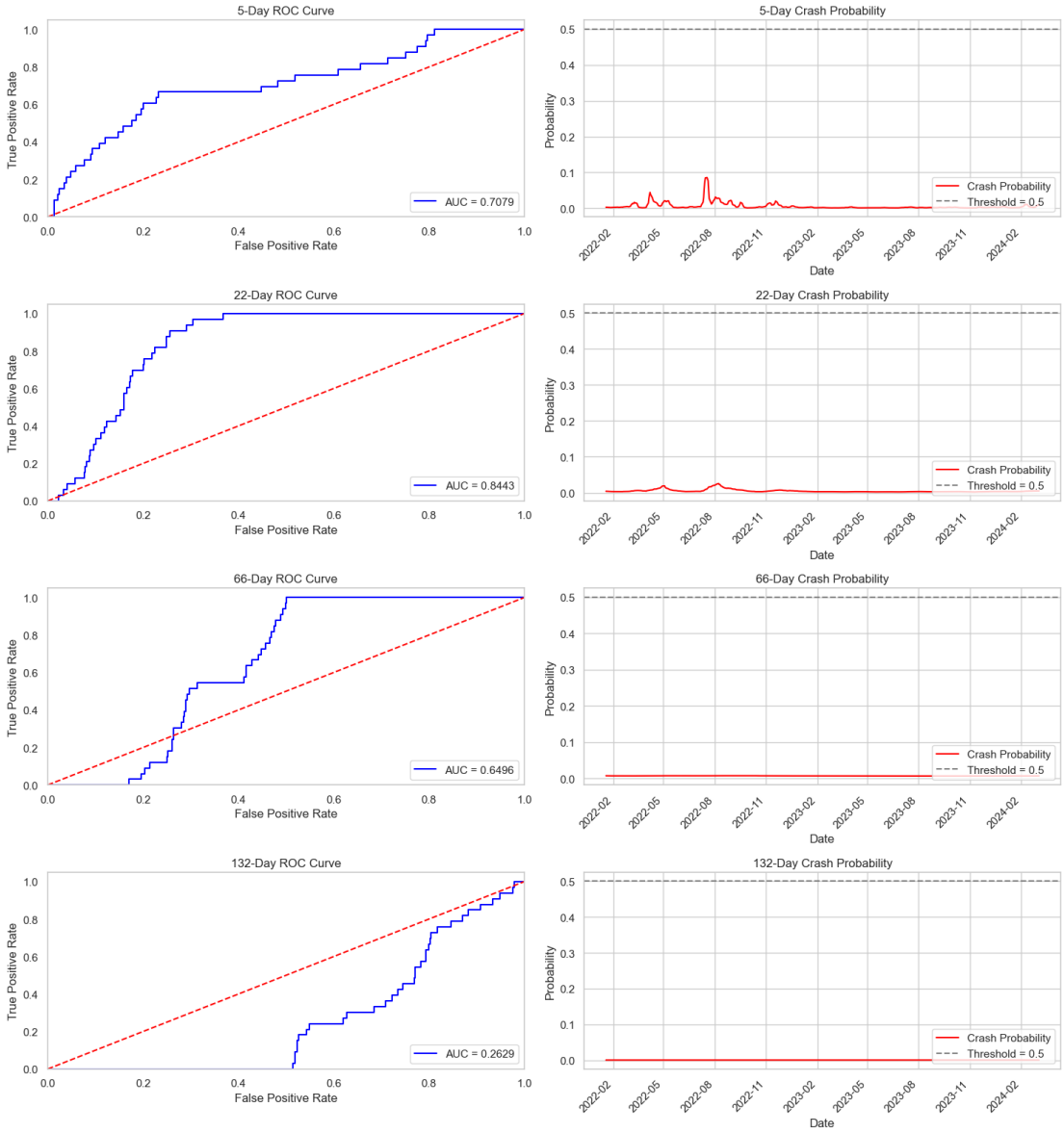


Figure A.11: ROC curves and crash probability plots for LSTM with Sentiment features across window sizes 5, 22, 66, and 132.

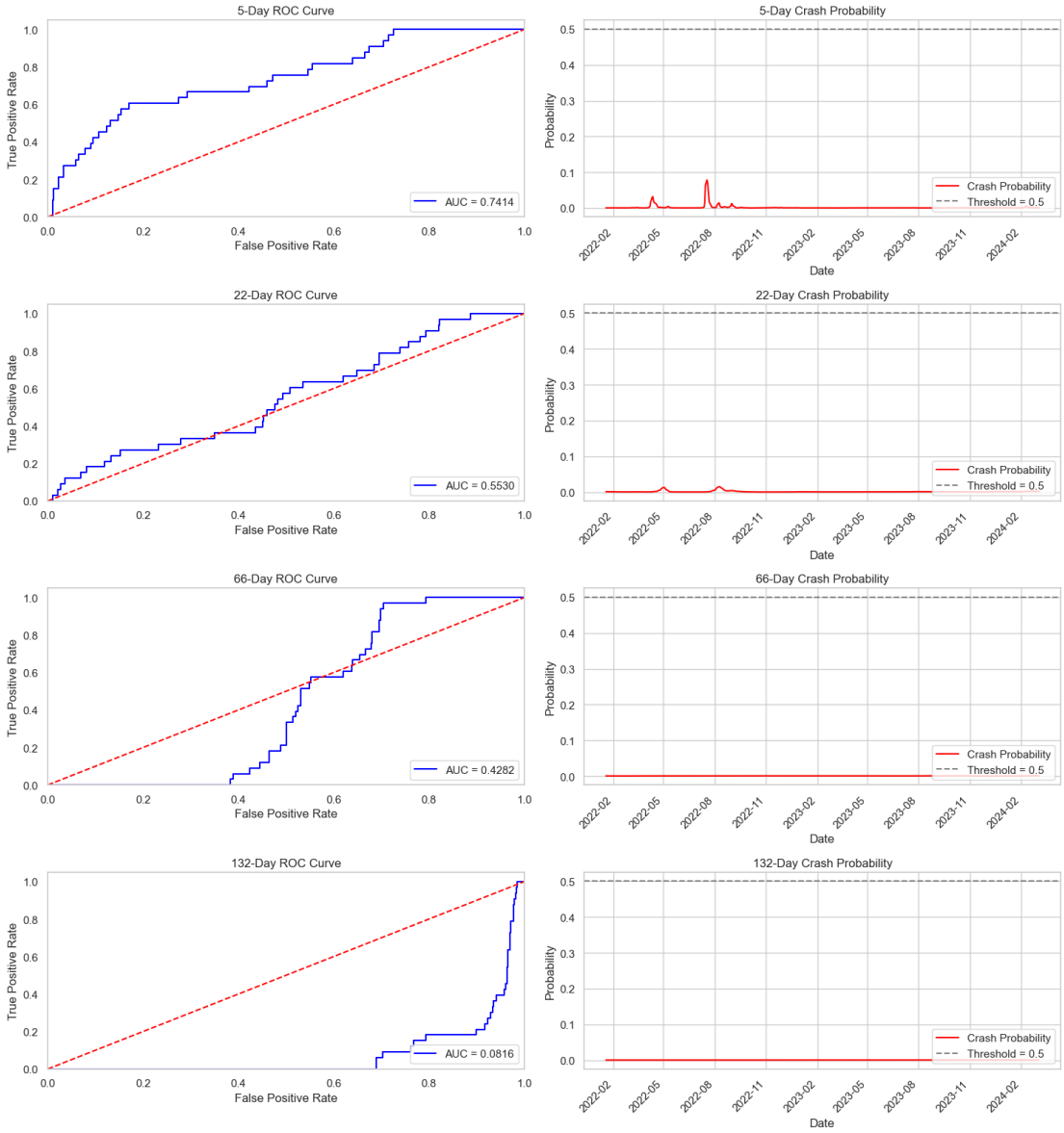


Figure A.12: ROC curves and crash probability plots for LSTM with Combined features across window sizes 5, 22, 66, and 132.

Appendix B Evaluation Metrics

Table B.1: The evaluation metrics along with their mathematical definitions.

Metric	Equation	Description
TP	–	True Positives: Correctly predicted crash instances.
TN	–	True Negatives: Correctly predicted non-crash instances.
FP	–	False Positives: Non-crash instances incorrectly predicted as crashes.
FN	–	False Negatives: Crash instances missed by the model.
TPR	$TP/(TP+FN)$	True Positive Rate / 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.
TNR	$TN/(TN+FP)$	True Negative Rate / Specificity: Indicates the proportion of non-crash events correctly predicted.
FPR	$FP/(FP+TN)$	False Positive Rate: Reflects the likelihood of the model incorrectly raising a crash alert when none occurred.
FNR	$FN/(TP+FN)$	False Negative Rate: Indicates how often the model fails to detect an actual crash, especially critical in risk-sensitive domains.
PPV	$TP/(TP+FP)$	Positive Predictive Value / Precisions: Shows the proportion of predicted crashes that were actual crashes.
FOR	$FN/(FN+TN)$	False Omission Rate: Measures the probability of missing a crash among the instances predicted as non-crash.
NSR	FPR/TPR	Noise-to-Signal Ratio: Quantifies the level of false alarms relative to correct crash predictions. A lower NSR indicates a more reliable EWS.
ACC	$(TP+TN)/(TP+TN+FP+FN)$	Accuracy: Represents the overall proportion of correct predictions.
$F1$	$2\cdot(PPV\cdot TPR)/(PPV+TPR)$	F1-Score: Balances measure of a model's ability to correctly detect crashes while minimizing false alarms.

Table B.2: Weights for weighted scoring equation.

Metric	Weight
w_{TPR}	+0.60
w_{TNR}	+0.30
w_{FPR}	-0.10
w_{FNR}	-0.30
w_{PPV}	+0.20
w_{FOR}	-0.10
w_{NSR}	-0.10
w_{ACC}	0.20

Table B.3: Performance of all models sorted by weighted score (Score).

Model	TPR	TNR	FPR	FNR	PPV	FOR	NSR	ACC	AUC	F1	Score
CNN_Sentiment_22	0.9394	0.7333	0.2667	0.0606	0.1902	0.0055	0.2839	0.7462	0.8342	0.3163	0.8971
Static_Logit_Combined_22	0.8605	0.7374	0.2626	0.1395	0.2216	0.0162	0.3052	0.7472	0.8283	0.3524	0.831
LSTM_Sentiment_66	0.9697	0.499	0.501	0.0303	0.1143	0.004	0.5167	0.5284	0.6496	0.2045	0.7488
CNN_Sentiment_66	0.9697	0.4929	0.5071	0.0303	0.1131	0.0041	0.5229	0.5227	0.6443	0.2025	0.7443
LSTM_Market_22	0.6061	0.9152	0.0848	0.3939	0.3226	0.0279	0.14	0.8958	0.8521	0.4211	0.7385
Dynamic_Logit_Combined_22	0.6279	0.8828	0.1172	0.3721	0.3176	0.0353	0.1866	0.8625	0.8339	0.4219	0.7321
Dynamic_Logit_Sentiment_132	0.4186	0.9879	0.0121	0.5814	0.75	0.0486	0.029	0.9424	0.7915	0.5373	0.7026
LSTM_Sentiment_22	0.6667	0.8222	0.1778	0.3333	0.2	0.0263	0.2667	0.8125	0.8443	0.3077	0.7021
Dynamic_Logit_Market_132	0.9535	0.4384	0.5616	0.0465	0.1285	0.0091	0.589	0.4796	0.592	0.2265	0.6953
Dynamic_Logit_Combined_132	0.4186	0.9838	0.0162	0.5814	0.6923	0.0488	0.0386	0.9387	0.8171	0.5217	0.6877
LSTM_Market_66	0.9697	0.3879	0.6121	0.0303	0.0955	0.0052	0.6312	0.4242	0.444	0.1739	0.6682
CNN_Combined_66	0.9697	0.3717	0.6283	0.0303	0.0933	0.0054	0.6479	0.4091	0.5826	0.1702	0.6566
CNN_Market_132	0.9394	0.4121	0.5879	0.0606	0.0963	0.0097	0.6258	0.4451	0.4662	0.1746	0.655
LSTM_Market_132	0.9091	0.4121	0.5879	0.0909	0.0935	0.0145	0.6467	0.4432	0.4332	0.1695	0.6242
CNN_Market_66	0.9394	0.3657	0.6343	0.0606	0.0899	0.0109	0.6753	0.4015	0.4096	0.164	0.6214
Dynamic_Logit_Sentiment_5	0.7209	0.6444	0.3556	0.2791	0.1498	0.0363	0.4932	0.6506	0.7186	0.248	0.6137
Static_Logit_Combined_132	0.2558	1	0	0.7442	1	0.0607	0	0.9405	0.7248	0.4074	0.6123
Static_Logit_Market_132	0.8372	0.4848	0.5152	0.1628	0.1237	0.0283	0.6153	0.513	0.5479	0.2156	0.6104
Static_Logit_Combined_66	1	0.2364	0.7636	0	0.1021	0	0.7636	0.2974	0.4053	0.1853	0.5981
Static_Logit_Combined_5	0.8372	0.4586	0.5414	0.1628	0.1184	0.0299	0.6467	0.4888	0.6233	0.2075	0.5981
Static_Logit_Market_66	0.9767	0.2586	0.7414	0.0233	0.1027	0.0078	0.7591	0.316	0.4159	0.1858	0.5895
CNN_Market_5	0.9394	0.3172	0.6828	0.0606	0.084	0.0126	0.7269	0.3561	0.5319	0.1542	0.5864
Dynamic_Logit_Market_22	1	0.2141	0.7859	0	0.0995	0	0.7859	0.277	0.5007	0.1811	0.5823
Static_Logit_Sentiment_132	0.2326	0.998	0.002	0.7674	0.9091	0.0626	0.0087	0.9368	0.7748	0.3704	0.5706
LSTM_Combined_66	0.9394	0.2949	0.7051	0.0606	0.0816	0.0135	0.7505	0.3352	0.4282	0.1501	0.5704
CNN_Market_22	0.5152	0.8404	0.1596	0.4848	0.1771	0.037	0.3098	0.8201	0.7582	0.2636	0.5646
Dynamic_Logit_Market_66	1	0.1859	0.8141	0	0.0964	0	0.8141	0.2509	0.4156	0.1759	0.5624
Dynamic_Logit_Sentiment_22	0.4186	0.9152	0.0848	0.5814	0.3	0.0523	0.2027	0.8755	0.8109	0.3495	0.5524
Static_Logit_Sentiment_22	0.4419	0.8949	0.1051	0.5581	0.2676	0.0514	0.2377	0.8587	0.814	0.3333	0.552
Static_Logit_Market_22	0.6279	0.6606	0.3394	0.3721	0.1385	0.0466	0.5405	0.658	0.6837	0.2269	0.5299
CNN_Combined_132	0.5758	0.7273	0.2727	0.4242	0.1234	0.0374	0.4737	0.7178	0.6866	0.2032	0.5263
Dynamic_Logit_Combined_5	0.5116	0.7818	0.2182	0.4884	0.1692	0.0515	0.4264	0.7602	0.7018	0.2543	0.5112
Static_Logit_Sentiment_66	1	0.1131	0.8869	0	0.0892	0	0.8869	0.184	0.293	0.1638	0.5112
LSTM_Market_5	0.6061	0.6687	0.3313	0.3939	0.1087	0.0378	0.5467	0.6648	0.664	0.1843	0.5092
CNN_Sentiment_5	0.4545	0.8444	0.1556	0.5455	0.163	0.0413	0.3422	0.8201	0.6654	0.24	0.5051
Static_Logit_Market_5	0.8605	0.2889	0.7111	0.1395	0.0951	0.0403	0.8264	0.3346	0.5355	0.1713	0.4893
Dynamic_Logit_Combined_66	1	0.0667	0.9333	0	0.0851	0	0.9333	0.1413	0.3654	0.1569	0.4786
LSTM_Sentiment_5	0.3333	0.9071	0.0929	0.6667	0.193	0.0467	0.2788	0.8712	0.7079	0.2444	0.4431
LSTM_Combined_5	0.2424	0.9677	0.0323	0.7576	0.3333	0.0496	0.1333	0.9223	0.7414	0.2807	0.4381
Dynamic_Logit_Sentiment_66	1	0	1	0	0.0799	0	1	0.0799	0.3344	0.148	0.432
CNN_Combined_5	0.2727	0.9434	0.0566	0.7273	0.2432	0.0489	0.2074	0.9015	0.6265	0.2571	0.4261
LSTM_Sentiment_132	0.9697	0.0202	0.9798	0.0303	0.0619	0.0909	1.0104	0.0795	0.2629	0.1164	0.399
LSTM_Combined_132	0.9697	0.0141	0.9859	0.0303	0.0615	0.125	1.0167	0.0739	0.0816	0.1157	0.3913
Static_Logit_Sentiment_5	0.2558	0.9152	0.0848	0.7442	0.2075	0.066	0.3317	0.8625	0.6022	0.2292	0.3705
CNN_Combined_22	0.2424	0.8929	0.1071	0.7576	0.1311	0.0535	0.4417	0.8523	0.6578	0.1702	0.3225
Dynamic_Logit_Market_5	0.2791	0.8343	0.1657	0.7209	0.1277	0.0698	0.5936	0.79	0.5383	0.1752	0.3021
CNN_Sentiment_132	0.0303	0.998	0.002	0.9697	0.5	0.0608	0.0667	0.9375	0.5142	0.0571	0.3012
LSTM_Combined_22	0.2424	0.8485	0.1515	0.7576	0.0964	0.0562	0.625	0.8106	0.553	0.1379	0.2708

Appendix C Code Repository

https://github.com/namolert/dynamic-ews

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.