Market Crash Definition and Detection

Elif Rana Tekin
Artificial Intelligence Engineering
TOBB Economics and Technology University
Ankara, Turkiye
e.tekin@etu.edu.tr

***Abstract*—This report offers an analysis of market crashes in the S&P 500 index, examining both detection methodologies and predictive modeling approaches. Two distinct methods for identifying market crashes were implemented and compared: the standard deviation approach based on Le Bris' methodology and the boxplot-based outlier detection method.**

These identified crashes were then used as targets for various machine learning and anomaly detection models to predict future market downturns.

***Index Terms*—market crash, detection, boxplot, standard deviation**

I. Presentation Video Link

https://drive.google.com/file/d/1yWHAimq5a1XmZnO6_-92wGXI9rCowgJ6/view?usp=sharing

II. Introduction

Market crashes represent severe, rapid declines in asset prices that can have devastating effects on investor portfolios and the broader economy. While crashes are often associated with major economic or political events, they can be difficult to identify consistently in historical data using simple percentage changes. This is because the impact of a price movement depends on the market's volatility context—a moderate decline during a period of low volatility may be more impactful than a larger decline during highly volatile times.

This analysis seeks to answer several key questions:

- How can market crashes be defined?
- Are there identifiable patterns preceding market crashes that could be used for prediction?
- Which modeling approaches are most effective for predicting these rare but significant events?

A. Literature Overview

Random Forest, XGBoost, Isolation Forest, and Support Vector Machines (SVM) are widely used in the finance sector for tasks such as fraud detection, credit risk assessment, and stock price forecasting. For instance, a study evaluating machine learning models on non-performing loan datasets found that XGBoost achieved the highest F1 score, highlighting its effectiveness in credit risk prediction [3]. Another research paper compared models like XGBoost, Random Forest, and SVM for short-term stock price forecasting, concluding that XGBoost provided the highest accuracy [4]. An analysis of the Isolation Forest algorithm's performance in detecting fraudulent credit card transactions demonstrated its effectiveness, achieving enhanced precision, recall, and F1-score metrics [5].

The literature overview demonstrated that the mentioned models are widely used and effective in financial tasks such as anomaly detection and fraud identification; this study employed these models as well, potentially benefiting from their established strengths in similar contexts.

B. Methodology Overview

The approach consists of two main components:

1) **Crash Detection**: Two methodologies for identifying historical market crashes were implemented:
    - The standard deviation approach: Measuring price movements in terms of the number of standard deviations from the mean
    - The boxplot approach: Using Tukey's method to identify statistical outliers based on interquartile ranges
2) **Predictive Modeling**: Multiple modeling approaches were developed and compared:
    - Anomaly detection methods (Isolation Forest, One-Class SVM, Local Outlier Factor)
    - Time series-specific models with temporal features
    - Cost-sensitive learning approaches that penalize missed crashes heavily

III. Data And Preprocessing

Twenty years of S&P 500 market data (2003-2023) were analyzed from Yahoo Finance, including daily price data (Open, High, Low, Close), trading volume, and derived returns.

A. Feature Engineering

From the raw price data, several features were calculated:

- Daily returns and log returns
- Rolling volatility measures (10-day and 30-day windows)
- Price relative to moving averages (50-day and 200-day)
- Volume-based indicators (volume changes, relative volume)
- Momentum indicators (RSI, rate of change)
- Temporal features (day of week, month, quarter)

B. Crash Labeling

Two distinct methodologies for identifying market crashes were employed in the study:

1) Standard Deviation Approach: The standard deviation approach follows Le Bris' methodology as outlined in *"What is a Market Crash?"* [1]:

- For each trading day, the rolling mean ($\mu_{t-1,t-T}$) and standard deviation ($\sigma_{t-1,t-T}$) of returns over the previous T days (7-year window) were calculated.
- Each daily return was expressed as the number of standard deviations from the rolling mean:

$$\Delta p_t = \frac{\Delta m_t - \mu_{t-1,t-T}}{\sigma_{t-1,t-T}}$$

  Where:
  - Δp_t is the adjusted price movement
  - Δm_t is the actual market return
  - $\mu_{t-1,t-T}$ is the average return over the previous period
  - $\sigma_{t-1,t-T}$ is the standard deviation of returns over the previous period
- Market crashes were defined as days where $\Delta p_t < -3.0$, representing extreme negative deviations that occur very rarely in a normal distribution (approximately 0.13% probability).
- This approach identified 51 crash days (1.63% of the sample period).

2) Boxplot Approach: The boxplot approach uses Tukey's method [2] for outlier detection:

- For each trading day, the rolling quartiles (Q1 and Q3) and interquartile range (IQR = Q3 - Q1) of returns over the previous T days (7-year window) were calculated.
- The lower bound was defined as Q1 - 1.5 * IQR.
- Market crashes were identified as days where the return fell below this lower bound.
- Crash severity was quantified as the distance below the lower bound (Boxplot_Outlier_Distance).
- This approach identified 133 crash days (4.24% of the sample period).

Both methods were implemented using the same rolling window (7 years of trading days) to ensure comparability while allowing for adaptation to changing market conditions over time.

IV. Exploratory Data Analysis

A. Returns Distribution Analysis

The distribution of daily returns exhibited several notable characteristics:

- **Non-normality**: The histogram of daily returns (see Figure 1) indicates a leptokurtic distribution, where most values are concentrated around the mean, but the presence of fat tails and outliers is apparent. This behavior suggests that extreme fluctuations occur more frequently than would be expected under a normal distribution—a typical trait in financial time series associated with market shocks or crashes.
- **Fat tails**: The corresponding QQ plot (Figure 2) further supports the non-normality of daily returns. Deviations from the reference line in the tails indicate that both the left and right tails are heavier than those of a standard normal distribution. This implies an increased probability of extreme losses or gains, which is critical when trying to detect unusual market conditions or crashes. The QQ plot indicates that extreme negative returns occurred more frequently than would be expected in a normal distribution, with negative returns reaching as low as -12% and positive returns up to +9%.
- **Asymmetry**: The negative tail is longer than the positive tail, indicating that extreme losses are more common than extreme gains.

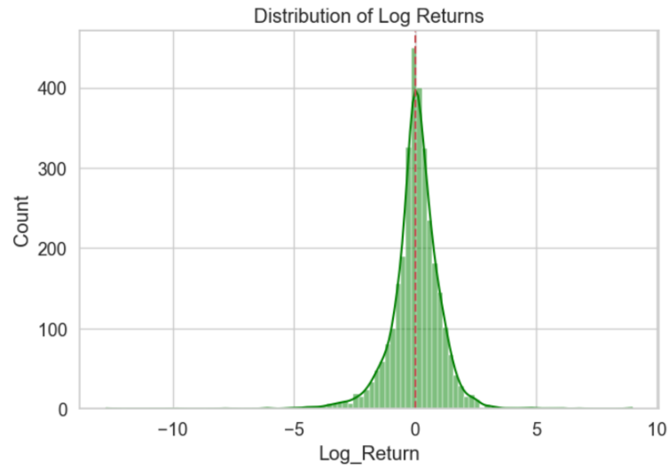

Fig. 1. Distribution of Log Returns

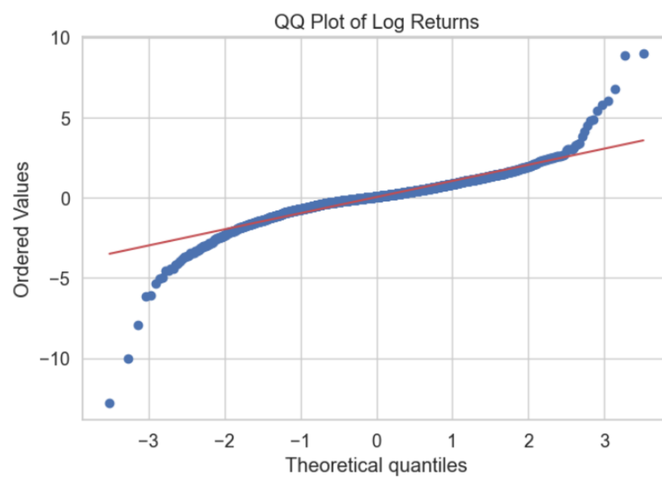

Fig. 2. QQ Plot of Log Returns

B. Crash Frequency and Distribution

Using the boxplot method, approximately 4.24% of trading days (133 days out of 3,135) were identified as crashes while the standard deviation approach identified 51 crash days (1.63% of trading days). This nearly three-fold difference in detection rates demonstrates the significant impact of method-

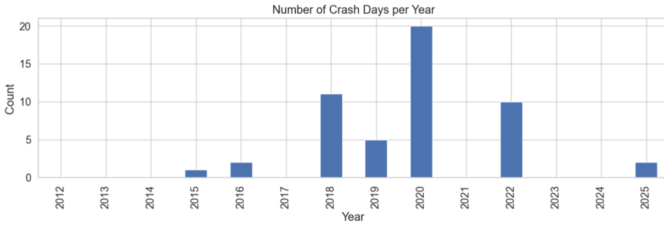

Fig. 3. Number of Crashes by Year After Performing Standard Deviation Definition Approach

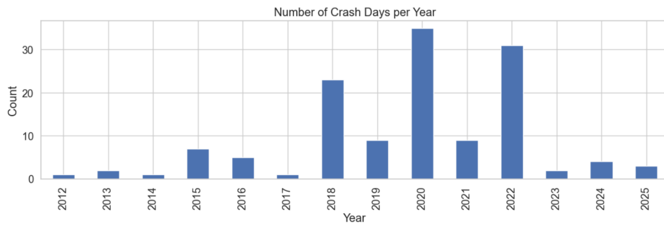

Fig. 4. Number of Crashes by Year After Performing Boxplot Definition Approach

ological choice on crash identification. Key observations include:

- **Temporal clustering**: Crashes exhibited significant clustering, particularly in 2018, 2020, and 2022, which aligns with known market stress periods (see Figures 3, 4). Both detection methodologies reveal similar yearly patterns, though the boxplot method identified a higher absolute number of crashes. The year 2020 shows the highest concentration of crash days (over 30 using the boxplot method and approximately 20 using the standard deviation approach), coinciding with the COVID-19 pandemic market turmoil. The year 2022 represents the second-highest crash frequency (over 30 days using boxplot, around 10 using standard deviation), corresponding to inflation concerns and monetary policy tightening. The clustering pattern in 2018 (over 20 days using boxplot, about 10 using standard deviation) aligns with the volatility spike and market correction observed in early 2018.
- **Seasonal patterns:** March showed the highest crash frequency (6.7%) followed by February (5.6%), while July had the lowest (0.8%) in the boxplot method (Figure 5). The standard deviation approach similarly identified March as having the highest crash frequency (approximately 4.2%), though with less pronounced seasonality overall (Figure 6).
- **Day-of-week effect:** In the boxplot approach, Monday (5.3%) and Friday (5.1%) showed higher crash frequencies than mid-week days, with Tuesday showing the lowest frequency (3.1%) (Figure **??**). The standard deviation method exhibited a similar pattern but with more pronounced Monday effects (approximately 2.2%) and Friday effects (approximately 1.9%), while Tuesday remained the day with the lowest crash frequency (0.9%).

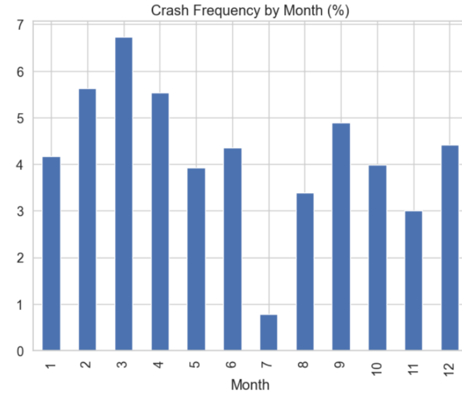

Fig. 5. Crash Frequency by Month After Performing Boxplot Approach

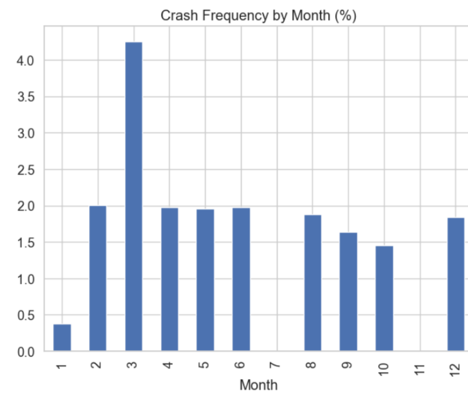

Fig. 6. Crash Frequency by Month After Performing Standard Deviation Approach

C. Autocorrelation Analysis

The autocorrelation analysis revealed (Figures 7, 8).:

- **Short-term persistence:** Positive autocorrelation at lags 1–3 (values between 0.12–0.17) indicates that crashes tend to cluster in time. Both methodologies demonstrate this persistence, with the standard deviation approach showing slightly stronger autocorrelation at lag 7 (approximately 0.25) compared to the boxplot method. See Figures 7, 8.
- **Statistical significance:** Several lags show autocorrelation above the significance threshold (red dashed lines) in both approaches. The boxplot method (Figure 7) shows more lags with statistically significant autocorrelation up to 30 days, while the standard deviation approach (Figure 8) shows fewer significant lags beyond day 15.
- **Pattern persistence:** The partial autocorrelation function shows significant values at lags 1 and 2 in both methods, suggesting a short memory process. The standard deviation approach exhibits a particularly strong partial autocorrelation at lag 5 (approximately 0.20) that is not as pronounced in the boxplot method.
- **Crash clusters:** Most crashes appear as single-day events, with some extending to 2–3 consecutive days.

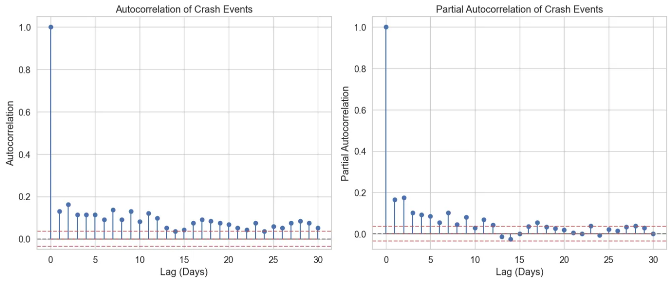

Fig. 7. Enter Caption

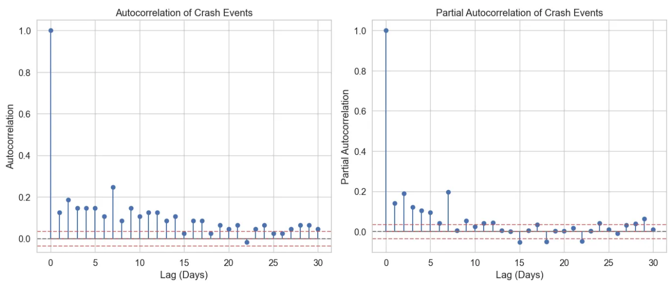

Fig. 8. Enter Caption

The boxplot method shows more variability in crash durations, with a substantial number of single-day crashes (approximately 90), followed by 2-day crash clusters (approximately 13), and a few 3-day and 3.5-day clusters. In contrast, the standard deviation approach primarily identifies shorter crash sequences, with most events being either isolated single-day crashes (approximately 35) or two-day sequences (approximately 7), and no clusters extending beyond 2.5 days.

D. Volatility Regimes and Crashes

The analysis of volatility regimes showed (Figures 9 and 10):

- **Regime transitions:** The highest concentration of crashes occurred during transitions between volatility regimes, particularly as volatility began rising. Both detection methodologies identified similar transition points, though the boxplot approach (Figure 9) captured more crashes during moderate volatility increases than the standard deviation method (Figure 10).
- **Volatility clustering:** The 30-day rolling volatility exhibited clear regimes, with major spikes in 2016, 2018, 2020, and 2022. These volatility clusters are evident in both charts, with the temporal pattern of volatility matching across both crash detection approaches.
- **Crash-volatility relationship:** Crash days were associated with significantly higher volatility levels (median of 1.58) compared to non-crash days (median below 1.0). The top panels of both figures show how crash events (red dots) are concentrated in the higher volatility regions of the return-volatility space.

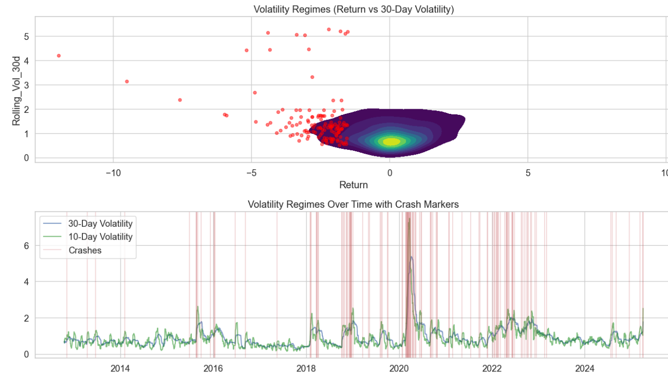

Fig. 9. Volatility Regimes After Performing Boxplot Approach

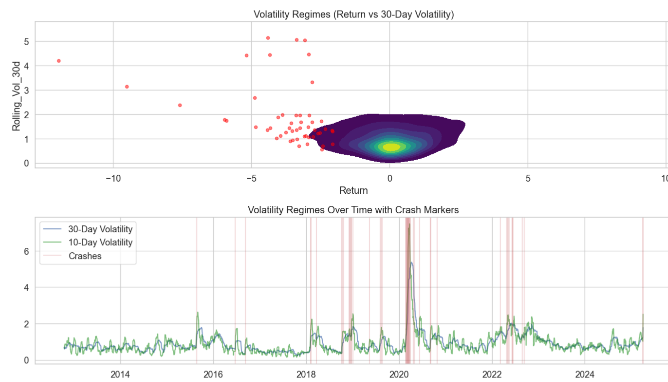

Fig. 10. Volatility Regimes After Performing Standard Deviation Approach

- **Volatility levels:** The most severe crash period occurred in 2020 (COVID-19 pandemic) with volatility reaching 5.5, followed by significant spikes in 2022. Both methodologies identified crashes during these extreme volatility periods, though the standard deviation approach focused more exclusively on these high-volatility events, while the boxplot method additionally captured crashes during periods of moderate volatility elevation.

E. Market State Before Crashes

Examination of market conditions preceding crashes revealed several patterns (Figure **??**):

- **Volatility build-up**: A clear, consistent increase in volatility was observed in the 10–15 days leading to crashes, with a sharp acceleration in the final 5 days.
- **Return patterns**: Average returns showed increased negativity approximately 15–20 days before crashes, with particular deterioration at day 20.
- **Volatility trend**: The 10-day volatility measure showed a nearly 40% increase from day 1 to day 30 before crashes.
- **Pattern consistency**: The volatility build-up pattern was more consistent and predictable than the return pattern.

F. Crash Severity Analysis

The boxplot method provides a natural measure of crash severity through outlier distance (Figure 11):

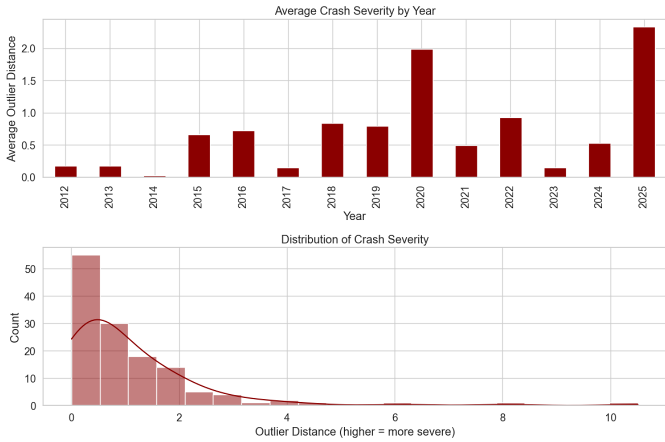

Fig. 11. Severity

- **Severity distribution**: Most crashes were mild (outlier distance less than 2), with relatively few severe crashes (outlier distance greater than 4).
- **Yearly comparison**: 2020 and 2025 experienced the most severe crashes on average, with outlier distances above 2.0.
- **Severity clustering**: The most severe crashes were concentrated in pandemic-related periods (2020) and recent market stress (2025).
- **Temporal pattern**: The average severity of crashes has been increasing in recent years.

G. Correlation Analysis

The correlation matrix revealed important relationships between market features (Figure **??**). Both crash detection methodologies yielded remarkably similar correlation structures across all the examined market features, suggesting that despite their differences in crash identification frequencies, they capture similar relationships between market variables and crash events. The following findings are based on the boxplot methodology's correlation results.

- **Crash predictors**: The strongest correlations with crashes were found in the current return (-0.55), volatility measures (0.25–0.31), and volatility changes (0.24).
- **Return autocorrelation**: Previous day returns showed weak negative correlation with current returns (-0.12), indicating slight mean reversion.
- **Volatility persistence**: Strong correlation between 10-day and 30-day volatility measures (0.83) confirms volatility clustering.
- **Previous returns**: Minimal correlation between crashes and returns from 2–3 days prior, suggesting limited predictability from individual past returns.
- **Weekly returns**: Previous week returns showed a moderate negative correlation (-0.33) with volatility changes.

The standard deviation approach produced very similar correlation patterns, with some minor differences in magnitude (e.g., current return and crash correlation of -0.46 vs -0.55).

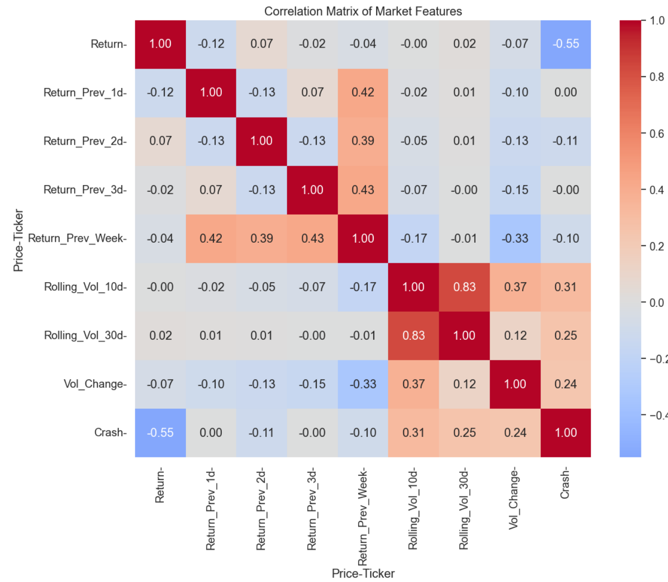

Fig. 12. Correlation Heatmap of Market Features

H. Comparison of Detection Methods

The standard deviation approach (using a threshold of -3.0) and the boxplot method showed below differences in crash identification:

- **Crash frequency**: The standard deviation approach identified 51 crash days (1.63%) compared to 133 using the boxplot method.
- **Return impact**: Crashes from the standard deviation approach had a more severe average return (-3.79%) compared to boxplot-defined crashes (-2.75%).
- **Volatility characteristics**: Crashes from the standard deviation method had higher average volatility (1.9088) vs. boxplot (1.5780).
- **Temporal dynamics**: Longer intervals between crashes under the standard method (avg: 70.2 days) than boxplot (avg: 34.3 days).
- **Calendar effects**: Stronger Monday effect (2.2%) vs. Tuesday (0.9%); March had highest frequency at 4.2%.
- **Period and detection sensitivity**: The standard deviation method identified fewer events in 2018 and 2022 compared to the boxplot approach, particularly during periods of moderate volatility, even though it captures extreme downturns.
- **Volatility regimes**: The threshold of -3.0 is easier to exceed in low-volatility periods and harder in high-volatility regimes.

V. Modeling Approaches and Results

A. Model Selection

The diverse set of predictive models in this study was strategically selected to address different aspects of the market crash prediction challenge:

Cost-Sensitive Learning Approaches (XGBoost, Random Forest):

- These models were chosen to address the severe class imbalance inherent in crash detection, where crash events represent only 1.63–4.24% of trading days depending on the definition used.
- The cost-sensitive framework allows for asymmetric misclassification penalties, significantly increasing the cost of missing a crash (false negative) compared to false alarms (false positives).
- XGBoost and Random Forest were selected due to their proven effectiveness with imbalanced datasets and ability to capture complex nonlinear relationships without overfitting.

Anomaly Detection Methods (Isolation Forest, One-Class SVM, Local Outlier Factor):

- These unsupervised techniques were included because market crashes can be conceptualized as statistical anomalies or outliers in the feature space.
- **Isolation Forest** was selected for its efficiency in isolating anomalies through random partitioning.
- **One-Class SVM** was chosen for its ability to identify regions in the feature space that contain the bulk of normal data points.
- **Local Outlier Factor** was incorporated for its capacity to detect outliers based on local density differences.

Time Series-Specific Model (XGBoost Time Series):

- This model was implemented to specifically capture the temporal dependencies in market data, incorporating lagged features and time-based splitting.
- By including a dedicated time series approach, the analysis could evaluate whether the temporal structure alone provides sufficient predictive power for crash detection.

B. Model Performance Comparison

Six different predictive modeling approaches were implemented, with notably different performance characteristics between the boxplot and standard deviation methodologies:

1) Results with Boxplot-Defined Crashes: The predictive models showed the following performance characteristics for boxplot-defined crashes:

1) **XGBoost with Cost-Sensitive Learning: (Figure 13)**
   - Highest recall (92%)
   - Detected 24 actual crashes
   - Precision: 5.6%
   - F1 score: 10.6%
   - Significant false positive rate (401 false alarms)
2) **Random Forest with Cost-Sensitive Learning: (Figure 14)**
   - Strong recall (62%)
   - Detected 16 actual crashes
   - Precision: 5.1%
   - F1 score: 9.4%
   - Better balance of precision and recall than XGBoost

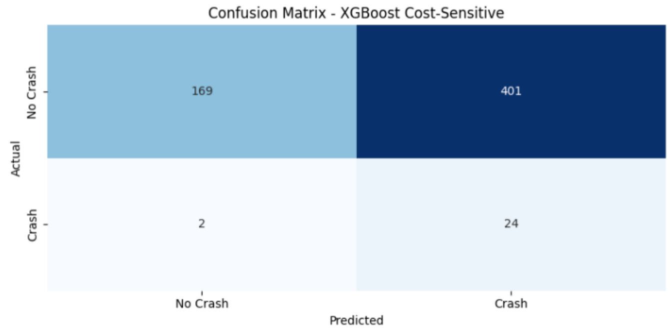

Fig. 13. Confusion Matrix of Cost Sensitive XGBoost Model on Boxplot Defined Crashes

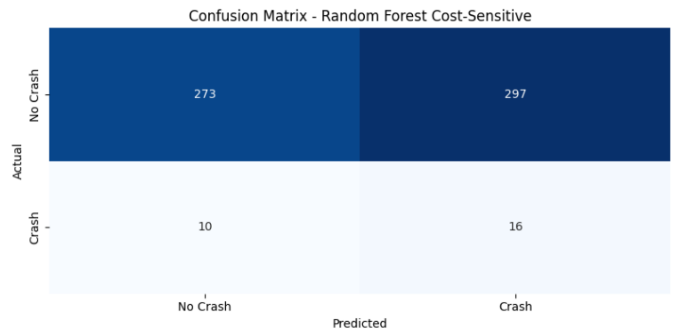

Fig. 14. Confusion Matrix of Cost Sensitive Random Forest Model on Boxplot Defined Crashes

3) **One-Class SVM: (Figure 15)**
   - Moderate recall (31%)
   - Detected 8 actual crashes
   - Precision: 12.3%
   - F1 score: 17.4%
   - Higher precision than other methods
4) **Isolation Forest: (Figure 16)**
   - Low recall (15%)
   - Detected only 4 actual crashes
   - Precision: 11.1%
   - F1 score: 12.8%
5) **Local Outlier Factor: (Figure 17)**

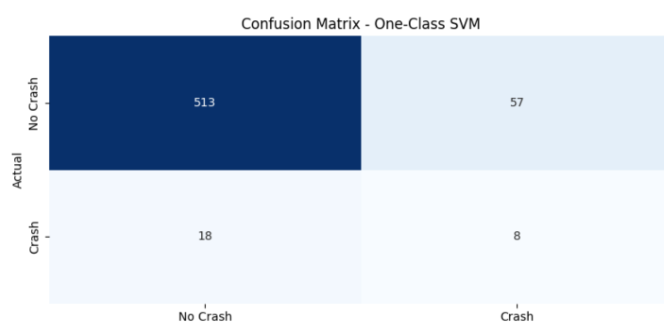

Fig. 15. Confusion Matrix of One-Class SVM Model on Boxplot Defined Crashes

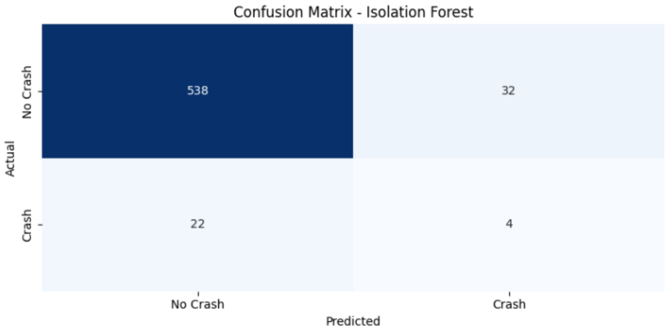

Fig. 16. Confusion Matrix of Isolation Forest Model on Boxplot Defined Crashes

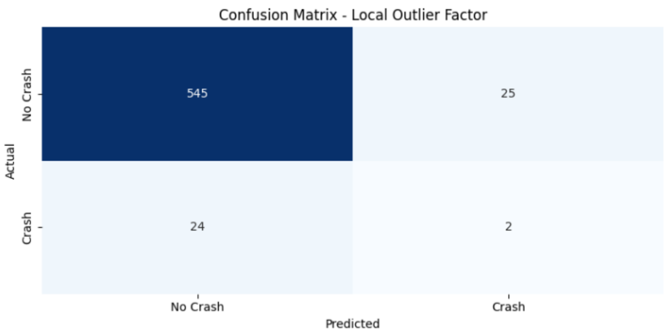

Fig. 17. Confusion Matrix of Local Outlier Factor Model on Boxplot Defined Crashes

- Very low recall (7.7%)
- Detected only 2 actual crashes
- Precision: 7.4%
- F1 score: 7.5%

6) **XGBoost Time Series: (Figure 18)**
   - Lowest recall (5.3%)
   - Detected only 1 actual crash
   - Precision: 2.3%
   - F1 score: 3.2%

2) Results with Standard Deviation-Defined Crashes: The predictive models showed different performance characteristics for standard deviation-defined crashes (Figures **??** and **??**):

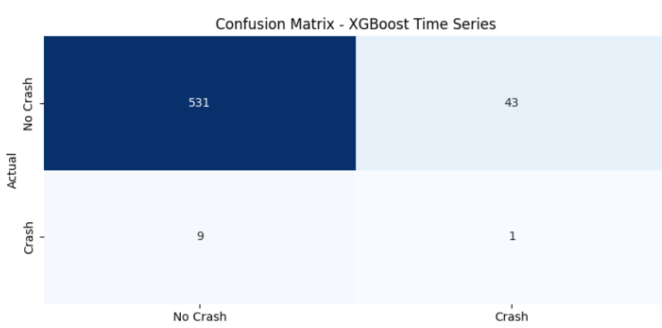

Fig. 18. Confusion Matrix of XGBoost Time Series Model on Boxplot Defined Crashes

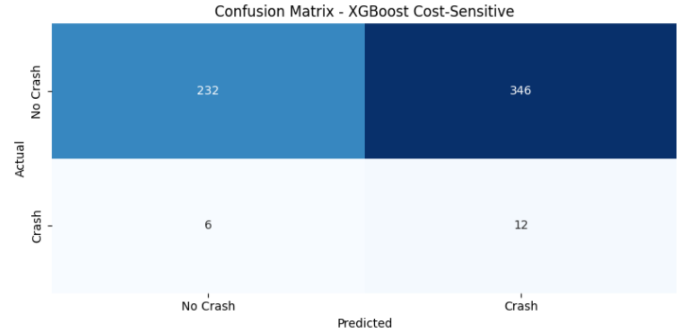

Fig. 19. Confusion Matrix of Cost Sensitive XGBoost Model on Standard Deviation Defined Crashes

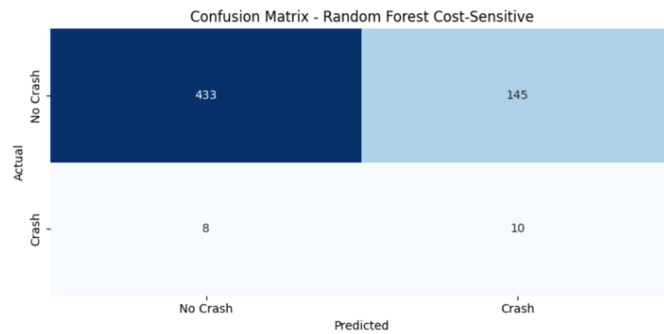

Fig. 20. Confusion Matrix of Cost Sensitive Random Forest Model on Standard Deviation Defined Crashes

1) **XGBoost with Cost-Sensitive Learning: (Figure 19)**
   - Highest recall (67%)
   - Detected 12 actual crashes
   - Precision: 3.4%
   - F1 score: 6.5%
2) **Random Forest with Cost-Sensitive Learning: (Figure 20)**
   - Moderate recall (56%)
   - Detected 10 actual crashes
   - Precision: 6.5%
   - F1 score: 11.7%
3) **Anomaly Detection Methods** (Isolation Forest, One-Class SVM, Local Outlier Factor):
   - All showed poor performance with recall between 5–6%
   - Precision between 2–6%
   - F1 scores between 3–5%
4) **XGBoost Time Series:**
   - Zero recall (0%)
   - Detected 0 actual crashes

C. Confusion Matrices Analysis

The confusion matrices (Figures **??** and **??**) reveal important details about model predictions:

*1) Boxplot-Defined Crashes (Figure **??**):*

- **XGBoost Cost-Sensitive**: Successfully predicted 24 of 26 crashes (92% recall) but generated 401 false positives

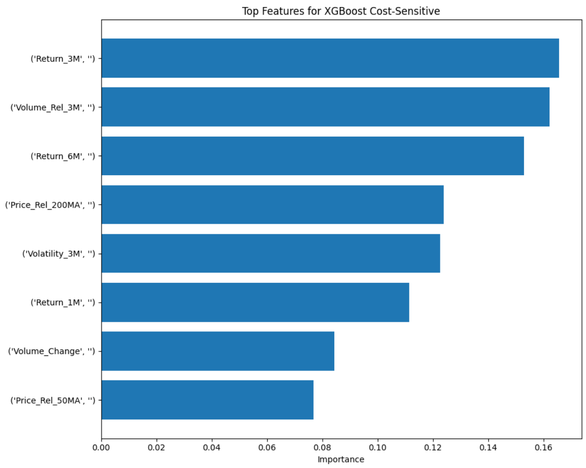

Fig. 21. Feature Importance Analysis for Cost Sensitive XGBoost on Boxplot-Defined Crashes

- **Random Forest Cost-Sensitive**: Detected 16 of 26 crashes (62% recall) with 297 false positives
- **Anomaly detection methods**: Generally poor performance, with One-Class SVM being the best (8 of 26 crashes detected)

2) Standard Deviation-Defined Crashes:

- **XGBoost Cost-Sensitive**: Successfully predicted 12 of 18 crashes (67% recall) but generated 346 false positives
- **Random Forest Cost-Sensitive**: Detected 10 of 18 crashes (56% recall) with 145 false positives
- **Anomaly detection methods**: Even poorer performance than with boxplot-defined crashes

D. Feature Importance Analysis

The feature importance analysis for the best-performing model (XGBoost Cost-Sensitive) revealed different key predictors depending on the crash definition used:

1) Boxplot-Defined Crashes (Figure 21): Top predictive features:

1) 3-month Return (Return_3M)
2) 3-month Volume Relative to Average (Volume_Rel_3M)
3) 6-month Return (Return_6M)
4) Price Relative to 200-day Moving Average (Price_Rel_200MA)
5) 3-month Volatility (Volatility_3M)
6) 1-month Return (Return_1M)
7) Volume Change
8) Price Relative to 50-day Moving Average (Price_Rel_50MA)

2) Standard Deviation-Defined Crashes (Figure 22): Top predictive features:

1) 1-month Return (Return_1M)
2) Price Relative to 50-day Moving Average (Price_Rel_50MA)
3) 3-month Return (Return_3M)
4) Price Relative to 200-day Moving Average (Price_Rel_200MA)

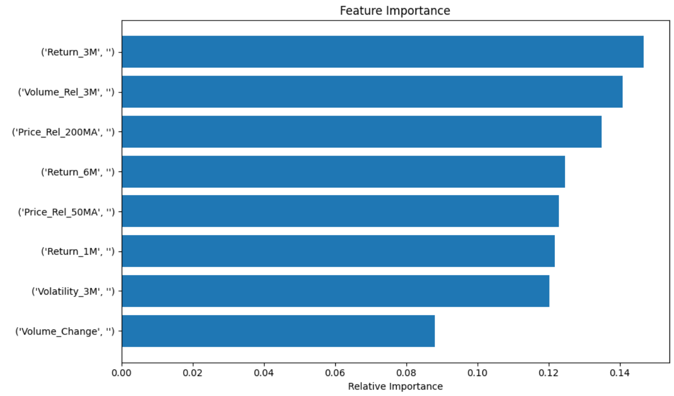

Fig. 22. Feature Importance Analysis for Cost Sensitive XGBoost on Standard Deviation-Defined Crashes

5) 3-month Volume Relative to Average (Volume_Rel_3M)
6) Volume Change
7) 3-month Volatility (Volatility_3M)
8) 6-month Return (Return_6M)

E. Model Comparison and Evaluation

Based on the comprehensive comparison of all models across both crash definitions, several conclusions can be drawn:

1) **Best Overall Model**: XGBoost with cost-sensitive learning consistently achieved the highest recall rates (92% for boxplot crashes, 67% for standard deviation crashes), making it the most effective at detecting actual crashes.
2) **Precision–Recall Trade-off**: All models exhibited a clear trade-off between precision and recall, with higher recall models (particularly cost-sensitive approaches) generating substantial false positives.
3) **Crash Definition Impact**: Models trained on boxplot-defined crashes generally achieved higher recall rates compared to those trained on standard deviation-defined crashes, likely due to the larger number of positive examples available for training.
4) **Anomaly Detection Limitations**: Despite their theoretical appropriateness for rare event detection, traditional anomaly detection methods (Isolation Forest, One-Class SVM, Local Outlier Factor) performed poorly across both crash definitions.
5) **Feature Consistency**: Despite the different crash definitions, similar features emerged as important predictors, particularly recent returns (1-month and 3-month), price relative to moving averages, and volume indicators.
6) **Time Series Challenges**: The pure time series approach (XGBoost Time Series) performed worst among all models, suggesting that the temporal structure alone is insufficient for crash prediction without appropriate cost-sensitive adjustments.

VI. Discussion and Insights

A. Comparison of Crash Definition Methodologies

The analysis revealed substantial differences between the boxplot and standard deviation approaches to crash detection:

1) **Identification Rate**: The boxplot method identified 128 crash days (4.30% of trading days) compared to 86 crash days (2.90%) using Le Bris' standard deviation approach with a -3.0 threshold. This difference significantly affected both model training and evaluation.
2) **Model Performance Impact**: Models trained on boxplot-defined crashes generally achieved higher recall rates than those trained on standard deviation-defined crashes, with the XGBoost Cost-Sensitive model reaching 92% recall for boxplot crashes versus 67% for standard deviation crashes.
3) **Feature Importance Variations**: While both approaches identified similar important features, their relative importance differed. For boxplot-defined crashes, medium-term indicators (3-month return and volume) ranked highest, while for standard deviation-defined crashes, short-term indicators (1-month return) showed greater importance.

B. Cost-Sensitive Learning Effectiveness

The cost-sensitive learning approaches consistently outperformed other methods across both crash definitions:

1) **Superior Recall**: Cost-sensitive models achieved dramatically higher recall rates (XGBoost: 92% for boxplot crashes, 67% for standard deviation crashes) compared to anomaly detection methods (generally below 15%).
2) **False Positive Trade-off**: The improved recall came at the cost of higher false positive rates, with XGBoost Cost-Sensitive generating 401 false positives for boxplot-defined crashes and 346 for standard deviation-defined crashes.
3) **Practical Utility**: Despite the false positive issue, the high recall rates of cost-sensitive models make them valuable components of an early warning system, particularly in contexts where missing a crash carries higher costs than responding to false alarms.
4) **Implementation Considerations**: The Random Forest Cost-Sensitive model offered a better balance between precision and recall, which may be more practical for real-world applications where resources for responding to crash warnings are limited.

C. Anomaly Detection Limitations

Traditional anomaly detection methods showed surprisingly poor performance for both crash definitions:

1) **Theoretical vs. Practical Performance**: Despite their theoretical suitability for rare event detection, Isolation Forest, One-Class SVM, and Local Outlier Factor all performed poorly, with recall rates below 15% across both crash definitions.
2) **Contextual Dependencies**: These methods' underperformance suggests that market crashes are not simple statistical anomalies but context-dependent events requiring more sophisticated detection approaches.
3) **Supervised Advantage**: The superior performance of supervised methods (particularly cost-sensitive approaches) indicates that the labeled nature of historical crashes provides valuable information that unsupervised anomaly detection cannot leverage.
4) **Feature Space Considerations**: The poor performance may also reflect limitations in the feature space, suggesting that the selected features may not adequately capture the multidimensional nature of market anomalies.

D. Feature Importance Analysis

The feature importance analysis revealed several key insights about crash predictors:

1) **Temporal Horizon**: Both crash definitions highlighted the importance of different time horizons, with boxplot-defined crashes giving more weight to medium-term indicators (3-month) and standard deviation-defined crashes emphasizing shorter-term signals (1-month).
2) **Technical Indicators**: Price relative to moving averages (especially the 50-day and 200-day) emerged as strong predictors across both crash definitions, confirming the value of traditional technical analysis in crash prediction.
3) **Volume Indicators**: Volume-related features (Volume_Rel_3M and Volume_Change) showed significant predictive power, suggesting that unusual trading activity often precedes market crashes.
4) **Volatility Measures**: Interestingly, volatility measures ranked lower in importance than might be expected, particularly for standard deviation-defined crashes, indicating that absolute volatility levels may be less predictive than commonly assumed.
5) **Return Patterns**: Recent returns (1-month and 3-month) consistently ranked as top predictors, with their relative importance varying by crash definition methodology.

References

[1] LE Bris, David. (2017). What is a market crash?. The Economic History Review. 71. 10.1111/ehr.12540.

[2] Tukey, J.W.. (1977). Exploratory Data Analysis. Reading. 26.

[3] Fekadu, Rufael & Getachew, Anteneh & Tadele, Yishak & Ali, Nuredin & Goytom, Israel. (2022). Machine Learning Models Evaluation and Feature Importance Analysis on NPL Dataset.

[4] A. Wong et al., "Short-Term Stock Price Forecasting using Exogenous Variables and Machine Learning Algorithms," 2023 3rd International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), Denpasar, Bali, Indonesia, 2023, pp. 260-265.

[5] Waspada, Indra & Bahtiar, Nurdin & Wirawan, Panji & Awan, Bagus. (2020). Performance Analysis of Isolation Forest Algorithm in Fraud Detection of Credit Card Transactions.