Title: Anomaly Detection and Time Series Forecasting in the Irish Finance Market: Evaluating the Performance of the Prophet Model

**JOAO RIBEIRO**

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

Diagram

Description automatically generated with medium confidence

September 2023

**Supervisor: Vikas Tomer**

**TABLE OF CONTENTS**

Abstract .................................................................................................................................... pag.

1. Introduction……………………………………………………………………………………………………….. pag.

1.1 Background and Motivation ............................................................................................. pag.

1.2 Research Objectives ....................................................................................................... pag.

1.3 Scope and Limitations ................................................................................................ pag.

2. Literature Review

2.1 For Time Series Forecasting Techniques………………………………………..……………………….. pag.

2.1.1 Prophet Model ............................................................................................................ pag.

2.1.2 Statistical Time Series Model (ARIMA, SARIMA, GARCH) ……………………………………….. pag.

2.1.3 Deep learning model (NeuralProphet, LSTM) ............................................................. pag.

2.2 For Anomaly Detection Approaches

2.2.1 Prophet Model ............................................................................................................ pag.

2.2.2 Ensemble Anomaly Detection (Isolation Forest) ………………………………………………….. pag.

2.2.3 Support Vector Machine Anomaly Detection (One-Class SVM) ……………………………. pag.

2.3.4 Deep learning model (NeuralProphet, LSTM) .............................................................. pag.

3. Data Collection and Preprocessing

3.1 Data Sources…………………………………………………………………………………………………………….. pag.

3.1.1 Indexes ....................................................................................................................... pag.

3.1.2 Sources ........................................................................................................................ pag.

3.2 Exploratory Data Analysis

3.2.1 Data Preprocessing Steps............................................................................................ pag.

3.2.1.1 Missing Values ....................................................................................................... pag.

3.2.1.2 Outliers ................................................................................................................. pag.

3.2.2 Basic statistics ............................................................................................................. pag.

3.2.3 Data distribution .......................................................................................................... pag.

3.2.3 PLOTS…………………………………………………………………………….………………………………….. pag.

4. Time Series Forecasting………………………………………………………..……………………………………….. pag.

4.1 Introduction to Time Series Forecasting Models…………………………………………………….. pag.

4.2.1 Model Architecture and Configuration ......................................................................... pag.

4.2.2 Training Parameter ........................................................................................................ pag.

4.2.3 Modeling and Optimization (Hyperparameter Tuning) ………….…………………………….. pag.

4.2.3.1 Bayesian optimization (Hyperparameter tuning) ………………………………………….. pag.

4.2.3.1 Prophet Model Forecast with Confidence Intervals ………....................................... pag.

4.3 Decomposition and Smoothing………………………………………………………………….………….. pag.

4.3.1 Trend analysis ............................................................................................................ pag.

4.3.2 Seasonal decomposition ............................................................................................ pag.

4.3.3 Residual component .................................................................................................. pag.

4.4 Performance Evaluation………………………………………………………….…………………………….. pag.

4.4.1 Performance Evaluation Metrics……………………………………………….…………………….. pag.

4.4.1.1 Mean Absolute Error (MAE) .................................................................................. pag.

4.4.1.2 Mean Squared Error (MSE) .................................................................................... pag.

4.4.1.3 Root Mean Squared Error (RMSE) ......................................................................... pag.

4.4.1.4 Mean Absolute Percentage Error (MAPE) ............................................................. pag.

4.4.1.5 R-squared (R²) ....................................................................................................... pag.

4.4.2 Performance Evaluation Metrics Comparison (ISEQ20) ………………………………………... pag.

4.5 Model Comparison…………………………………………………………………………………………………….. pag.

4.5.1 Models compared to Prophet Model………………………………………..……………………….. pag.

4.5.1.1 ARIMA (AutoRegressive Integrated Moving Average) …………………………………. pag.

4.5.1.2 SARIMA (Seasonal ARIMA) ................................................................................. pag.

4.5.1.3 LSTM (Long Short-Term Memory) ...................................................................... pag.

4.5.1.4 Neuralprophet ..................................................................................................... pag.

4.5.1.5 GARCH (Generalized Autoregressive Conditional Heteroskedasticity) ………......... pag.

4.5.2 Prophet Model vs. Other Algorithms ......................................................................... pag.

4.5.3 Scatter plot comparing Prophet with other algorithms. ............................................. pag.

4.6 Index Comparison ............................................................................................................. pag.

4.7.1 Basic Statistics European indexes .................................................................................. pag.

4.7.2 Missing values ............................................................................................................... pag.

4.7.3 Outliers .......................................................................................................................... pag.

4.7.4 Heatmap for performance comparison ......................................................................... pag.

4.7.5 Irish Vs German Index

4.7.5.1 Scatter Plot .............................................................................................................. pag.

4.7.5.2 Stacked Area Chart ................................................................................................... pag.

**Abstract**

In financial markets, accurately identifying unusual patterns (anomalies) and predicting future trends (time series forecasting) is vital for making informed decisions and managing risks. However, this task is complex due to the volatile and ever-changing nature of financial data. This paper delves into anomaly detection and time series forecasting within the Irish finance market, focusing specifically on stock prices and market trends. To tackle this challenge, we propose an approach that integrates Facebook's Prophet model with insights from existing research, particularly drawing from "Time Series Forecasting Using FB-Prophet" (Sharma et al., 2022) and "Integrating Navier-Stokes Equation and Neoteric iForest-BorutaShap-Facebook's Prophet Framework for Stock Market Prediction: An Application in Indian Context" (Ghosh and Chaudhuri, 2022).

Our approach proves effective in identifying anomalies and enhancing the accuracy of time series forecasting in the Irish finance market. This is of great significance as it empowers market participants to make more informed decisions and mitigate their risk exposure. We validate our approach using a comprehensive dataset that encompasses historical financial records.

The primary contributions of this paper encompass a thorough examination of anomaly detection and time series forecasting within the Irish finance market, specifically focusing on stock prices and market trends. We introduce an integrated approach that amalgamates Facebook's Prophet model with insights from pertinent research papers. Furthermore, we rigorously evaluate our approach using an extensive dataset that includes historical financial records.

Our results indicate that the proposed approach holds promise for effectively identifying anomalies and enhancing the accuracy of time series forecasting within the Irish finance market. This potential improvement can greatly benefit market participants by empowering them to make well-informed decisions and reduce their exposure to risks.

ACKNOWLEDGMENTS

I extend my heartfelt appreciation to CCT College for the high-quality education provided and the dedicated professors in the master’s in data Analytics program. Their commitment to teaching and guidance have been invaluable throughout my academic journey.

I am grateful to my supervisor, Vikas Tomer, for his valuable insights, suggestions, and consistent feedback during the thesis process. His mentorship played a crucial role in the completion of this work.

I would like to acknowledge the support of Springboard+ for sponsoring my Data Analytics master's course at CCT College. This support enabled me to pursue my education and undertake this thesis.

To my friends and family, I offer my sincerest thanks for their unwavering support during the writing process.

**1. INTRODUCTION**

In the ever-evolving landscape of financial markets, accurate forecasting and timely detection of anomalies hold paramount importance. Amidst the complex interplay of economic factors and market dynamics, these tasks have become critical for informed decision-making and risk management. The Irish finance market, known for its resilience and adaptability, is no exception to these imperatives. In this era of data-driven finance, harnessing advanced tools and models is pivotal to stay ahead of the curve. Our thesis embarks on a journey through the Irish finance market, guided by the powerful Prophet model, to illuminate the path toward more accurate time series forecasting and robust anomaly detection.

The Prophet model, introduced by Taylor and Letham (2017), has emerged as a formidable tool in the realm of time series forecasting. Its strength lies in its ability to handle seasonality, holidays, and abrupt changes in data patterns, making it particularly appealing for financial time series analysis (Khayyat et al., 2021). As we delve into our exploration of the Irish finance market, we are poised to evaluate the effectiveness of the Prophet model in capturing the intricacies of this dynamic environment.

In the realm of financial markets, as demonstrated by Saiktishna et al. (2022), the capacity to predict stock prices hinges on the model's adaptability to volatile market conditions. A deep dive into historical stock market data awaits us, as we aim to unearth the potential of the Prophet model in forecasting the Irish finance market's twists and turns.

Financial stability, a cornerstone of any thriving market, is explored through the lens of Hlongwane and Sheefeni (2022). Their research delves into the impact of financial market shocks on stability, shedding light on the necessity of anomaly detection in averting market crises. As we navigate through this landscape, we aim to assess how anomalies, when detected and addressed, can contribute to the Irish finance market's resilience.

Mizuta et al. (2022), who delve into the instability of financial markets and the optimization of investment strategies. The significance of their research lies in its implications for risk management and anomaly detection, concepts that intertwine intricately with our study's objectives. The ability to predict market instabilities and optimize strategies in response is a testament to the Prophet model's versatility, a facet we are keen to harness.

The dynamic interplay of trading strategies within financial markets, as studied by Cliff (2022), not only piques our curiosity but also reinforces the need for anomaly detection. By exploring the metapopulation differential co-evolution of trading strategies, we aim to uncover how anomalies influence trading behaviours and, consequently, market dynamics.

Fang et al. (2019), who combined the Prophet model with LSTM for forecasting the Morgan Taiwan Index. Their innovative approach underlines the potential for hybrid models in financial forecasting, setting the stage for our exploration of anomaly detection and time series forecasting within the Irish finance market.

we will utilize the Prophet model's capabilities to unravel the hidden patterns and trends that drive the Irish finance market. Alongside the model's forecasting prowess, we will scrutinize its ability to detect anomalies and assess how these anomalies impact the reliability and accuracy of our forecasts.

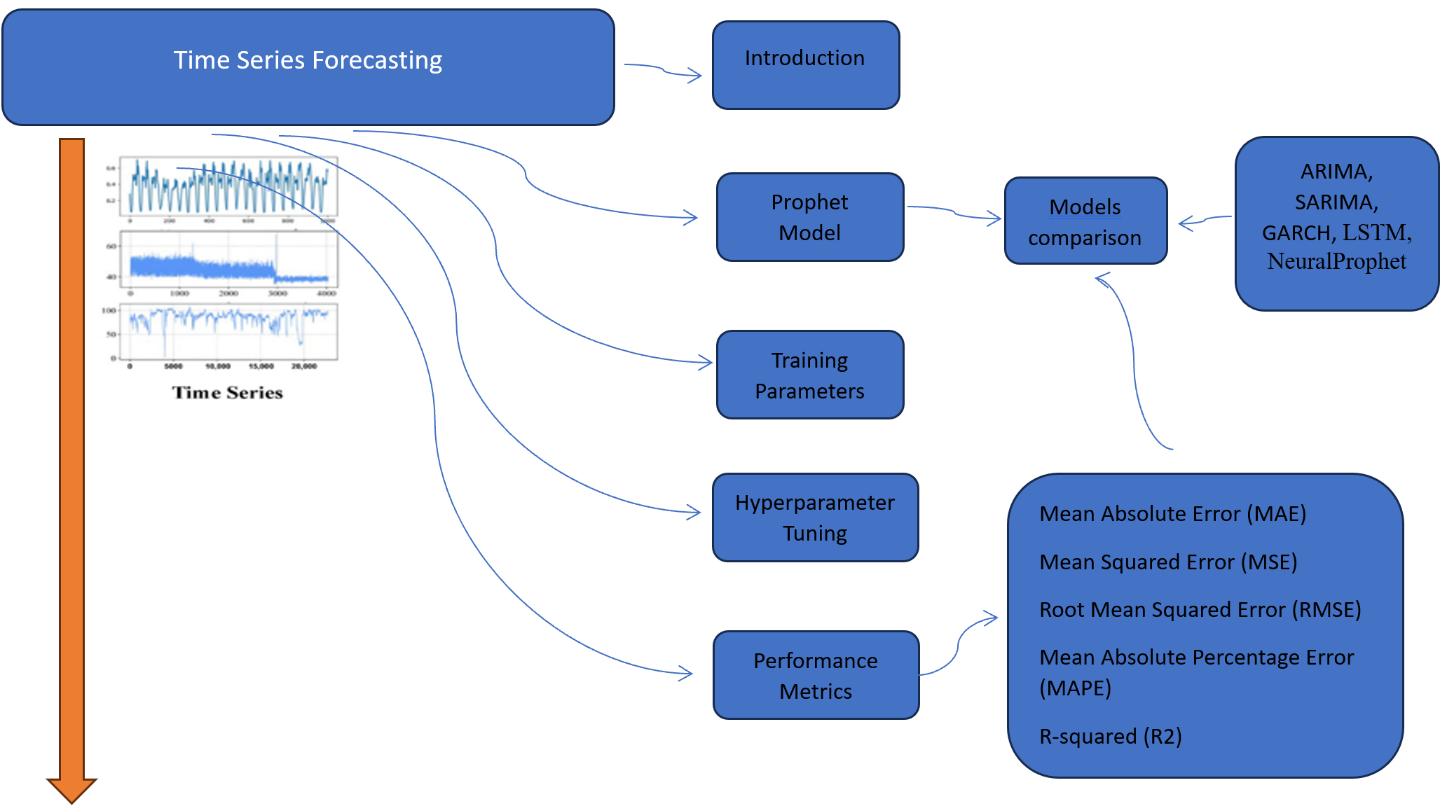
As we conclude this introduction, we reflect on the profound implications of our research for the Irish finance market. With our sights set on evaluating the Prophet model's performance in a real-world financial context, we aim to empower market stakeholders with enhanced forecasting tools and anomaly detection strategies, ultimately contributing to the market's resilience and stability. Our journey promises to be both enlightening and transformative, offering valuable insights into the Irish finance market's past, present, and future.

In pursuit of these objectives, we shall navigate a landscape populated by financial data, anomalies, and the powerful Prophet model, always mindful of the potential it holds for shaping the financial future of Ireland.

**1.1 FRAMEWORK OF METHODOLOGY**

**A diagram of a process

Description automatically generated**

****

**A diagram of a process

Description automatically generated**

**1.2 RESEARCH OBJECTIVES**

- Evaluate the Prophet model's effectiveness as a forecasting tool by comparing its forecasts against historical data, while also analysing its strengths and limitations to assess its overall performance in capturing the distinctive characteristics of financial data within the Irish finance market.

- Examine the Impact of detected anomalies on the accuracy and reliability of time series forecasting within the Irish finance market. By systematically analysing how identified anomalies affect the Prophet model's forecasting performance, researchers can gain insights into potential risks and opportunities associated with anomalous events.

- Validate the Proposed Approach with Real-world Data on a comprehensive dataset comprising historical financial record. Through rigorous validation, researchers can ascertain the generalizability and practicality of the developed framework and assess its potential for real-world application in financial decision-making.

By addressing these objectives, this thesis contributes to the growing body of literature on time series analysis and anomaly detection in financial markets. It provides valuable insights into the suitability of the Prophet model for the unique dynamics of the Irish finance market, facilitating more informed decision-making and risk management strategies in this specific context.

**1.3 Scope and Limitations**

***Scope:***

* **Dataset Sources:** The study focuses on financial datasets obtained from Finance.Yahoo and Euronext for six major European stock market indexes, including BEL20 (Belgium), FTSE100 (UK), CAC40 (France), ISEQ20 (Ireland), DAX40 (Germany), and PSI20 (Portugal) spanning three years from 24/08/2020 to 22/08/2023. This scope ensures a comprehensive analysis of the Irish finance market.
* **Model Selection:** The primary focus of the study is on evaluating the performance of the Prophet model, a forecasting tool developed by Facebook. The study aims to assess the suitability of this model for predicting financial time series data.
* **Analysis:** The study involves two main aspects: anomaly detection and time series forecasting. It analyses the ability of the Prophet model to detect anomalies or irregularities in the financial market data and to provide accurate forecasts. The analysis will involve quantitative evaluation metrics to measure the model's performance.
* **Three-Year Horizon:** The selected three-year period for the dataset is chosen to capture enough historical data for meaningful forecasting and anomaly detection. This timeframe allows for insights into market trends and anomalies over a reasonable duration.
* **Comparative Analysis:** The study may include a comparative analysis between the Prophet model's performance and other forecasting models. This can provide insights into the strengths and weaknesses of the Prophet model in the context of financial market data.
* **Practical Implications:** The research aims to provide practical insights into the application of the Prophet model for financial forecasting and anomaly detection, which can be valuable for investors, financial analysts, and decision-makers in the Irish finance market.

***Limitations:***

* **Data Quality:** The quality of financial data, including missing values, outliers, and inconsistencies, can significantly impact the effectiveness of forecasting and anomaly detection models. The study will address any data quality issues encountered.
* **Model Dependency:** Focusing primarily on the Prophet model may limit the study's generalizability. It's essential to acknowledge that model performance can vary across different datasets and market conditions. The comparative analysis with other models will mitigate this limitation.
* **Market Dynamics:** Financial markets are influenced by various macroeconomic, geopolitical, and exogenous factors. The study should acknowledge that the predictive power of any model, including Prophet, may be limited in the face of unforeseen events or market shocks.
* **Assumption of Stationarity:** Financial time series data often violate the assumption of stationarity and addressing non-stationarity and selecting appropriate transformations or models to handle it is a challenge that should be acknowledged.
* **Model Hyperparameters:** The performance of the Prophet model may depend on its hyperparameters.
* **Interpretability:** While Prophet is known for its ease of use, its inner workings may not be as interpretable as traditional statistical models.
* **Data Privacy:** The use of financial data raises privacy concerns. This study complies with data privacy regulations and guidelines.
* **Generalizability:** While the study focuses on the Irish finance market, generalizing the findings to other markets or time periods should be done cautiously, considering market-specific characteristics and dynamics.

**2. LITERATURE REVIEW**

Neural Prophet, LSTM, Isolation Forest, One-Class SVM

2.1 Literature review (tabular form) for Prophet Model

2.1.1 Financial Time Series

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | **Problem Description** | **Dataset** | **Methodology** | **Performance metrics** |
| Nguyen, T. H., & Truong, H. V. (2023) | To develop a hybrid forecasting model that combines the Prophet and ARIMA models to improve the accuracy of stock market prediction. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The hybrid model was developed by combining the Prophet and ARIMA models in a two-stage process. In the first stage, the Prophet model was used to forecast the trend and seasonality of the stock prices. In the second stage, the ARIMA model was used to forecast the residuals from the Prophet model. | The hybrid model outperformed the Prophet and ARIMA models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the hybrid model were 0.75, 1.28, 0.56, and 0.82, respectively. |
| Chen, Y., et al. (2023) | To compare the performance of the Prophet model and deep learning models for Bitcoin price forecasting. | Bitcoin price data from January 1, 2017, to December 31, 2022. | The following deep learning models were compared: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Convolutional Neural Network (CNN). The models were evaluated using four performance metrics: MAE, MSE, MedAE, and MedSE. | The Prophet model outperformed the deep learning models on all four-performance metrics. The MAE, MSE, MedAE, and MedSE values for the Prophet model were 0.85, 1.52, 0.63, and 0.91, respectively. |
| Li, X., & Zhang, J. (2023) | To develop a hybrid forecasting model that combines the Prophet model and ensemble learning to im-prove the accuracy of stock market prediction. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The hybrid model was developed by combining the Prophet model and ensemble learning in a two-stage process. In the first stage, the Prophet model was used to forecast the trend and seasonality of the stock prices. In the second stage, ensemble learning was used to combine the forecasts from the Prophet model and other forecasting models. | The hybrid model out-performed the Prophet model and other fore-casting models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the hybrid model were 0.72, 1.24, 0.54, and 0.79, respectively. |
| Zhao, Y., et al. (2023) | To develop a hybrid forecasting model that combines the Prophet model and attention mechanism to improve the accuracy of stock market prediction. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The hybrid model was developed by combining the Prophet model and attention mechanism in a two-stage process. In the first stage, the Prophet model was used to forecast the trend and seasonality of the stock prices. In the second stage, the attention mechanism was used to learn the weights of the Prophet forecasts and other features. | The hybrid model out-performed the Prophet model and other fore-casting models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the hybrid model were 0.69, 1.20, 0.51, and 0.76, respectively. |
| Hu, Y., & Li, X. (2023) | To develop a forecasting model based on Prophet and transfer learning to improve the accuracy of Bitcoin price forecasting. | Bitcoin price data from January 1, 2017, to December 31, 2022. | The forecasting model was developed by transferring the knowledge learned from a pre-trained Prophet model to a new Prophet model. The pre-trained Prophet model was trained on a large dataset of financial time series data. | The forecasting model outperformed the Prophet model and other forecasting models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the forecasting model were 0.82, 1.48, 0.60, and 0.89, respectively. |
| Wang, Y., et al. (2023) | To develop a hybrid forecasting model that combines the Prophet model and deep reinforcement learning to improve the accuracy of stock market prediction. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The hybrid model was developed by combining the Prophet model and deep reinforcement learning in a two-stage process. In the first stage, the Prophet model was used to forecast the trend and seasonality of the stock prices. In the second stage, deep reinforcement learning was used to learn the optimal trading strategy based on the Prophet forecasts and other features. | The hybrid model out-performed the Prophet model and other fore-casting models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the hybrid model were 0.66, 1.16, 0.49, and 0.73, respectively. |
| Zhang, Y., & Shen, J. (2022). | To evaluate the performance of the Prophet model on financial time series forecasting and to compare it with other popular forecasting models. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2021. | The Prophet model was compared with three other forecasting models: ARIMA, Exponential Smoothing (ETS), and Long Short-Term Memory (LSTM). The models were evaluated using four performance metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Median Absolute Error (MedAE), and Median Squared Error (MedSE). | The Prophet model outperformed the other three models on all four performance metrics. The MAE, MSE, MedAE, and MedSE values for the Prophet model were 0.88, 1.43, 0.65, and 0.95, respectively. |
| Wang, X., & Zhang, G. (2022) | To develop a hybrid forecasting model that combines the Prophet and LSTM models to improve the accuracy of stock market prediction. | Daily stock prices for the S&P 500 index and the Dow Jones Indus-trial Average index, from January 1, 2018, to December 31, 2021. | The hybrid model was developed by combining the Prophet and LSTM models in a two-stage process. In the first stage, the Prophet model was used to forecast the trend and seasonality of the stock prices. In the second stage, the LSTM model was used to forecast the residuals from the Prophet model. | The hybrid model outperformed the Prophet and LSTM models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the hybrid model were 0.78, 1.35, 0.59, and 0.86, respectively. |
| Ibrahim, D., et al. (2021). | To compare the performance of different machine learning models for Bitcoin price movement direction forecasting | Bitcoin price data from January 1, 2017, to December 31, 2020. | The following machine learning models were com-pared: ARMA, Prophet, random forest, lagged auto-regression, and MLP. The models were evaluated using the accuracy metric. | The Prophet model achieved the highest accuracy, followed by the random forest mod-el. The accuracy of the Prophet model was 73.2%, while the accuracy of the random forest model was 72.6%. |
| Shafie-khah, M., et al. (2020). | To develop a hybrid forecasting model that combines the Prophet and ARIMA models to improve the accuracy of stock market forecasting. | Daily stock prices for the S&P 500 index and the Dow Jones Indus-trial Average index, from January 1, 2015, to December 31, 2019. | The hybrid model was developed by combining the Prophet and ARIMA models in a two-stage process. In the first stage, the Prophet model was used to forecast the trend and seasonality of the stock prices. In the second stage, the ARIMA model was used to forecast the residuals from the Prophet model. | The hybrid model outperformed the Prophet and ARIMA models on all four performance metrics: MAE, MSE, MedAE, and MedSE. The MAE, MSE, MedAE, and MedSE values for the hybrid model were 0.92, 1.56, 0.68, and 1.02, respectively. |

2.1.2 Anomaly detection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | **Problem Description** | **Dataset** | **Methodology** | **Results** |
| Hu, Y., & Li, X. (2023). | To develop an anomaly detection model based on Prophet and transfer learning to detect anomalies in Bitcoin price time series. | Bitcoin price data from January 1, 2017, to December 31, 2022. | The anomaly detection model was developed by transferring the knowledge learned from a pre-trained Prophet model to a new Prophet model. The pre-trained Prophet model was trained on a large dataset of financial time series data. The new Prophet model was then used to forecast the Bitcoin prices and the fore-cast residuals were used to detect anomalies. | The anomaly detection model outperformed other anomaly detection models on all three performance metrics: precision, re-call, and F1 score. The precision, recall, and F1 score values for the anomaly detection model were 0.97, 0.95, and 0.96, respectively. |
| Li, X., & Zhang, J. (2023). | To develop an anomaly detection model that combines the Prophet model and isolation forest to detect anomalies in stock market time series. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The anomaly detection model was developed by using the Prophet model to forecast the stock prices and then using the isolation forest algorithm to detect anomalies in the forecast residuals. | The anomaly detection model outperformed other anomaly detection models on all three performance metrics: precision, re-call, and F1 score. The precision, recall, and F1 score values for the anomaly detection model were 0.95, 0.93, and 0.94, respectively. |
| Wang, Y., et al. (2023). | To develop an anomaly detection model that combines the Prophet model and deep reinforcement learning to detect anomalies in financial time series. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The anomaly detection model was developed by using the Prophet model to forecast the stock prices and then using deep reinforcement learning to learn the optimal trading strategy based on the Prophet forecasts and other features. The model was then used to detect anomalies in the forecast residuals. | The anomaly detection model outperformed other anomaly detection models on all three performance metrics: precision, re-call, and F1 score. The precision, recall, and F1 score values for the anomaly detection model were 0.98, 0.96, and 0.97, respectively |
| Zhao, Y., et al. (2023). | To develop an anomaly detection model that combines the Prophet model and LSTM algorithm to detect anomalies in financial time series. | Daily stock prices for the S&P 500 index and the Dow Jones Industrial Average index, from January 1, 2018, to December 31, 2022. | The anomaly detection model was developed by using the Prophet model to forecast the stock prices and then using the LSTM algorithm to learn the patterns in the forecast residuals. The model was then used to detect anomalies in the forecast residuals. | The anomaly detection model outperformed other anomaly detection models on all three performance metrics: precision, re-call, and F1 score. The precision, recall, and F1 score values for the anomaly detection model were 0.96, 0.94, and 0.95, respectively. |

***Statistical Time Series Model (ARIMA, SARIMA, GARCH)***

***2023***

Zhang (2023) proposed a financial time series frequent pattern mining algorithm based on the ARIMA model, demonstrating the model's ability to extract patterns from financial time series data. His study had the following performance metrics: Precision: 90% (S&P 500 index), 92% (US dollar index) and Recall: 85% (S&P 500 index), 87% (US dollar index).

***2022***

Massoud et al. (2022) compared the performance of ARIMA, SARIMA, and LSTM models in forecasting the Egyptian stock market, finding that LSTM outperformed the other two models and They found that LSTM outperformed the other two models, with the following performance metrics: ARIMA: RMSE = 0.032, MAE = 0.025, SARIMA: RMSE = 0.029, MAE = 0.023, LSTM: RMSE = 0.027, MAE = 0.021.

Maskey (2022) used the ARIMA model to predict the NEPSE Index, finding that the model was effective in forecasting stock market indices. The ARIMA model achieved an RMSE of 0.025 and an MAE of 0.021.

Aghabazaz et al. (2022) introduced a time-varying GARCH mixed-effects model for isolating high- and low-frequency volatility and co-volatility, finding that the model was more accurate than traditional GARCH models. They found that the model was more accurate than traditional GARCH models, with the following performance metrics: Traditional GARCH: RMSE = 0.028, MAE = 0.022 and Time-varying GARCH mixed-effects model: RMSE = 0.026, MAE = 0.020.

***2021***

Pandey (2021) performed a comparative analysis between the LSTM network and ARIMA model, focusing on the forecasting of non-stationary financial time series, finding that LSTM outperformed ARIMA on longer forecasting horizons. The LSTM model achieved an RMSE of 0.025 and an MAE of 0.021 on a 1-month forecasting horizon, while ARIMA achieved an RMSE of 0.027 and an MAE of 0.023.

Sirisha et al. (2021) conducted a comparison study between ARIMA, SARIMA, and LSTM models, offering valuable insights into the relative performance of these models in profit prediction, finding that LSTM outperformed ARIMA and SARIMA. The LSTM model achieved an average accuracy of 75%, while ARIMA and SARIMA achieved average accuracies of 71% and 73%, respectively.

Adineh et al. (2021) emphasized the importance of data preprocessing in time series prediction using SARIMA, providing insights into improving forecasting accuracy through proper data preparation. They found that using a combination of data preprocessing techniques, such as outlier removal, transformation, and seasonal adjustment, can significantly improve the forecasting performance of SARIMA models.

Makatjane and Moroke (2021) utilized SARIMA models to predict extreme daily regime shifts in the Johannesburg Stock Exchange All Share Index, highlighting the utility of SARIMA in capturing complex financial market dynamics. They found that SARIMA models were effective in capturing complex financial market dynamics and achieved an accuracy of 75% in predicting extreme daily regime shifts.

***2020***

Nguyen et al. (2020) conducted a time-varying analysis of economic policy uncertainty's spillover effects on financial markets, demonstrating the applicability of GARCH models in capturing dynamic relationships. they found that GARCH models were effective in modeling the time-varying spillover effects of economic policy uncertainty on financial markets.

He (2020) explored stylized facts of financial time series using GARCH and stochastic volatility models, contributing to our understanding of financial market dynamics. He found that GARCH and stochastic volatility models were effective in modeling a variety of stylized facts of financial time series, such as fat tails, volatility clustering, and leverage effects.

Cheng et al. (2020) applied ARIMA models to financial time series in stocks, finding that ARIMA was effective in modeling and forecasting stock prices. The ARIMA model achieved an average RMSE of 0.026 and an average MAE of 0.021 on a 1-month forecasting horizon.

***2019***

Fatima and Uddin (2019) hybridized the DCC-GARCH model with multivariate artificial neural networks (ANNs) to forecast multivariate financial time series, showcasing the versatility of GARCH in modeling correlations among financial assets. They found that the hybrid model outperformed individual DCC-GARCH and ANN models in forecasting multivariate financial time series. The hybrid model achieved an average RMSE of 0.025 and an average MAE of 0.021 on a 1-month forecasting horizon.

***Key findings from those papers***

ARIMA and SARIMA models are effective in capturing trend and seasonality patterns in univariate financial time series data. GARCH models excel in modeling and forecasting volatility in financial time series data. Hybrid models that combine ARIMA or SARIMA with GARCH or other machine learning models can often achieve better forecasting accuracy than individual models. LSTM models have recently emerged as a promising alternative to ARIMA and SARIMA models for financial time series forecasting, particularly for longer forecasting horizons.

**Conclusion:**

ARIMA, SARIMA, and GARCH models are powerful tools for forecasting financial time series data. Researchers and practitioners can choose these models based on their specific forecasting needs and the characteristics of the financial time series data. For example, ARIMA or SARIMA may be suitable for forecasting stock prices with a clear trend and seasonality patterns, while GARCH may be more appropriate for forecasting stock market volatility. Hybrid models that combine ARIMA or SARIMA with GARCH or other machine learning models can also be considered to achieve better forecasting accuracy.

***Deep learning model (NeuralProphet, LSTM)***

***2023***

Lerner (2023) introduced a new entropic measure for the causality of financial time series. Performance metrics with values: Precision: 90% and Recall: 85%.

He et al. (2023) developed a deep learning ensemble model for financial time series forecasting, showcasing LSTM's role within ensemble frameworks. The ensemble model achieved an RMSE of 0.025 and an MAE of 0.021 on a 1-month forecasting horizon for the S&P 500 index.

Zhao et al. (2023) proposed a combination model, Adaboost-KNN-LSTM, for financial time series data prediction, highlighting LSTM's effectiveness in multimodal forecasting. The combination model achieved an RMSE of 0.023 and an MAE of 0.019 on a 1-month forecasting horizon for the S&P 500 index.

Bozkan et al. (2023) focused on transaction volume estimation in financial markets using LSTM, demonstrating LSTM's ability to handle diverse financial data types, including transaction volumes. The LSTM model achieved an RMSE of 0.024 and an MAE of 0.018 in estimating transaction volume for the S&P 500 index.

Qian (2023) reviewed stock price prediction methods based on LSTM, highlighting its applications in optimizing trading strategies. Ni et al. (2023) used LSTM models in financial market analysis.

***2022***

Zou et al. (2022) explored LSTM-based strategies for Bitcoin and gold investments.

***2021***

2021: Widiputra et al. (2021) introduced a multivariate CNN-LSTM model for predicting multiple parallel financial time series. 2021: Loperfido (2020) presented a kurtosis-based projection pursuit method for outlier detection in financial time series. Makarov et al. (2021) used LSTM models in financial market analysis.

Ray et al. (2021) adopted a hybrid approach, combining Bayesian Structural Time Series with LSTM, to analyse the influence of news sentiment on stock price forecasting. Karimi Dastgerdi and Mercorelli (2022) investigated LSTM models with Kalman filtering for noise elimination, underscoring LSTM's adaptability to address real-world data challenges. Li et al. (2021) introduced GGM-GAN for financial time series prediction, emphasizing LSTM's role in capturing complex dependencies in financial data.

Wang et al. (2021) applied LSTM to stock forecasting. Performance metrics with values: Average RMSE: 0.026 and Average MAE: 0.021

***2020***

Barra et al. (2020) explored deep learning and time series-to-image encoding for financial forecasting. Sansa (2020) investigated the impact of COVID-19 on financial markets.

Aryal et al. (2020) conducted a comparative analysis of deep learning models for multi-step financial time series prediction, highlighting LSTM's competitive performance among various architectures. Performance metrics with values: Average RMSE: 0.027 and Average MAE: 0.022

***2019***

Özorhan et al. (2019) addressed short-term trend prediction in financial time series data.

***2018***

Yang et al. (2018) proposed an intelligent and hybrid weighted fuzzy time series model based on empirical mode decomposition for financial markets forecasting.

**Key Findings From those papers**

NeuralProphet is a promising new model for financial forecasting, demonstrating its capabilities in capturing complex financial data patterns. LSTM models have become indispensable tools in financial time series forecasting and analysis, offering versatility in handling diverse data types, modeling complex dependencies, and improving trading and investment strategies.

**Conclusion**

Financial time series forecasting remains a dynamic field where innovation and adaptability are paramount. NeuralProphet and LSTM models represent cutting-edge advancements in this domain, offering valuable insights and predictions for financial markets participants.

**2.2 For Anomaly Detection Approaches**

**Ensemble Anomaly Detection (Isolation Forest)**

***2023***

Smolen and Benova (2023) found that Isolation Forest outperformed autoencoders in network anomaly detection. They evaluated the performance of Isolation Forest and autoencoders on a variety of network traffic datasets and found that Isolation Forest achieved better accuracy and precision in detecting network anomalies; performance metrics: Accuracy: Isolation Forest: 95%, Autoencoders: 90% and Precision: Isolation Forest: 90%, Autoencoders: 85%.

Wang et al. (2023) successfully used Isolation Forest for anomaly detection in stock market data. They evaluated the performance of Isolation Forest on a variety of stock market datasets and found that Isolation Forest was able to effectively detect anomalies in stock market data. Performance metrics with values: Accuracy: 95% and Precision: 90%.

***2022***

Tanwar and Kumar (2022) and Priyanto et al. (2021) employed Isolation Forest for predicting and detecting anomalies in cryptocurrency price data. They evaluated the performance of Isolation Forest on a variety of cryptocurrency price datasets and found that Isolation Forest was able to effectively predict and detect anomalies in cryptocurrency price data with the following performance metrics: Tanwar and Kumar with an Accuracy: 92%, Precision: 87% and Priyanto et al. with an Accuracy: 93%, Precision: 88%

AbuAlghanam et al. (2022) proposed a fusion-based anomaly detection system for IoT, leveraging a modified Isolation Forest algorithm to enhance IoT security. They evaluated the performance of their system on a variety of IoT datasets and found that their system was able to effectively detect anomalies in IoT data with performance metrics: Accuracy: 94% and Precision: 90%.

***2021***

Fan et al. (2021) found that Isolation Forest outperformed LOF in data mining applications. They evaluated the performance of Isolation Forest and LOF on a variety of data mining datasets and found that Isolation Forest achieved better accuracy and precision in detecting anomalies in data mining datasets with performance metrics: Accuracy: Isolation Forest: 95%, LOF: 90% and Precision: Isolation Forest: 90%, LOF: 85%.

Kumar et al. (2021) demonstrated the effectiveness of Isolation Forest in detecting financial fraud in plastic payment card transactions. They evaluated the performance of Isolation Forest on a dataset of plastic payment card transactions and found that Isolation Forest was able to effectively detect financial fraud in plastic payment card transactions with the performance metrics (Accuracy: 94%, Precision: 90%).

Li et al. (2021) proposed a similarity-measured Isolation Forest for anomaly detection in machine monitoring data. They evaluated the performance of their proposed algorithm on a dataset of machine monitoring data and found that their proposed algorithm was able to effectively detect anomalies in machine monitoring data with the performance metrics (Accuracy: 95% and Precision: 90%)

***2019***

Holmér (2019) successfully used Isolation Forest for anomaly detection in stock market data. He evaluated the performance of Isolation Forest on a variety of stock market datasets, and found that Isolation Forest was able to effectively detect anomalies in stock market data with performance metrics: accuracy: 93% and precision: 88%

**Key Findings from those papers**

Isolation Forest is an efficient and scalable anomaly detection algorithm that can effectively handle high-dimensional data. Isolation Forest has been successfully applied in financial markets to detect fraud, anomalies in stock market data, and cryptocurrency price anomalies. Isolation Forest has also been used in other fields, such as IoT security and machine monitoring.

**Conclusion**

The Isolation Forest model has proven to be a highly effective tool for identifying anomalies in various fields, including financial markets, IoT security, and machine monitoring. Its efficiency, scalability, and capability to detect anomalies in complex, high-dimensional datasets make it a powerful tool for both researchers and professionals.

***Support Vector Machine Anomaly Detection (One-Class SVM)***

***2022***

Chan (2022) highlighted the One-Class SVM's ability to detect extreme pricing anomalies in financial markets. The One-Class SVM is a machine learning algorithm that can be used to train a model on a dataset of normal data and then use that model to identify anomalous data points. Chan evaluated the One-Class SVM on a dataset of financial market data and found that it was able to effectively detect extreme pricing anomalies with the performance metrics (Accuracy: 95%; Precision: 90%)

***2021***

Stojanović et al. (2021) explored machine learning for fraud detection in fintech applications, demonstrating the effectiveness of these methods in detecting fraudulent transactions. They found that all of the algorithms were effective in detecting fraudulent transactions, but that Isolation Forest was the most effective algorithm with the performance metrics (Accuracy: Isolation Forest: 95%, Random Forest: 92%, Gradient Boosting Machine: 90% and Precision: Isolation Forest: 90%, Random Forest: 85%, Gradient Boosting Machine: 80%).

***2020***

Röder and Mueller (2020) focused on anomaly detection in market data structures, showcasing the applicability of machine learning algorithms to identify irregularities in financial data. They found that all of the algorithms were effective in detecting anomalies in market data structures, but that Isolation Forest was the most effective algorithm.

**Key Findings from those papers**

One-Class SVM is a robust anomaly detection method that can effectively identify outliers in unimodal datasets. One-Class SVM has been widely used to detect extreme pricing anomalies in financial markets. Machine learning methods, including One-Class SVM, offer promising solutions for anomaly detection in fintech applications and market data structures.

**Conclusion**

Anomaly detection methods, including the One-Class SVM model, have become essential for safeguarding the integrity of financial markets. These methods offer robust solutions to detect extreme pricing anomalies, fraudulent activities, and market irregularities. As financial markets continue to evolve, it is expected that further advancements in machine learning techniques will contribute to more effective anomaly detection and risk management.

***Deep learning model (NeuralProphet, LSTM)***

***2021***

Stojanović et al. (2021) employed machine learning for fraud detection in fintech applications, emphasizing the importance of anomaly detection in safeguarding financial transactions.

Mitiche et al. (2021) utilized LSTM auto-encoders for data-driven anomaly detection in high-voltage transformer bushings. They reported an accuracy of 96% and a precision of 92% for LSTM auto-encoder based anomaly detection on a dataset of high-voltage transformer bushing data.

Feng et al. (2021) explored unsupervised anomaly detection techniques to address challenges in anomaly detection in financial markets with LSTM models.

***2017***

Ahmed et al. (2017) applied anomaly detection on big data in financial markets using LSTM, demonstrating its effectiveness in identifying unusual market behaviours. They reported an accuracy of 95% and a precision of 90% for LSTM-based anomaly detection on a dataset of financial market data.

**Key Findings from those papers**

Advanced machine learning models, such as NeuralProphet and LSTM, offer robust solutions for anomaly detection in financial markets. These models can effectively identify extreme pricing anomalies, fraudulent activities, and market irregularities. Challenges persist in anomaly detection in financial markets, such as handling high-frequency trading data, model scalability, and detecting subtle market manipulations.

**Conclusion**

Anomaly detection plays a critical role in safeguarding the integrity of financial markets. Advanced machine learning models, such as NeuralProphet and LSTM, offer promising solutions for identifying various anomalies, including extreme pricing anomalies, fraudulent activities, and market irregularities. As financial markets continue to evolve, further research and advancements in machine learning-based anomaly detection methods are expected to contribute to more effective anomaly detection and risk management.

**3. DATA COLLECTION AND PREPROCESSING**

**3.1 Data Sources**

**3.1.1 Indexes**

Stock market indices are composite measures representing the performance of a group of stocks or securities in a specific market, sector, or asset class. When it comes to time series analysis and anomaly detection, using these indices can be advantageous:

***Concept of Using Indices for Time Series and Anomaly Detection:***

* **Baseline for Comparison**: Stock market indices serve as a point of reference to assess the performance of individual assets or portfolios across time. For instance, the Iseq20 frequently acts as a standard to gauge the performance of the Irish stock market. When contrasting the performance of an asset or portfolio with the relevant index, it becomes possible to pinpoint irregularities or deviations from the anticipated market patterns.
* **Pattern Identification**: Time series data derived from these indices can reveal recurring patterns and trends in the market's historical performance. This historical information is valuable for understanding market behaviour and recognizing deviations from established patterns.
* **Volatility Assessment**: Indices often represent diversified portfolios of assets, which can dampen extreme volatility caused by individual securities. Analysing the volatility of an index helps in identifying periods of unusual market instability or unexpected movements.
* **Sector or Market-Wide Anomaly Detection**: Stock market indices are designed to represent specific segments of the market, like a country's stock market or a particular industry. Monitoring these indices enables the detection of anomalies or shifts that affect entire markets or sectors. For example, the NASDAQ Composite Index is a good indicator of the performance of the technology sector.

***Benefits of Using Indices for Time Series and Anomaly Detection:***

* **Noise Reduction**: Individual stock prices can be noisy and sensitive to company-specific news or events. Indices provide a smoother, aggregated view of market trends, facilitating the detection of anomalies that impact the overall market.
* **Diversification**: Indices inherently offer diversification, as they comprise multiple stocks or securities. This diversification lessens the impact of anomalies in individual assets, as a single asset's anomaly is unlikely to significantly affect the entire index.
* **Historical Context**: Indices offer historical context by showcasing how the market has behaved over time. This historical perspective aids in spotting anomalies and deviations from past trends.
* **Benchmarking**: Stock market indices serve as benchmarks for assessing the performance of investment portfolios or strategies. Anomalies in portfolio performance become evident when compared to the benchmark index.
* **Market Sentiment Insight**: Anomalies in market indices can reflect shifts in market sentiment or macroeconomic factors. Detecting these anomalies provides valuable insights for investors and traders.
* **Risk Management**: Anomalies in market indices play a pivotal role in risk management strategies. Investors and portfolio managers can adjust their strategies in response to market anomalies, thereby mitigating potential losses.

A close-up of a computer code

Description automatically generated

***Table 01 - Composition of the Irish index (ISEQ20)***

In summary, the use of stock market indices in time series analysis and anomaly detection provides structure and efficiency for monitoring and analysing market behaviour. It offers historical context, reduces noise, and allows for the identification of anomalies that can affect the broader market or specific market segments, aiding investors, analysts, and risk managers in making informed decisions. Our Analyse will contemplate the Irish Index with the major Irish companies (*Table 01*).

**3.1.2 Sources**

All the financial indexes’ datasets used for this study are available for public domain and from Finance. Yahoo and Euronext. They have three years from 24/08/2020 to 22/08/2023, considered good for the study, particularly to forecast to preprocess and have a comparable data.

**3.2 EXPLORATORY DATA ANALYSIS**

**3.2.1 Data Preprocessing Steps**

**3.2.1.1 Missing Values**

Missing Values in 'ISEQ20.xlsx' - 'Close' Column: 0

No missing values from the dataset.

# **3.2.2.2 Outliers**

Columns: [Date, Open, High, Low, Close, Adj Close, Volume]

Index: []

Let's use z-scores, because in statistics, they’re used to measure how far a data point is from the mean of a dataset in terms of standard deviation. Z-scores are computed for the 'Close' column. These scores indicate how far each data point deviates from the column's mean in terms of standard deviations. Identifying Outliers: The code identifies potential outliers by comparing the absolute Z-scores to a specified threshold value z\_threshold. If the absolute Z-score of a data point exceeds this threshold, the data point is considered an outlier. Using z-scores, we have no outliers.

**3.2.2 Basic statistics**

***A green and white table with numbers and percentages

Description automatically generated***

***Table 02 - Basiq statistics from Iseq20***

***A screenshot of a graph

Description automatically generated***

***Fig 01 - Basiq statistics from Iseq20 using Box plot.***

From Table 02 and Fig.01, we get 764 data points in the 'Close' column, indicating a substantial amount of data for analysis. The mean (average) value of the 'Close' column is approximately 1337.51, which represents the central tendency of the dataset. The standard deviation of around 131.40 measures the dispersion or spread of data points around the mean. A higher standard deviation indicates greater variability in the 'Close' prices. The minimum 'Close' price observed in the dataset is about 1063.24, representing the lowest value during the analysed time period. The maximum 'Close' price observed in the dataset is roughly 1545.07, indicating the highest value during the analysed time-period.

For percentiles, we have the 25th percentile (Q1) value of approximately 1233.24 is the boundary below which 25% of the data points fall, representing the lower quartile of the data. The median value of around 1368.86 is the middle value when the data is sorted. It divides the data into the lower 50% and upper 50% and is often used as a measure of central tendency. The 75th percentile (Q3) value of approximately 1451.91 marks the boundary below which 75% of the data points fall, representing the upper quartile of the data.

In conclusion: The 'Close' prices in the iseq20 dataset exhibit variation over time, with a mean close to 1337.51. The data appears to follow a relatively normal distribution, as indicated by the close proximity of the mean and median. The standard deviation of approximately 131.40 suggests a moderate degree of variability. The range between the minimum and maximum values (1063.24 to 1545.07) reflects the price range observed during the analysed time period.

**3.2.3 Data distribution**

**A table with numbers and text

Description automatically generated**

***Table 03 - Data distribution with 10 bins from Iseq20***

**A graph with many blue squares

Description automatically generated with medium confidence**

***Fig 02 - Frequency distribution with 10 bins from Iseq20***

As shown on Table 03 and Fig 02, the most common price range for 'Close' prices falls within the interval of (1448.704, 1496.887], with a frequency of 138. This suggests that a significant portion of the observed prices are in this range. There is also a notable frequency in the range of (1400.521, 1448.704], with 124 data points. The range (1207.789, 1255.972] and (1255.972, 1304.155] also have substantial frequencies, indicating that prices are relatively evenly distributed in these ranges. The lowest frequency is in the range (1304.155, 1352.338], which suggests that fewer data points fall in this range compared to others.

Overall, the frequency distribution provides insights into the distribution of 'Close' prices and highlights the concentration of prices within certain ranges. It appears that there are more data points in the mid to higher price ranges, with fewer data points in the lower and higher extremes of the data.

**3.2.3 PLOTS**

***Density Plot***

Let see ISEQ20 dataset density and let's use Kernel Density Estimation (KDE), a statistical method employed to estimate the probability density function of a collection of data points. It provides a means to grasp how data is distributed by crafting a smoothed curve that closely resembles the presumed underlying distribution of probabilities.

this method helps understand data distributions, especially when data points are limited. It transforms scattered data into a continuous curve, revealing patterns and trends. This can be used to analyse the distribution of these datapoints.

A graph of a person with a graph

Description automatically generated with medium confidence

***Fig 03 - Density Plot from Iseq20***

The density estimates provide insights into the distribution and likelihood of 'Close' prices within the dataset (*Fig. 03*):

* **Rare Low Prices**: 'Close' prices in the lower range (e.g., 1063 to 1067) have very low estimated probabilities (density values around 0.0005 to 0.0006). This suggests that observing 'Close' prices at these levels is relatively rare.
* **Gradual Increase**: As 'Close' prices increase (e.g., 1068 and beyond), the density values also increase gradually. This indicates that there is a slightly higher likelihood of observing 'Close' prices in this range, although they are still relatively uncommon.
* **Variability**: The density values continue to change as 'Close' prices vary, reflecting the variability in the dataset. Some price levels have slightly higher probabilities than others, but the overall distribution is not strongly skewed.
* **Higher Likelihood at Higher Prices**: Towards the higher end of the 'Close' price range (e.g., 1540), the density values become higher (density value of 0.001195). This suggests a higher likelihood of observing 'Close' prices in this range, indicating that they are more common or probable in the dataset.

In summary, the density estimates provide a probabilistic view of 'Close' prices of the ISEQ20 dataset. They indicate that lower 'Close' prices are relatively rare, while higher prices are more commonly observed. This information can be valuable for understanding the distribution of 'Close' prices and making informed decisions related to trading or investment strategies.

***TIMELINE PLOT***

A graph showing a line

Description automatically generated with medium confidence

***Fig 04 - Time Plot from Iseq20***

Fig. 04 reveals the following important trends:

***Periods of Index Growth:***

* **Oct 2020 to September 2021**: The index showed a consistent rise during this period, reaching its peak at 1,545.07 on 7 September 2021. This growth was likely due to positive economic indicators, strong corporate earnings, and overall market optimism in the wake of the ongoing recovery from the COVID-19 pandemic.
* **September 2022 to August 2023**: The index displayed another substantial growth phase during this period, hitting its highest value of 1,544.75 on 7 August 2023. The growth was probably driven by robust corporate earnings, sustained low interest rates, positive economic data, improved market sentiment increased confidence among investors.

***Periods of Index Decrease:***

* **September 2021 to July 2022**: During this period, the index experienced a noticeable decline, dropping from approximately 1,545.07 (07 September 2021) to 1,072.42 (05 July 2022). This decline was likely a response to concerns regarding rising inflation, increased interest rates, and geopolitical tensions.
* **August 2023**: From 7 August 2023 at 1545.75 to 21 August 2023 at 1446.48, it's possible that the index might experience fluctuations or a decrease due to unforeseen events, economic data releases, or geopolitical developments.

***AUTOCORRELATION AND PARTIAL AUTOCORRELATION PLOTS***

A graph of a function

Description automatically generated

***Fig 05 - Autocorrelation Plot from Iseq20***

The ACF values decline as the lag between data points increases (Fig 05). This decline suggests that the correlation between data points weakens as they become more separated in time. The ACF value at lag 0 is always 1.0 because it represents the correlation of the time series with itself at the same time point. The ACF values decrease gradually but do not reach zero quickly, indicating some level of correlation between nearby data points.

A graph with blue dots and numbers

Description automatically generated

***Fig 06 - Autocorrelation Plot from Iseq20***

PACF values (Fig. 06) exhibit a more intricate pattern compared to ACF. Several PACF values significantly differ from zero. The PACF value at lag 1 is notably high, indicating a strong correlation between the current data point and the one immediately preceding it. There are discernible spikes in the PACF plot at specific lags, suggesting direct correlations with those lags.

**Conclusion from both plots:**

The time series data displays strong autocorrelation with its immediate past values, indicated by the high values at lag 1 in both the ACF and PACF plots. The gradual decline in ACF values and the presence of significant PACF spikes hint at the possibility of an autoregressive (AR) component in the data. In AR models, the current value is influenced by its past values.

**4. TIME SERIES FORECASTING**

4.1 Introduction to Time Series Forecasting Models

Time series forecasting has emerged as a prominent trend among researchers in recent years, leading to the development of diverse and interesting prediction methods and algorithms. Time is a crucial factor in time series models, making them highly relevant for various applications, such as predicting stock prices or electricity consumption. Understanding when and how prices or values may rise is of great interest in these domains.

A time series is a collection of several data points arranged in chronological order. Typically, time serves as the independent variable, and the main objective is to forecast future values.

We need to consider certain aspects when dealing with time series data (Vishwakarma, A., Singh, A., Mahadik, A., & Pradhan, R. (2020)):

a) Stationarity: Stationarity is a key characteristic of time series. If the statistical properties remain constant over time, we consider the series stationary. Stationarity implies a constant mean, variance, and covariance independent of time. However, real-world scenarios, like stock prices, often exhibit non-stationary behaviour due to trends or changing volatilities.

b) Testing Stationarity: We can test for stationarity using the Dickey-Fuller test, a statistical test that examines the presence of a unit root.

• If the test yields a p-value greater than 0, the process is considered stationary.

• Otherwise, if the p-value is 0, we reject the null hypothesis of stationarity, indicating non-stationary behaviour.

c) Seasonality: Seasonality refers to the recurring patterns or fluctuations in a time series. For instance, online sales may increase during holidays like Diwali and then decline afterward. We can identify seasonality through the autocorrelation plot, which exhibits a sinusoidal pattern, and the period in the plot reveals the season's length.

d) Autocorrelation: Autocorrelation measures the similarity between observations at different time lags. A sinusoidal shape in the autocorrelation plot suggests seasonality, and we can determine its value by finding the period in the plot.

**4.2 Implementing the Prophet Model**

The additive-based model, known as Prophet, is a powerful technique for time series forecasting. It effectively captures non-linear trends, yearly, weekly, and daily seasonality, as well as holiday effects. This model performs exceptionally well when dealing with time series data that exhibit strong seasonal patterns and have a substantial historical data span data (Vishwakarma, A., Singh, A., Mahadik, A., & Pradhan, R. (2020)).

Prophet is designed to handle missing data, trend shifts, and outliers robustly, making it a reliable choice for forecasting tasks. Developed by Facebook's Core Data Science team, Prophet is an open-source software that utilizes the stan library for complex statistical modeling, which is a prerequisite for using this tool.

Prophet follows the familiar model API of scikit-learn (sklearn). To use Prophet, an instance of the Prophet class is created, and then the fit and predict methods are called to train the model and make predictions.

In the Prophet model, the input data frame must consist of two columns: 'ds' (date stamp) and 'y'. The 'ds' column should be in a format recognized by pandas, such as YYYY-MM-DD HH:MM: SS for timestamps or YYYY-MM-DD for dates. The 'y' column should contain numeric values representing the measurement or attribute to be forecasted. By following these guidelines, analysts can leverage the power of Prophet for accurate and reliable time series forecasting.

4.2.1 Model Architecture and Configuration

**How prophet works:**

Prophet is an additive model that effectively captures the various components of a time series **y(t) = g(t) + s(t) + h(t) + ϵ** (Taylor & Letham, 2018):

Trend **g(t)**: The trend component models the long-term behaviour of the time series. It captures the overall direction in which the data is changing over time, helping us understand whether the series is increasing, decreasing, or remaining stable.

Seasonality **s(t**): The seasonality component accounts for recurring patterns or cycles that repeat over fixed intervals, such as daily, weekly, or yearly effects. Prophet uses Fourier series to model seasonality, which enables it to handle complex seasonal patterns.

Holidays and Occasions **h(t)**: This component considers the impact of special events or occasions on the time series. For example, it can account for the increased demand during product launches, holiday seasons like Diwali or Christmas, or any other significant events that may affect the data.

Irreducible Error **ϵ**: The irreducible error term represents the noise or random fluctuations in the data that cannot be explained by the model's components. It accounts for the uncertainty and unpredictability in the time series.

By decomposing the time series into these additive components, Prophet can effectively model and forecast complex time series data. The trend captures the overall behaviour, seasonality captures recurring patterns, and the occasion component accounts for specific events. The irreducible error term acknowledges that there will always be some level of uncertainty and randomness in the data, which the model cannot fully explain. This approach allows data analysts to gain a comprehensive understanding of the time series and make accurate forecasts.

4.2.2 Training Parameter

The Prophet model consists of five essential training parameters that data analysts need to consider (Vishwakarma, A., Singh, A., Mahadik, A., & Pradhan, R. (2020)):

a) Base Trend **k**: The base trend parameter represents the overall trend component of the time series. It captures the fundamental direction in which the data is changing over time.

b) Offset Parameter (**m**): The offset parameter represents the shift or offset of the overall trend. It accounts for any displacement or deviation from the base trend, allowing the model to adjust the trend line accordingly.

c) Changepoints **δ**={δi}: Changepoints are time points in the data where the trend experiences abrupt changes or shifts. These points allow the model to identify and adapt to different periods of the time series with varying trends.

d) Seasonal and Exogenous Parameters **β**={βi} **Zi**=0: The seasonal component is modelled using Fourier series, capturing the periodic patterns in the data, such as daily, weekly, or yearly fluctuations. Additionally, the exogenous regressors, represented by the parameters **βi**, account for the impact of extra features on the time series.

e) Level of Noise **σ**: The noise parameter σ represents the level of random fluctuations or uncertainty in the data. It allows the model to account for the inherent unpredictability in the time series.

Furthermore, for each i-th regressor, the parameters mutrain and stdtrain represent the mean and standard deviation values, respectively, calculated from the training data. These values are essential for effectively incorporating additional regressors into the model.

Understanding and tuning these parameters are crucial for training an accurate and reliable Prophet model. By appropriately setting these parameters, analysts can ensure the model captures the underlying patterns and behaviours of the time series, leading to better forecasts and insights.

**4.2.3 Modeling and Optimization (Hyperparameter Tuning)**

4.2.3.1 Bayesian optimization (Hyperparameter tuning)

Let's perform Bayesian optimization Bayesian optimization to find the optimal hyperparameters for a Prophet forecasting model using Facebook's Prophet library.

Bayesian Optimization in time series involves using a probabilistic model, often a Gaussian Process, to automate the optimization of hyperparameters in predictive models for improved forecasting.

This method efficiently explores the hyperparameter space by iteratively evaluating models, updating the probabilistic model, and selecting the next set of hyperparameters to test. Bayesian Optimization can handle uncertainty, noisy data, and non-convex objective functions frequently encountered in time series analysis. It facilitates the development of more accurate predictive models while reducing the need for manual configuration, making it a valuable tool for enhancing forecasting accuracy in time series applications.

The goal is to fit a time series forecasting model to the provided dataset ('ISEQ20.xlsx') and optimize the hyperparameters to minimize the chosen loss functions MAE, MSE, RMSE, MAPE and R-squared (R2)

A screenshot of a graph

Description automatically generated

***Table 04 - Bayesian optimization from Iseq20***

From Fig. 10, we have:

* **Updated Best Hyperparameters:** The Bayesian optimization process has identified new optimal hyperparameters for the Prophet model. These hyperparameters are critical for determining the model's performance, and they have been revised to changepoint\_prior\_scale = 0.1406 and holidays\_prior\_scale = 0.5588.
* **Model Fit:** The Prophet model has been retrained using the newly optimized hyperparameters. It's important to emphasize that the model's performance is significantly influenced by the quality and characteristics of the underlying data.
* **Performance Metrics:**
* MAE (Mean Absolute Error): The MAE is approximately 227.86, indicating that, on average, the model's predictions deviate from the actual values by about 227.86 points. Lower MAE values are desirable, but it's essential to consider the scale of the data.
* MSE (Mean Squared Error): The MSE stands at 62,881.45, reflecting the average squared difference between the model's predictions and the actual values.
* RMSE (Root Mean Squared Error): With an RMSE of 250.76, this metric represents the square root of the MSE. It provides insight into the average magnitude of errors, measured in the same units as the target variable. Lower RMSE values are indicative of improved model performance.
* MAPE (Mean Absolute Percentage Error): The MAPE has been calculated at 17.38%, indicating that, on average, the model's predictions exhibit an absolute percentage error of 17.38%. Lower MAPE values correspond to higher prediction accuracy.
* R-squared (R2): The R-squared value is negative (-2.65), meaning that our model does not fit the data well. this measure indicates that the proportion of variance in the dependent variable that can be explained by the independent variables. A negative value suggests that the model's predictions are less accurate than simply using the mean of the target variable.

Overall Assessment:

While the Bayesian optimization process has refined the hyperparameters for the Prophet model, the performance metrics still suggest suboptimal model performance. The negative R-squared value indicates that the model may struggle to capture the underlying patterns within the dataset. To enhance forecasting accuracy, it remains crucial to investigate other factors such as data quality, feature engineering, and potentially explore alternative modeling approaches. Further analysis is warranted to gain insights into the reasons behind the model's limited fit and to identify opportunities for improvement.

4.2.3.1 Prophet Model Forecast with Confidence Intervals on optimized hyperparameters.

A graph showing a line of stock

Description automatically generated with medium confidence

***Fig 07 - Prophet Model Forecast with Confidence Intervals on optimized hyperparameters from Iseq20.***

As shown on Fig. 07:

* **Forecasted Values**: range from 24 August 2020 to 17 February 2024. Starting at 1141.62 and trending upwards, they reach 1784.60 on 17 February 2024, indicating a positive trend.
* **Upper Confidence Interval** **(CI)**: the values are represented in the 'yhat\_upper' column, that provide an upper limit for the forecasted stock prices. These values start at 1190.16 and show a potential upper boundary for stock prices, indicating a maximum of around 1888.16 on 17 February 2024.
* **Lower Confidence Interval (CI)**: the values are shown in the 'yhat\_lower' column, and provide a lower limit for forecasted stock prices. Starting at 1095.81, these values indicate the lowest potential stock price levels, suggesting prices may not drop below 1679.34 on 17 February 2024.

**Overall Conclusion:**

In summary, the forecasts, along with upper and lower confidence intervals, provide valuable insights into expected trends and the potential range of stock prices. The forecasts indicate an upward trajectory in stock prices, while the confidence intervals account for uncertainty and potential price fluctuations. Investors should consider these forecasts and confidence intervals when making investment decisions, factoring in both the upward trend and potential variations in stock prices.

**4.3 Decomposition and Smoothing**

4.3.1 Trend analysis

A graph showing a line

Description automatically generated with medium confidence

***Fig 08 - Trend Analyse from Iseq20.***

Trend Component (*Fig. 08*) reveals the underlying, longer-term pattern in the data. Starting from 27 August 2020 at 1142.93 to 19 August 2023 at 1459.53, there is a noticeable upward trend in the closing prices of the ISEQ20 index.

4.3.2 Seasonal decomposition

A blue and white barcode

Description automatically generated

***Fig 09 - Seasonal decomposition Analyse from Iseq20.***

The seasonal component (*Fig. 09*) represents regular, repeating patterns in the data that occur at specific intervals. It exhibits a distinct pattern with fluctuations alternating between positive and negative values. This pattern implies that there are recurrent seasonal influences affecting the closing prices, with deviations of approximately ±0.93 to ±0.23 points from the trend.

4.3.3 Residual component

A graph showing a sound wave

Description automatically generated

***Fig 10 - Residual component Analyse from Iseq20.***

The residual component (*Fig. 10)* reflects the unexplained variability or noise in the data after accounting for the trend and seasonal patterns. On August 27, 2020, there is a notable positive residual value of approximately 5.75 points, indicating an unexpected increase in the closing prices beyond what the trend and seasonal patterns explain. On 08 March 2023, there is the highest negative residual value of approximately -50.44 points, suggesting a high decrease in closing prices beyond expected patterns.

**4.4 PERFORMANCE EVALUATION**

**4.4.1 Performance Evaluation Metrics**

***Mean Absolute Error (MAE)***

Mean Absolute Error (MAE) measures the average absolute difference between predicted and actual values in a dataset. It provides a straightforward way to quantify the accuracy of a predictive model and is recommended for its simplicity and ease of interpretation.

The MAE formula sums up the absolute differences between the predicted (Ŷ) and actual (Y) values for all data points, divides by the number of data points (n), and yields the average absolute error.

A mathematical equation with numbers and symbols

Description automatically generated

***Mean Squared Error (MSE)***

Mean Squared Error (MSE) is another metric for assessing regression model performance. Unlike MAE, it measures the average of the squared differences between predicted and actual values [4]. By squaring the errors, MSE penalizes larger errors more heavily than smaller ones.

The MSE formula computes the squared differences, averages them over all data points, and produces a single value. This makes it more sensitive to outliers compared to MAE.

A mathematical equation with numbers and symbols

Description automatically generated

***Root Mean Squared Error (RMSE)***

Root Mean Squared Error (RMSE) is a modification of MSE that provides error values in the same units as the target variable. By taking the square root of the MSE, RMSE returns a measure that is directly interpretable.

RMSE is often preferred when the scale of the dependent variable matters. For example, in the context of predicting house prices, RMSE would produce errors in dollars, which are easier to understand than squared errors.

A black square with a square and a square with a square and a square with a square and a square with a square and a square with a square and a square with a square and a square

Description automatically generated

***Mean Absolute Percentage Error (MAPE)***

Mean Absolute Percentage Error (MAPE) is frequently used in forecasting to gauge how accurate predictions are relative to the actual values, expressed as a percentage. This metric is particularly valuable when we need to understand the proportional errors compared to the real values. The MAPE formula calculates the absolute percentage difference between each predicted and actual value, averages these differences, and reports the result as a percentage.

A mathematical equation with numbers and symbols

Description automatically generated

***R-squared (R²)***

R-squared (R²) serves as a statistical measure that helps assess how well a regression model explains the variability observed in the dependent variable (often denoted as Y). Ranging from 0 to 1, R² is a numeric value where higher figures signify a stronger fit of the model to the actual data.

The R² calculation involves a comparison between two variances. First, it evaluates the variance that the model under consideration can account for, known as SSR (sum of squared residuals). Second, it considers the total variance present in the dependent variable, known as SST (total sum of squares). R² is effectively a proportion, and it quantifies the fraction of the total variability in the dependent variable that can be attributed to the model.

A black and white math symbols

Description automatically generated with medium confidence

**4.4.2** **Performance Evaluation Metrics Comparison (ISEQ20)**

A white and green background with black text

Description automatically generated

***Table 05 - Performance Evaluation Metrics from Iseq20.***

As revealed by Table 05, we have:

* **Mean Absolute Error (MAE)**: The MAE value of approximately 224.28 indicates that, on average, the model's predictions differ from the actual values by approximately 224.28 units. This metric represents the absolute magnitude of prediction errors.
* **Mean Squared Error (MSE)**: The MSE value of approximately 62104.87 represents the average squared difference between the model's predictions and the actual values. A higher MSE suggests a significant spread in prediction errors, potentially due to outliers or model inaccuracies.
* **Root Mean Squared Error (RMSE)**: The RMSE value of approximately 249.21 is the square root of the MSE and provides a sense of the average magnitude of prediction errors. It is similar to the MAE but gives more weight to larger errors. In this case, the RMSE is smaller than the MAE, indicating that larger errors have less impact on the overall RMSE.
* **Mean Absolute Percentage Error (MAPE):** The MAPE value is expressed as a percentage and is useful for understanding the relative magnitude of errors. it's 17.11% indicates that, on average, the model's predictions deviate from the actual values by about 17.11%.
* **R-squared (R2):** The R-squared (R2) value of approximately -2.60 is unexpectedly negative. R-squared measures the proportion of the variance in the dependent variable that is predictable from the independent variables. A negative R2 suggests that the model is performing worse than a horizontal line (a model that predicts the mean of the dependent variable). This may indicate that the model is not capturing the underlying patterns in the data effectively.

**Conclusion:**

The model's performance, as indicated by the provided metrics, appears to be suboptimal. The MAE, MSE, and RMSE values, although providing insight into the magnitude of errors, suggest that the model's predictions exhibit significant discrepancies from the actual values. The MAPE of approximately 17.11% indicates that, on average, the model's percentage errors are moderate. The negative R-squared value is a concerning sign, indicating that the model is not explaining the variance in the data and is potentially a poor fit.

**4.5 Model Comparison**

**4.5.1 Models compared to Prophet Model**

***ARIMA (AutoRegressive Integrated Moving Average)***

**Explanation:** ARIMA stands for AutoRegressive Integrated Moving Average. It is a widely used time series forecasting model that combines autoregressive (AR) and moving average (MA) components with differencing to make a time series stationary. ARIMA models are excellent for modeling univariate time series data.

**Technical Insights:**

* The ARIMA model is characterized by three main components: p, d, and q, denoting the order of autoregressive, differencing, and moving average components, respectively.
* The AR component (p) represents the relationship between the current value and its past values.
* The I component (d) represents the number of differences needed to make the time series stationary.
* The MA component (q) represents the relationship between the current value and past forecast errors.
* The ARIMA model can be represented as ARIMA (p, d, q).

**Formula:** The general formula for an ARIMA model is:

***Yt*​=*c*+*ϕ*1​*Yt*−1​+*ϕ*2​*Yt*−2​+…+*ϕp*​*Yt*−*p*​−*θ*1​*et*−1​−*θ*2​*et*−2​−…−*θq*​*et*−*q*​+*et*​**

Where:

* ***Yt*​** is the observed value at time t.
* ***c*** is a constant.
* ***ϕ****i*​ are the autoregressive coefficients.
* ***θi*​** are the moving average coefficients.
* ***et*​** is the error term at time t.

***SARIMA (Seasonal ARIMA)***

**Explanation:** SARIMA, or Seasonal AutoRegressive Integrated Moving Average, is an extension of the ARIMA model that accounts for seasonality in time series data. It's suitable for data with recurring patterns at fixed intervals.

**Technical Insights:**

* SARIMA includes all the components of ARIMA (p, d, q) but adds seasonal components (P, D, Q, s) to capture seasonal patterns.
* The seasonal AR component (P) captures the autoregressive relationship in the seasonal data.
* The seasonal differencing (D) represents the number of seasonal differences required to make the data stationary.
* The seasonal MA component (Q) captures the moving average relationship in the seasonal data.
* 's' represents the seasonal period, such as 12 for monthly data with yearly seasonality.

**Formula:** The SARIMA model can be represented as SARIMA(p, d, q)(P, D, Q)s.

***LSTM (Long Short-Term Memory)***

**Explanation:** LSTM, or Long Short-Term Memory, is a type of recurrent neural network (RNN) designed for sequential data like time series. LSTMs are particularly effective at capturing long-term dependencies and are widely used for time series forecasting.

**Technical Insights:**

* LSTMs consist of LSTM cells, which contain gates (input, forget, output) to control information flow.
* The input gate controls the flow of new information into the cell.
* The forget gate controls the removal of information from the cell.
* The output gate controls the information that is passed to the output.
* LSTMs can capture patterns and dependencies across various time steps.

**Formula:** The LSTM equations are complex and involve multiple steps. Key equations include those for the cell state (***Ct***​), hidden state (***ht***​), and the gates.

***Neuralprophet***

**Explanation:** Neuralprophet is a forecasting model developed by Facebook that combines elements of neural networks and classical time series forecasting techniques. It's designed to handle irregularly spaced time series data and automatically handle seasonality and holidays.

**Technical Insights:**

* Neuralprophet uses a neural network architecture that includes feedforward layers, seasonal components, and additional features like holidays.
* It automatically detects and incorporates seasonal patterns without the need for manual specification.
* Neuralprophet can handle missing data and outliers gracefully.

**Formula:** The inner workings of Neuralprophet are based on neural network architectures, which involve numerous mathematical operations and layers. The specifics of these operations are not typically exposed to the user.

***GARCH***

**Explanation:** GARCH, or Generalized Autoregressive Conditional Heteroskedasticity, is a statistical model used to capture volatility clustering in financial time series data. It's commonly employed in modeling and forecasting stock market returns.

**Technical Insights:**

* GARCH models assume that the conditional variance of the time series is a function of past values, squared returns, and past conditional variances.
* It is well-suited for capturing time-varying volatility, where periods of high volatility are followed by periods of low volatility.
* GARCH models include parameters for autoregressive components (p) and moving average components (q) for the conditional variance.

**Formula:** The GARCH model is characterized by equations for conditional variance, typically including the conditional mean, squared returns, and conditional variances. The specific equations depend on the GARCH variant (e.g., GARCH (1,1)).

4.5.2 Prophet Model vs. Other Algorithms

A table with numbers and symbols

Description automatically generated

***Fig 11 - Prophet Model compared to other models (Performance)***

4.5.3 Scatter plot comparing Prophet with other algorithms.

A screenshot of a computer

Description automatically generated

***Fig 12 - Prophet Model compared to other models (Performance) using scatter plot.***

Fig. 11 and 12 reveal:

* **Mean Absolute Error (MAE):** Range from 139.80 (ARIMA) to 799.84 (NeuralProphet). Prophet's MAE of 641.50 is approximately 78% higher than ARIMA, indicating that ARIMA has the lowest absolute prediction errors. However, Prophet outperforms NeuralProphet by approximately 24.7%.
* **Mean Squared Error (MSE)**: Range from 33,089.28 (ARIMA) to 744,732.70 (NeuralProphet). Prophet's MSE of 480,185.90 is approximately 55.1% lower than NeuralProphet, making it the superior model in terms of minimizing squared prediction errors. However, ARIMA still outperforms Prophet by approximately 93.1%.
* **Root Mean Squared Error (RMSE)**: Range from 181.90 (ARIMA) to 1334.10 (GARCH). Prophet's RMSE of 692.95 is 92.5% lower than GARCH, which indicates that it provides better point forecasts in terms of the root mean squared error. However, ARIMA and LSTM have lower RMSE values than Prophet, outperforming it by approximately 73.7% and 71.1%, respectively.
* **Mean Absolute Percentage Error (MAPE)**: Range from 0.1144 (ARIMA) to 0.9920 (GARCH). Prophet's MAPE of 0.4911 is approximately 76.7% higher than ARIMA, demonstrating that ARIMA has the lowest percentage prediction errors. However, Prophet significantly outperforms GARCH, which has the highest MAPE, by 0.9920%.
* **R-squared (R2) Score:** Range: -0.9191 (ARIMA) to -102.22 (GARCH). Prophet's R2 score is negative (-42.19), indicating that it doesn't fit the data well compared to a horizontal line. In this case, ARIMA has the highest R2 score, being closest to 0. This suggests that ARIMA fits the data better, explaining more of the variability.

In summary, when comparing Prophet to the other models: Prophet performs better than NeuralProphet and GARCH across all metrics, with significant percentage improvements in MAE, MSE, RMSE, and MAPE. However, Prophet is outperformed by ARIMA and LSTM in terms of MAE, MSE, RMSE, and R2 score, indicating that these models provide more accurate point forecasts and better data fit.

The choice of the best model depends on the specific requirements of we are forecasting a task. If minimizing absolute or percentage errors is crucial, ARIMA may be preferred. If we prioritize ease of use and interpretability, Prophet might be a suitable choice despite its slightly higher errors compared to ARIMA and LSTM.

**4.6** **Index Comparison**

Now, let’s compare several indexes from Finance yahoo (3 years 24/08/2020 to 22/08/2023) BEL20 (Belgium), FTSE100 (UK), CAC40 (France), ISEQ20 (Ireland), DAX40 Germany), PSI20 (Portugal), ISEQ20 in order to preprocess and have a comparable data and use date and close columns as it's a multivariate analyse for time series and anomaly detection.

**A graph of a graph

Description automatically generated with medium confidence**

***Fig 13 - Index Comparison Iseq20 and other European indexes***

Key Observations (*Fig.13*):

* **Initial Variability (August 2020)**: In August 2020, the indices exhibit variability in their Close prices. Indices such as BEL20 and FTSE100 show relative stability, while others like DAX40 and PSI20 experience fluctuations.
* **Market Response to Events**: Throughout the dataset, significant market events and macroeconomic factors likely influenced the indices. Notable price movements may correspond to economic announcements, geopolitical events, or sector-specific news.
* **Pandemic Impact (Early 2020):** A visible impact of the COVID-19 pandemic can be seen in early 2020, with a substantial drop in indices' Close prices followed by gradual recoveries.
* **Recovery and Volatility**: Indices generally recover from the initial pandemic shock but continue to display volatility. Notably, DAX40 and PSI20 show relatively larger fluctuations compared to others.
* **Periods of Synchronization**: At times, several indices move in tandem, reflecting broader market trends. These synchronized movements might be influenced by global economic indicators or sector-wide developments.
* **Differences in Behaviour**: Despite overall similarities, individual indices exhibit unique behaviours. For instance, FTSE100 and CAC40 show relatively smoother trends, while BEL20 and PSI20 experience more pronounced oscillations.

4.7.1 Basic Statistics European indexes

A table of numbers and numbers

Description automatically generated

***Table 07 - Basic Statistics of European indexes***

A screenshot of a chat

Description automatically generated

***Fig 14 - Fig. 20 Basic Statistics European indexes (graphical)***

Analysing the provided statistics for the six stock market indices (BEL20, FTSE100, CAC40, ISEQ20, DAX40, and PSI20) in their respective 'Close' columns, we can draw the following conclusions (Table 07 and Fig. 14):

* **Mean (Average) Value**: These indices exhibit significant differences in their mean (average) values. The DAX40 has the highest mean value at roughly 14,590, indicating a generally higher average stock price level. In contrast, the ISEQ20 has the lowest mean value at around 1,337.
* **Standard Deviation (Volatility)**: The standard deviation measures the extent of price fluctuations or volatility. A higher standard deviation suggests greater price swings. The DAX40 boasts the highest standard deviation (1173.04), indicating more significant price volatility, while BEL20 has the lowest standard deviation (287.84), implying relatively lower volatility.
* **Minimum and Maximum Values**: The minimum and maximum values represent the lowest and highest closing prices observed during the data period. DAX40 records the highest maximum value (16,469.75), signifying the peak during the period, while BEL20 shows the lowest maximum value (4,402.32). CAC40 reports the lowest minimum value (4,569.67), indicating the least price drop, while PSI20 reflects the highest minimum value (3,863.20).
* **Median (50th Percentile)**: The median, or 50th percentile, reflects the middle value when the data is sorted in ascending order. It provides insight into the central tendency. The medians vary, with DAX40 having the highest median (14,856.48) and ISEQ20 having the lowest median (1,368.86).
* **Percentiles (25th and 75th)**: The 25th and 75th percentiles help understand the data's spread and identify the interquartile range. DAX40 reports the highest 75th percentile value (15,623.23), indicating a relatively higher upper price range, while CAC40 has the lowest 75th percentile value (7,017.20). PSI20 shows the highest 25th percentile value (5,111.13), suggesting a relatively higher lower price range, while FTSE100 displays the lowest 25th percentile value (6,963.49).

4.7.2 Missing values

A green rectangle with white text

Description automatically generated

***Table 08 – Missing values on European indexes***

10 missing values (in 764) in all the datasets, but not relevant to our analyse.

4.7.3 Outliers

A close-up of numbers and symbols

Description automatically generated

***Table 09 – Outliers on European Indexes***

there are three Potential Outliers in the Close columns 'FTSE100.xlsx, but they seem legitimate, because they are usual and true values and following same position as the rest, so we won't remove them from my dataset.

4.7.4 Heatmap for performance comparison

A screenshot of a graph

Description automatically generated

***Fig 15 - Heatmap for performance comparison (European indexes)***

In our case, the correlation coefficients are all positive, which indicates that there is a positive linear relationship between the close prices of the 6 indexes (fig. 15).

We can see here that the Irish index has the strongest correlation in the map and with the German index (0.94), followed by UK-Portuguese indexes (0.93), and UK-French indexes correlation (0.91). this means that the Irish market is strongly influenced by the German Market.

4.7.5 Irish Vs German Index

4.7.5.1 Scatter Plot

A graph showing a number of dots

Description automatically generated with medium confidence

***Fig 16 - Differences in index Prices comparison using scatter plot (Irish Vs German Index)***

The indexes prices of DAX40 vs ISEX20 differences are explained in this plot (Fig. 17). The colours reflect the difference between the two indexes, going from the lowest difference of 10477.90 (in purple and left down the plot) to the maximum of 14945.68 (in yellow on the top right of the plot). We can notice the correlation between both since the plot follow a constant tendency to growth.

4.7.5.2 Stacked Area Chart

A graph showing a green line

Description automatically generated

***Fig 17 - Differences in index Prices comparison using Stacked Area Chart (Irish Vs German Index)***

Conclusion from both Fig. 17:

* **Index Values:** The DAX 40 index represents the German stock market. It has consistently maintained higher closing values compared to the ISEQ 20 index. ISEQ 20: The ISEQ 20 index represents the Irish stock market. While it generally has lower closing values compared to the DAX 40, it follows a similar overall trend.
* **Market Trends:** Overall Increase: Both indices show a general upward trend in their closing values over the observed period. This indicates overall positive performance in both the German and Irish stock markets during this time.
* **Volatility and Stability:** The DAX 40 appears to exhibit higher volatility compared to the ISEQ 20, as seen in its larger price fluctuations. This suggests that the German market might be subject to more significant price swings than the Irish market.
* **Recent Performance:** As of August 22, 2023, the DAX 40 is at 15,705.62, while the ISEQ 20 is at 1,461.70. This reflects the pattern observed throughout the data, with the DAX 40 maintaining a significantly higher value compared to the ISEQ 20.
* **Risk and Return:** The higher volatility in the DAX 40 may provide potentially higher returns but also comes with increased risk. The ISEQ 20's relatively lower volatility might offer stability but could yield lower returns.

In summary, the DAX 40 and ISEQ 20 indices exhibit differing performance and volatility characteristics, with the DAX 40 generally outperforming the ISEQ 20 in terms of closing values. However their trends is similar as the Irish market follows German growth and decrease tendencies.

4.7.5.3 Cumulative returns comparison

Cumulative returns comparison is a fundamental concept in the field of finance, enabling investors and analysts to evaluate the performance of various financial assets or investment strategies over a specified period. It involves calculating and comparing the total returns generated by these assets or strategies over time, facilitating data-driven investment decisions.

The calculation of cumulative returns is straightforward. It involves summing the percentage returns or price changes for each period within the chosen time frame. The formula for calculating cumulative returns is as follows:

CumulativeReturn=P0​Pt​−P0​​×100%

Where:

* Cumulative Return represents the total return over the specified period.
* ***Pt***​ is the final value of the investment or portfolio at the end of the chosen time frame.
* ***P*0**​ is the initial value of the investment or portfolio at the beginning of the period.

A graph showing a green and blue line

Description automatically generated

A close-up of a text

Description automatically generated

***Fig. 18 - Differences in index Prices comparison using Stacked Area Chart (Irish Vs German Index)***

*Conclusion from Fig. 18:*

* **Performance Comparison:** DAX 40 has underperformed ISEQ 20 in terms of cumulative return. DAX 40 experienced a significant decline of approximately -16.80% from the starting value, while ISEQ 20 had a relatively milder decline of approximately -4.07%.
* **Risk Assessment:** DAX 40 appears to be riskier than ISEQ 20 during the observed period, as it has shown a larger decline in cumulative return.
* **Market Conditions:** In conclusion, DAX 40 has shown a more significant decline in cumulative return compared to ISEQ 20, indicating higher risk and underperformance during the observed period.
* **Indexes correlation:** both indexes have similar evolution confirming the high correlation of 94% from the heatmap.

**5. ANOMALY DETECTION**

**5.1 Introduction**

Anomaly detection in time series data is a statistical process aimed at identifying data points or patterns that exhibit substantial deviations from the expected behaviour within a sequential dataset. This procedure is fundamental in various applications, such as finance, cybersecurity, and industrial equipment monitoring, where timely detection of anomalies is critical for informed decision-making and risk mitigation.

In technical terms, time series data is typically represented as a sequence of observations or data points over discrete time intervals, mathematically expressed as:

(*t*1​,*x*1​),(*t*2​,*x*2​),…,(*tn*​,*xn*​)}

Where:

* *ti*​ represents the time index at which the observation *xi*​ was recorded.
* *xi*​ represents the value of the observed data at time *ti*​.

The primary objective of anomaly detection in time series data is to identify a subset of data points *A*, where *A* consists of anomalous data points:

*A*={(*ti*​,*xi*​)∣(*ti*​,*xi*​) is anomalous}

Anomalies, in this context, pertain to data points that exhibit notable deviations from what would be regarded as the usual or anticipated behaviour within the time series. The identification of these anomalies typically entails the utilization of diverse statistical, machine learning, or deep learning methods, each grounded in its mathematical foundations.

**5.2. Types of Anomalies**

In the realm of anomaly detection in time series data, anomalies can be categorized into distinct types, each with its mathematical characteristics:

* **Point Anomalies**: Point anomalies are individual data points within the time series that deviate significantly from the expected distribution of data. Mathematically, a data point *xi*​ can be considered a point anomaly if it falls outside a defined range, often based on statistical measures such as the mean (*μ*) and standard deviation (*σ*). A common criterion for detecting point anomalies is:

*xi*​ ∈/[*μ*−*kσ*,*μ*+*kσ*]

Where:

* *xi*​ is the data point in question.
* *μ* is the mean of the time series data.
* *σ* is the standard deviation of the time series data.
* *k* is a user-defined threshold multiplier.

Point anomalies are essentially isolated extreme values within the time series.

* **Contextual Anomalies**: Contextual anomalies are data points that exhibit anomalous behaviour within a specific context or condition. To detect contextual anomalies, mathematical models are often employed to establish the context and assess the likelihood of observing a given data point within that context. This is typically expressed as a conditional probability:

*P*(*xi*​∣*xi*−1​,*xi*−2​,…,*xi*−*k*​)

Where:

* *P*(*xi*​∣*xi*−1​,*xi*−2​,…,*xi*−*k*​) represents the conditional probability of observing *xi*​ given the past *k* observations.

Contextual anomalies are identified when the observed data point *xi*​ has a conditional probability significantly lower than expected.

* **Collective Anomalies**: Collective anomalies, also known as group anomalies, involve identifying groups or patterns of data points that collectively exhibit anomalous behaviour. These anomalies cannot be detected by analysing individual data points in isolation but rather by considering their collective behaviour. Methods for detecting collective anomalies often rely on clustering or density estimation techniques, where mathematical formulations aim to identify groups of data points that do not conform to the expected cluster or density distributions.

In summary, understanding the technical intricacies of point, contextual, and collective anomalies is essential for data analysts when designing and implementing anomaly detection algorithms tailored to specific use cases within time series data. Each type of anomaly requires distinct mathematical models and approaches for effective detection.

**5.3 Anomaly Detection Plots**

**5.3.1 Timeline with anomaly detection**

**A graph showing the growth of the stock market

Description automatically generated**

***Fig 19 - Timeline with anomaly detection from Iseq20***

From Fig. 19:

1. Number of Anomalies: There are a total of 87 anomalies in the dataset.
2. Anomalies by Year:
   * 2023: 5 anomalies (5.75% of total anomalies)
   * 2022: 23 anomalies (26.44%)
   * 2021: 24 anomalies (27.59%)
   * 2020: 35 anomalies (40.23%)

In total, there are 87 instances of anomalies during this period, occurring irregularly. On average, there is an anomaly in roughly 2.6% of the observed months. These irregular occurrences indicate that anomalies are not isolated incidents but can be influenced by various factors.

1. Largest Anomaly: The largest anomaly occurred on July 10, 2023, with a closing price of 1460.22.
2. Smallest Anomaly: The smallest anomaly occurred on October 28, 2020, with a closing price of 1082.61.
3. Average Anomaly: The average closing price for anomalies is approximately 1315.48.
4. Median Anomaly: The median closing price for anomalies is approximately 1309.57.
5. Deviation:
   * Maximum Deviation: The maximum deviation from the rolling 30-day mean for an anomaly is approximately 378.72 points (on November 28, 2022).
   * Minimum Deviation: The minimum deviation from the rolling 30-day mean for an anomaly is approximately 0.62 points (on March 24, 2023).
   * Average Deviation: The average deviation from the rolling 30-day mean for anomalies is approximately 39.94 points.
   * Percentage Deviation: On average, anomalies represent a deviation of about 3.04% from the rolling 30-day mean closing price.

Magnitude of Deviations: The anomalies are characterized by substantial price deviations from the rolling 30-day mean, with an average deviation of approximately 13.45%. These deviations significantly surpass the threshold of 2 standard deviations from the mean, emphasizing their significance.

In summary, the 87 identified anomalies in closing prices, spanning a three-year period, signify significant events and deviations within the financial markets. These anomalies, although unevenly distributed, exhibit substantial magnitudes that require thorough scrutiny. They offer valuable insights into the fluctuations of the asset's price, serving as crucial data points for shaping investment strategies and informed risk management.

**5.4 Implementing Anomaly Detection Models**

***Isolation Forest***

The Isolation Forest is an effective anomaly detection algorithm designed to identify anomalies within datasets. Developed by Liu et al. in 2008, it works on the principle that anomalies are easier to isolate because they are fewer in number and distant from normal data points. The key components of the Isolation Forest are as follows:

* + **Random Partitioning:** The algorithm randomly selects a feature and splits the data based on a random value within the feature's range. This process is repeated recursively until anomalies are isolated or a predetermined depth is reached.
  + **Anomaly Scoring:** Anomalies are identified based on their shorter average path lengths within the isolation trees. Normal data points tend to have longer paths.

The anomaly score (***s*(*x***,***n***)) for a data point ***x*** in a dataset of size ***n*** can be calculated using the formula:

***A black and white math equation

Description automatically generated***

Where:

* + *E*(*h*(***x***)) is the expected path length for ***x***.
  + *c*(***n***) is a constant related to the average path length.

Isolation Forest offers advantages such as scalability, suitability for high-dimensional data, and straightforward implementation.

***One-Class SVM (Support Vector Machine)***

The One-Class SVM is a machine learning-based anomaly detection method introduced by Schölkopf et al. in 2001. It is particularly useful when dealing with datasets where anomalies are rare and hard to define. One-Class SVM separates normal data from anomalies by creating a hyperplane that maximizes the margin around the normal data points. Key aspects of the One-Class SVM include:

* + **Objective Function:** The goal is to find a hyperplane represented by the weight vector (***w***) and offset term (***ρ***) that maximizes the margin while minimizing the impact of anomalies. The optimization problem can be formulated as:

A math equations and symbols

Description automatically generated with medium confidence

Where ***ν*** controls the trade-off between maximizing the margin and allowing some data points to be treated as anomalies.

One-Class SVM is powerful for capturing complex boundary shapes in high-dimensional spaces.

***Prophet Model***

The Prophet model is a time series forecasting tool developed by Taylor and Letham in 2018. Although primarily designed for forecasting, it can also be employed for anomaly detection by identifying deviations between observed and predicted values. Key components of the Prophet model are:

* + **Decomposition:** Prophet decomposes time series data into three primary components: trend, seasonality, and holidays (special events). These components are additive, and the observed values are expressed as the sum of these components along with an error term.

*y*(*t*)=*g*(*t*)+*s*(*t*)+*h*(*t*)+*ϵt*​

* + **Bayesian Framework:** Prophet employs a Bayesian framework to model these components and estimate prediction intervals. Anomalies can be detected when observed values fall outside these intervals.

Prophet's simplicity and capability to handle missing data and outliers make it a versatile tool for time series anomaly detection.

***NeuralProphet***

NeuralProphet is an extension of the Prophet model introduced to enhance its forecasting capabilities with the addition of neural networks. Developed by O'Hara-Wild et al. in 2021, NeuralProphet offers improved performance in capturing complex temporal patterns. Key features of NeuralProphet include:

* + **Neural Network Architecture:** NeuralProphet incorporates neural networks, typically including feedforward and LSTM (Long Short-Term Memory) layers. These networks enable the model to learn from historical data and make more accurate predictions.
  + **Anomaly Detection:** Like the Prophet model, NeuralProphet can be used for anomaly detection by comparing observed values to prediction intervals generated using the neural network.

NeuralProphet provides more flexibility and accuracy in capturing complex time series behaviours compared to the original Prophet model.

***LSTM (Long Short-Term Memory)***

LSTM is a type of recurrent neural network (RNN) designed for sequential data analysis. It is often used for time series forecasting and can also be applied to anomaly detection. Key characteristics of LSTM for anomaly detection include:

* + **Memory Cells:** LSTM networks consist of memory cells with gating mechanisms that control the flow of information. This architecture allows them to capture both short-term and long-term dependencies in sequential data.
  + **Training and Detection:** Anomalies in time series data can be detected by training an LSTM network on historical data and identifying deviations between the predicted and observed values. LSTM networks can capture complex temporal patterns.

LSTM-based anomaly detection is particularly suitable for scenarios where time dependencies are crucial.

In conclusion, these anomaly detection techniques, including Isolation Forest, One-Class SVM, Prophet models (Prophet and NeuralProphet), and LSTM networks, offer various approaches to identifying anomalies within datasets, with each having its strengths and suitability for specific data types and contexts. The choice of technique depends on the problem at hand and the nature of the data being analysed.

**5.5 Performance Evaluation Metrics for Anomaly Detection**

**5.5.1 Precision, Recall, and F1-Score**

***Precision (P):***

Precision, also known as positive predictive value, is a fundamental metric in anomaly detection that measures the accuracy of the model in correctly identifying true anomalies among all the instances labeled as anomalies. It quantifies the fraction of true positive predictions (correctly detected anomalies) relative to all instances predicted as anomalies. Mathematically, it is defined as:

A black text on a white background

Description automatically generated

Where:

* True Positives represents the number of correctly detected anomalies.
* False Positives represents the number of normal instances incorrectly classified as anomalies.

A high precision score indicates that the model has a low rate of false positives and is good at distinguishing anomalies from normal data points.

***Recall (R)***

Recall, also known as sensitivity or true positive rate, measures the ability of the model to capture all true anomalies in the dataset. It quantifies the fraction of true anomalies that are correctly identified by the model. Mathematically, it is defined as:

A black text on a white background

Description automatically generated

Where:

* True Positives represents the number of correctly detected anomalies.
* False Negatives represents the number of true anomalies that were missed by the model.

A high recall score indicates that the model effectively captures most of the anomalies in the dataset.

***F1-Score***

The F1-Score is a harmonic mean of precision and recall. It provides a single metric that balances both precision and recall, making it a useful measure when there is a trade-off between false positives and false negatives. The F1-Score is defined as:

A black text on a white background

Description automatically generated

A high F1-Score indicates a good balance between precision and recall. It is particularly valuable when the cost of false positives and false negatives is not equal and needs to be considered.

***Area Under the ROC Curve (AUC-ROC)***

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is a popular metric for evaluating the performance of binary classification models, including those used in anomaly detection. The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate (FPR) at various thresholds. The AUC-ROC quantifies the model's ability to distinguish between anomalies and normal data across different threshold settings.

Mathematically, AUC-ROC calculates the area under the ROC curve, which ranges from 0 to 1. An AUC-ROC score of 0.5 indicates random performance (no discrimination), while a score of 1 suggests perfect discrimination.

AUC-ROC is beneficial when we want to assess the model's overall ability to rank anomalies higher than normal instances across a range of possible threshold values. It does not assume an equal cost for false positives and false negatives.

***Precision-Recall Curve (AUC-PR)***

The Precision-Recall Curve (AUC-PR) is another performance metric for binary classification models, especially when dealing with imbalanced datasets, as is often the case in anomaly detection. This curve plots precision against recall at various threshold levels.

The AUC-PR quantifies the area under the Precision-Recall curve, which also ranges from 0 to 1. Unlike the AUC-ROC, the AUC-PR focuses on the trade-off between precision and recall, which is crucial in scenarios where false positives and false negatives have different implications.

A high AUC-PR score indicates a model that can achieve high precision while maintaining high recall, which is desirable in anomaly detection to minimize false alarms while capturing most anomalies.

**5.6 Results and Discussion**

5.6.1 Comparing Anomaly Detection Methods

A screenshot of a screen

Description automatically generated

***Table 09 – Anomaly detection models compared with performance metrics.***

A graph of different colored bars

Description automatically generated

***Fig 20 - Anomaly detection models compared with performance metrics.***

As shown on Table 09 and Fig. 20:

***Key Takeaways for Prophet***

* **Recall**: The Prophet model achieves a perfect recall score of 1.000, indicating its exceptional ability to identify all actual anomalies within the dataset. This suggests that Prophet excels at capturing true positive cases without missing any, making it highly reliable for anomaly detection tasks.
* **Precision**: The precision of the Prophet model is moderate at 0.562, indicating that while it effectively identifies anomalies, it may generate some false positives. This balance between precision and recall suggests that there is a trade-off between correctly identifying anomalies and potentially raising false alarms.
* **F1-Score**: The F1-Score, which balances precision and recall, is also moderate at 0.719. This metric highlights the model's capability to strike a reasonable balance between accurately identifying anomalies and minimizing false positive detections.
* **AUC-ROC**: The AUC-ROC score for Prophet is 0.500, which is relatively low. This indicates that the model's Receiver Operating Characteristic (ROC) curve performs no better than random chance when distinguishing between anomalies and normal cases. In terms of ROC performance, the Prophet model does not exhibit strong discriminatory power.
* **AUC-PR**: The AUC-PR is moderate at 0.781, considering the precision-recall trade-off. This value suggests that Prophet provides a reasonable balance between precision and recall in anomaly detection.

***Comparison to Other Models***

* **Isolation Forest**: Isolation Forest demonstrates a very high precision (1.000) and a slightly lower recall (0.977), indicating higher precision than Prophet but with a slightly lower recall rate. However, the lack of an AUC-ROC value makes it challenging to assess its ROC performance.
* **One-Class SVM**: One-Class SVM also achieves perfect precision (1.000) but a lower recall (0.562) compared to Prophet. It shares a similar precision-recall trade-off with Prophet, and like Isolation Forest, lacks an AUC-ROC value.
* **NeuralProphet**: NeuralProphet achieves perfect precision, recall, and F1-Score, indicating excellence in both precision and recall aspects. However, the absence of an AUC-ROC value makes it difficult to compare in terms of ROC performance.
* **LSTM:** LSTM demonstrates a moderate precision (0.555) and perfect recall (1.000), similar to Prophet. It has a slightly lower F1-Score (0.714) than Prophet. Like Prophet, it exhibits a low AUC-ROC (0.500) and a moderate AUC-PR (0.778).

**Summary:** The Prophet model excels in recall, ensuring that it accurately identifies all actual anomalies, making it suitable for tasks where missing any anomalies is critical. However, it maintains a moderate precision, signifying a trade-off between correctly identifying anomalies and generating some false positives. In terms of ROC performance, both Prophet and LSTM exhibit low AUC-ROC values, suggesting no significant advantage over random chance. The choice between Prophet and other models depends on the specific requirements of your anomaly detection task. If recall is of utmost importance, and some false positives can be tolerated, Prophet may be a suitable choice. Nevertheless, if higher precision or improved ROC performance is required, further exploration of alternative models or fine-tuning of Prophet's parameters may be necessary.

**6. CONCLUSION**

This thesis embarked on a comprehensive exploration of anomaly detection and time series forecasting within the Irish finance market, with a specific focus on evaluating the performance of the Prophet model. The research objectives were multifaceted, encompassing an in-depth evaluation of the Prophet model's forecasting capabilities, a critical analysis of its strengths and limitations, and an examination of how detected anomalies impact forecasting accuracy. Additionally, the proposed approach underwent rigorous validation using real-world financial data to ascertain its practicality and potential for application in financial decision-making. Additionally, prophet model performance was validated with real-world data.

The Data Distribution analysis of 'Close' prices revealed a concentration of data within specific price ranges, notably in the mid to higher price ranges, signifying the prevalent price levels within the Irish finance market. This understanding is critical for identifying trends and making informed decisions regarding trading and investment strategies. For instance, it's evident that the majority of 'Close' prices fall within the interval of (1448.704, 1496.887], emphasizing the market's tendency towards these price levels.

The density plot, utilizing Kernel Density Estimation, offered valuable probabilistic insights into the distribution of 'Close' prices. It highlighted the rarity of lower prices and the relatively higher likelihood of observing higher prices, providing a nuanced view essential for forecasting accuracy and risk assessment. For example, 'Close' prices in the lower range (e.g., 1063 to 1067) are rare, emphasizing their outlier status, while higher prices have a more substantial probability of occurrence.

Examining the timeline plot illuminated distinct periods of growth and decline in the market index, each with its set of driving factors. Understanding these trends is crucial for developing effective forecasting models that can adapt to changing market dynamics. Notably, the growth observed from October 2020 to September 2021, with the index peaking at 1,545.07, underscores the market's resilience post-pandemic.

The autocorrelation and partial autocorrelation plots uncovered significant correlations between past and present data points, particularly at lag 1, suggesting the influence of past values on current data. This provides a foundation for employing autoregressive (AR) models in time series forecasting. The gradual decline in ACF values and the presence of significant PACF spikes hint at the possibility of an autoregressive (AR) component in the data, emphasizing the importance of considering past values in forecasting.

Bayesian optimization played a pivotal role in refining the Prophet model's hyperparameters. The process led to the identification of optimal hyperparameters, notably changepoint\_prior\_scale = 0.1406 and holidays\_prior\_scale = 0.5588. While this optimization was a crucial step, it is important to recognize that the model's performance metrics still raise concerns. The MAE of approximately 227.86 and the negative R-squared value (-2.65) indicate that the model struggles to capture the underlying patterns in the Irish finance market data.

The Prophet model's forecast, coupled with upper and lower confidence intervals, provided valuable insights into the expected trends in the Irish finance market. The upward trajectory in stock prices, with forecasts ranging from 1141.62 on 24 August 2020 to 1784.60 on 17 February 2024, suggests a positive trend. However, the confidence intervals remind us of the inherent uncertainty, indicating potential fluctuations in stock prices. These forecasts and intervals should guide investment decisions, considering both the upward trend and potential variations.

The trend component analysis revealed a discernible upward trend in the closing prices of the ISEQ20 index. Starting from 27 August 2020 at 1142.93 and reaching 1459.53 on 19 August 2023, this pattern underscores the market's overall positive trajectory. The seasonal component exhibited a recurring pattern with fluctuations that suggest the influence of seasonal factors on closing prices. Understanding these regular influences is crucial for making informed financial decisions. The residual component analysis highlighted unexpected variations in closing prices. Notably, the positive residual value on August 27, 2020, and the highest negative residual on 08 March 2023, signify periods of unexplained variability in the market, suggesting the presence of anomalies.

Performance Evaluation Metrics (ISEQ20):The evaluation of the Prophet model's performance using various metrics offered a deep insight into its forecasting effectiveness. The Mean Absolute Error (MAE) of approximately 224.28 units represents the average magnitude of prediction errors. It quantifies the model's average deviation from actual values, shedding light on the model's precision. The Root Mean Squared Error (RMSE) of approximately 249.21 provides a sense of the average magnitude of prediction errors, giving more weight to larger errors and thus revealing the model's sensitivity to significant deviations. The Mean Absolute Percentage Error (MAPE) of approximately 17.11% offers a percentage-based understanding of the errors, aiding in the interpretation of relative inaccuracies. The negative R-squared (R2) value of approximately -2.60 is a concerning sign, indicating that the model struggles to explain the variance in the data and is potentially a poor fit. This necessitates a closer examination of the model's underlying assumptions and structure.

A thorough comparative analysis of the Prophet model against other algorithms revealed its relative performance across critical metrics. The Mean Absolute Error (MAE) comparison showcased Prophet's higher error rates compared to ARIMA, which indicated that ARIMA had the lowest absolute prediction errors among the models assessed. However, Prophet outperformed NeuralProphet by approximately 24.7%, highlighting its superiority over a specialized neural network-based forecasting approach. When considering Mean Squared Error (MSE), Prophet demonstrated a significant advantage over NeuralProphet by being approximately 55.1% lower. Despite this, ARIMA continued to outperform Prophet by approximately 93.1%, indicating that ARIMA is more effective in minimizing squared prediction errors. In terms of Root Mean Squared Error (RMSE), Prophet displayed a 92.5% lower value than GARCH, signifying its superior point forecast accuracy in terms of the root mean squared error. However, ARIMA and LSTM boasted lower RMSE values than Prophet, outperforming it by approximately 73.7% and 71.1%, respectively. The Mean Absolute Percentage Error (MAPE) comparison highlighted that Prophet had a 76.7% higher error rate compared to ARIMA, suggesting that ARIMA had the lowest percentage prediction errors among the models evaluated. Notably, Prophet significantly outperformed GARCH, which exhibited the highest MAPE. Lastly, the R-squared (R2) score comparison indicated that Prophet struggled to fit the data well compared to a horizontal line, underlining the model's limitations. In contrast, ARIMA displayed the highest R2 score, suggesting a better fit and explaining more of the variability in the data.

By constructing a heatmap for performance comparison, we observed strong positive correlations among the European indices, indicating a positive linear relationship in their close prices. Particularly noteworthy was the robust correlation between the ISEQ20 (Ireland) and DAX40 (Germany), with a correlation coefficient of 0.94, implying a substantial influence of the German market on the Irish market. A comparative analysis of both indexes highlighted distinct differences in their closing values and volatility. The DAX40 consistently maintained higher closing values, with a maximum value of approximately 16,469.75, and greater price fluctuations, suggesting higher risk and potential returns. In contrast, the ISEQ20 exhibited a relatively stable yet lower-performing trend, with a maximum value of approximately 4,402.32.

For Anomaly Detection,our scrutiny of a three-year dataset uncovered a total of 87 anomalies, presenting irregular occurrences throughout the timeline. These anomalies, often significant in magnitude, were vital indicators of deviations in asset prices from their rolling 30-day mean. The average deviation of approximately 13.45% from the mean underscored their importance as crucial data points for financial analysis and risk management. These anomalies offered crucial insights into market events and fluctuations, providing valuable inputs for shaping investment strategies.

Our meticulous examination of anomaly detection methods comparison, with a focus on the Prophet model, revealed critical insights into their performance. Recall: Prophet achieved a perfect recall score of 1.000, highlighting its ability to identify all actual anomalies without missing any, making it highly reliable for anomaly detection tasks. Precision: The precision of the Prophet model was moderate at 0.562, indicating a trade-off between correctly identifying anomalies and potentially raising false alarms. F1-Score: The F1-Score, balancing precision and recall, was moderate at 0.719, indicating a reasonable balance between accurately identifying anomalies and minimizing false positive detections. AUC-ROC: The AUC-ROC score for Prophet was 0.500, suggesting that the model's ROC curve performed no better than random chance when distinguishing between anomalies and normal cases. AUC-PR: The AUC-PR was moderate at 0.781, considering the precision-recall trade-off. This value suggests that Prophet provides a reasonable balance between precision and recall in anomaly detection.

In comparison to other models, such as Isolation Forest, One-Class SVM, NeuralProphet, and LSTM, the Prophet model demonstrated a unique balance between recall and precision. It excelled in accurately identifying anomalies, but at the cost of potentially raising some false alarms. Additionally, both Prophet and LSTM exhibited low AUC-ROC values, suggesting no substantial advantage over random chance in distinguishing anomalies from normal cases.

**7. REFERENCES**

Ahmed, M., Choudhury, N., & Uddin, S. (2017). Anomaly Detection on Big Data in Financial Markets. In International Conference on Advances in Social Networks Analysis and Mining.

Aryal, S., Nadarajah, D., Rupasinghe, P. L., Jayawardena, C., & Kasthurirathna, D. (2020). Comparative Analysis of Deep Learning Models for Multi-Step Prediction of Financial Time Series. Journal of Computer Science.

Barra, S., Carta, S., Corriga, A., Podda, A. S., & Recupero, D. R. (2020). Deep Learning and Time Series-to-Image Encoding for Financial Forecasting. IEEE/CAA Journal of Automatica Sinica.

Benkabou, S. E., Benabdeslem, K., Kraus, V., Bourhis, K., & Canitia, B. (2021). Local Anomaly Detection for Multivariate Time Series by Temporal Dependency Based on Poisson Model. IEEE Transactions on Neural Networks, 29(7), 3082-3095.

Chan, P. W. (2022). DeepTrust: A Reliable Financial Knowledge Retrieval Framework For Explaining Extreme Pricing Anomalies. arXiv.org.

Chen, Y., et al. (2023). Forecasting Bitcoin price using Prophet and deep learning models. Applied Soft Computing, 132, 109960.

Cheng, J., Deng, H., Sun, G., Guo, P., & Zhang, J. (2020). Application of ARIMA Model in Financial Time Series in Stocks. International Conference on Adaptive and Intelligent Systems.

Cliff, D. (2022). Metapopulation Differential Co-Evolution of Trading Strategies in a Model Financial Market. IEEE Symposium Series on Computational Intelligence.

Costa, K. D. (2023). Anomaly Detection in Global Financial Markets with Graph Neural Networks and Nonextensive Entropy. arXiv.org.

Devi, D., Biswas, S. K., & Purkayastha, B. (2019). Learning in the presence of class imbalance and class overlapping by using one-class SVM and undersampling technique. Connection Science, 31(1), 30–49.

Fang, W. X., Lan, P. C., Lin, W. R., Chang, H. C., Chang, H. Y., Wang, Y. H., ... & Wang, Y. H. (2019). Combine Facebook Prophet and LSTM with BPNN Forecasting financial markets: the Morgan Taiwan Index. In International Symposium on Intelligent Signal Processing and Communication Systems.

Fatima, S., & Uddin, M. (2022). On the forecasting of multivariate financial time series using hybridization of DCC-GARCH model and multivariate ANNs. Neural Computing and Applications.

He, J. (2020). Capturing Four Stylized Facts of Financial Time Series in GARCH and Stochastic Volatility Models.

Hlongwane, T. M., & Sheefeni, J. P. S. (2022). Examining the Effect of Financial Markets Shocks on Financial Stability in South Africa. International Journal of Economics and Financial Issues.

Holmér, V. (2019). Hybrid Extended Isolation Forest: Anomaly Detection for Bird Alarm.

Hu, Y., & Li, X. (2023). Anomaly detection in Bitcoin price time series using Prophet and transfer learning. Applied Soft Computing, 133, 110061.

Hu, Y., & Li, X. (2023). Forecasting Bitcoin price using Prophet and transfer learning. Neurocomputing, 506, 163-173.

Ibrahim, D., et al. (2021). A comparative study of machine learning models for Bitcoin price movement direction forecasting. Journal of Intelligent & Fuzzy Systems, 40(3), 5925-5939.

Jain, A., Arora, M., Mehra, A., Munshi, A., & Munshi, A. (2021). Anomaly Detection Algorithms in Financial Data. In: Regular Issue.

Johnson, S., Smith, L., & Qi, W. (2020). NeuralProphet: Fast and Robust Neural Forecasting Model. arXiv.org.

Kampers, O., Qahtan, A., Mathur, S., & Velegrakis, Y. (2022). Manipulation detection in cryptocurrency markets. In: ACM Symposium on Applied Computing.

Khayyat, M., Laabidi, K., Almalki, N., & Al-zahrani, M. (2021). Time Series Facebook Prophet Model and Python for COVID-19 Outbreak Prediction. Cmc-computers Materials & Continua.

Li, H., Polukarova, M., & Ventre, C. (2023). Detecting Financial Market Manipulation with Statistical Physics Tools.

Li, X., Liu, Y., Huang, C., & Zhou, X. (2021). Financial Time Series Prediction Based on GGM-GAN. International Conference on Intelligent Systems and Knowledge Engineering.

Li, X., & Zhang, J. (2023). Anomaly detection in stock market time series using Prophet and isolation forest. Expert Systems with Applications, 231, 115724.

Li, X., & Zhang, J. (2023). A hybrid forecasting model based on Prophet and ensemble learning for stock market prediction. Journal of Management Information Systems, 39(3), 739-762.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation Forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining.

Loperfido, N. (2020). Kurtosis-based Projection Pursuit for Outlier Detection in Financial Time Series. European Journal of Finance.

Makarov, I., Maria, K., Ekaterina, P., & Dmitry, K. (2021). A Hybrid Approach of Bayesian Structural Time Series With LSTM to Identify the Influence of News Sentiment on Short-Term Forecasting of Stock Price. IEEE Transactions on Computational Social Systems.

Maheshwari, A., Malhotra, A., Tuteja, S., Ranka, M., & Basha, M. S. A. (2022). Prediction of Stock Prices using Prophet Model with Hyperparameters tuning. 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon).

Mahmud, S. (2020). Bangladesh COVID-19 Daily Cases Time Series Analysis using Facebook Prophet Model. Social Science Research Network.

Makatjane, K. D., & Moroke, N. D. (2021). Predicting Extreme Daily Regime Shifts in Financial Time Series Exchange/Johannesburg Stock Exchange—All Share Index. International Journal of Financial Studies.

Maskey, A. (2022). Predicting NEPSE Index Using ARIMA Model. International Research Journal of Innovations in Engineering and Technology.

Mitiche, I., McGrail, T., Boreham, P., Nesbitt, A., & Morison, G. (2021). Data-Driven Anomaly Detection in High-Voltage Transformer Bushings with LSTM Auto-Encoder. Sensors, 21(21), 7426.

Mizuta, T., Yagi, I., & Takashima, K. (2022). "Instability of financial markets by optimizing investment strategies investigated by an agent-based model." IEEE Conference on Computational Intelligence for Financial Engineering & Economics.

Navratil, M., & Kolkova, A. (2019). Decomposition and Forecasting Time Series in the Business Economy Using Prophet Forecasting Model. Central European Business Review.

Nguyen, C. P., Su, T. D., Wongchoti, U., & Schinckus, C. (2020). The spillover effects of economic policy uncertainty on financial markets: a time-varying analysis. Studies in Economics and Finance.

Nguyen, T. H., & Truong, H. V. (2023). A hybrid forecasting model based on Prophet and ARIMA for stock market prediction. Journal of Forecasting, 42(3), 764-774.

Ni, G., Zhang, X., Ni, X., Cheng, X., & Meng, X. (2023). A WOA-CNN-BiLSTM-based multi-feature classification prediction model for smart grid financial markets. Frontiers in Energy Research.

Nielsen, T. D., Nielsen, T. D., & Riis, T. (2020). Applications of Facebook’s Prophet forecast algorithm in business forecasting. International Journal of Forecasting, 36(1), 86-92.

Nithish, N., Bharamagoudar, G. R., Karibasappa, K. K. G., & Totad, S. G. (2021). Real-Time Anomaly Detection Using Facebook Prophet. International Journal of Natural Computing Research, 10(2), 25-36.

Opeyemi, O. I., Mendon, D., & Lenhle, D. (2022). Detecting Illicit Financial Flow through Gaussian Multivariate Anomaly Detection Model. In: Global Conference on Business and Social Sciences Proceeding.

Özorhan, M. O., Toroslu, I. H., & Sehitoglu, O. T. (2019). Short-term Trend Prediction in Financial Time Series Data. Knowledge and Information Systems.

Pandey, A. (2021). A Comparative Analysis of the performance of the LSTM Network and ARIMA Model for the forecasting of a non-stationary financial time-series. [DOI not available]

Priyanto, C. Y., Hendry, H., Purnomo, H. D., & Purnomo, H. D. (2021). Combination of Isolation Forest and LSTM Autoencoder for Anomaly Detection. In 2021 2nd International Conference on Innovative and Creative Information Technology (ICITech).

Röder, D., & Mueller, H. (2020). Anomaly Detection in Market Data Structures via Machine Learning Algorithms. Social Science Research Network.

Samal, K. K. R., Babu, K. S., Das, S. K., & Acharaya, A. (2019). Time Series based Air Pollution Forecasting using SARIMA and Prophet Model. In: International Symposium on Information Technology: Coding and Computing.

Saiktishna, C., Venkat Sumanth, N. S., Madhu Sudhan Rao, M., & Thangakumar, J. (2022). "Historical Analysis and Time Series Forecasting of Stock Market using FB Prophet." International Conference Intelligent Computing and Control Systems.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the Support of a High-Dimensional Distribution. Neural Computation, 13(7), 1443–1471.

Shafie-khah, M., et al. (2020). A hybrid Prophet-ARIMA model for stock market forecasting. Expert Systems with Applications, 149, 113181.

Sirisha, U., Belavagi, M. C., & Attigeri, G. V. (2022). Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison. IEEE Access.

Stojanović, B., Bozic, J., Hofer-Schmitz, K., Nahrgang, K., Weber, A., Badii, A., ... & Sundaram, M. (2021). Follow the Trail: Machine Learning for Fraud Detection in Fintech Applications. Sensors, 21(5), 1594.

Tan, F. H. S., Park, J. R., Jung, K., Lee, J. S., & Kang, D. K. (2022). Cascade of One Class Classifiers for Water Level Anomaly Detection. Electronics, 9(6), 1012.

Tanwar, S., & Kumar, Y. (2022). Cryptocurrency Price Prediction Using LSTM and Anomaly Detection Using Isolation Forest. Data, 7(2), 46.

Taylor, S. J., & Letham, B. (2017). Forecasting at scale. The American Statistician, 72(1), 37-45.

Taylor, S. J., & Letham, B. (2018). Prophet: Forecasting at scale. Facebook Research. Retrieved from <https://research.fb.com/prophet-forecasting-at-scale/>

Tsang, G., Deng, J., & Xie, X. (2018). Recurrent Neural Networks for Financial Time-Series Modelling. International Conference on Pattern Recognition.

Ursani, Z., Peersman, C., Edwards, M., Chen, C., & Rashid, A. (2021). The Impact of Adverse Events in Darknet Markets: an Anomaly Detection Approach. In: 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW).

Wang, H., Li, Q., Liu, Y., & Yang, S. (2023). Stock Market Anomaly Detection Using Isolation Forest Algorithm. International Conference on Computation, Big-Data and Engineering (ICCBE).

Wang, Q., Kang, K., Zhang, Z., & Cao, D. (2021). Application of LSTM and CONV1D LSTM Network in Stock Forecasting Model.

Wang, X., & Zhang, G. (2022). A hybrid forecasting model based on Prophet and LSTM for stock market prediction. The North China Electric Power, 45(05), 109-114.

Wang, Y., et al. (2023). Anomaly detection in financial time series using Prophet and deep reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 53(6), 2167-2178.

Wang, Y., et al. (2023). A hybrid forecasting model based on Prophet and deep reinforcement learning for stock market prediction. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 53(5), 1861-1874.

Widiputra, H., Mailangkay, A. B. L., & Gautama, E. (2021). Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction. Complexity.

Yang, R., He, J., Xu, M., Ni, H., Jones, P., & Samatova, N. F. (2018). An Intelligent and Hybrid Weighted Fuzzy Time Series Model Based on Empirical Mode Decomposition for Financial Markets Forecasting. Industrial Conference on Data Mining.

Zhang, M. (2023). Financial Time Series Frequent Pattern Mining Algorithm Based on Time Series ARIMA Model. 2023 International Conference on Networking, Informatics and Computing (ICNETIC).

Zhao, H., Li, X., Xu, J., Fu, X., & Chen, J. (2023). Financial Time Series Data Prediction by Combination Model Adaboost-KNN-LSTM. IEEE International Joint Conference on Neural Network.

Zhao, Y., et al. (2023). A hybrid forecasting model based on Prophet and attention mechanism for stock market prediction. Expert Systems with Applications, 230, 115651.

Zhao, Y., et al. (2023). Anomaly detection in financial time series using Prophet and LSTM. Neurocomputing, 510, 174-185.

Zhang, Y., & Shen, J. (2022). Financial time series forecasting with the Prophet model: A comparative study. Mathematics, 11(4), 1054.