**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.

3.2.3.1 Box Plot ................................................................................................................. pag.

3.2.3.2 Density Plot ............................................................................................................ pag.

3.2.3.3 Timeline plot ......................................................................................................... pag.

3.2.3.4 Autocorrelation and partial autocorrelation ………………………………………………….. 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 Characteristics comparison ......................................................................................... pag.

4.5.3 Prophet Model vs. Other Algorithms ......................................................................... pag.

4.5.4 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.

4.7.5.3 Cumulative returns comparison ............................................................................... pag.

5. Anomaly Detection

5.1 Introduction ..................................................................................................................... pag.

5.2. Types of Anomalies .......................................................................................................... pag.

5.3 Anomaly Detection Plots………………………………………………………………………..……………….. pag.

5.3.1 Timeline with anomaly detection ................................................................................ pag.

5.4 Implementing Anomaly Detection Models………………………………………………………………….. pag.

5.4.1 Isolation Forest ........................................................................................................... pag.

5.4.2 One-Class SVM (Support Vector Machine): ................................................................. pag.

5.4.3 Prophet Model: ............................................................................................................ pag.

5.4.4 NeuralProphet: ............................................................................................................ pag.

5.5 Performance Evaluation Metrics for Anomaly Detection…………………………………………….. pag.

5.5.1 Precision, Recall, and F1-Score ..................................................................................... pag.

5.5.2 Area Under the ROC Curve (AUC-ROC) ......................................................................... pag.

5.6 Characteristics comparison .............................................................................................. pag.

6. Results and Discussion………………………………………………………………………………………………….. pag.

6.1 Comparing Anomaly Detection Methods ......................................................................... pag.

7. Integrated Analysis…………………………………………………………………………………….……………….. pag.

6.1 Forecasted Values ............................................................................................................ pag.

6.2 Confidence Intervals ......................................................................................................... pag.

6.3 Trend Analysis ................................................................................................................... pag.

6.4 Seasonal Decomposition .................................................................................................. pag.

6.5 Residual Component ........................................................................................................ pag.

6.6 Anomalies Overview ........................................................................................................ pag.

6.7 Anomaly Characteristics .................................................................................................... pag.

8. Conclusion .............................................................................................................................. pag.

9. References ............................................................................................................................. pag.

**Abstract**

Accurate anomaly detection and time series forecasting in financial markets are crucial for informed decision-making and risk management. This paper presents a comprehensive study of anomaly detection and time series forecasting in the Irish finance market, specifically focusing on stock prices and market trends. To achieve this, I propose an integrated approach that combines the use of Facebook's Prophet model with insights from relevant research papers. Specifically, I draw inspiration from the " Time Series Forecasting Using FB-Prophet” (Kirti 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) papers.

**1. Introduction**

The financial markets hold a crucial position in contemporary economies, functioning as indicators of economic well-being while also serving as platforms for investment, trading, and risk management. Over time, the Irish finance market, situated within the larger framework of global financial markets, has undergone significant expansion and evolution. In this dynamic landscape, the ability to accurately detect anomalies and forecast future market trends is of paramount importance for investors, policymakers, and financial institutions. Effective decision-making in the Irish finance market relies heavily on robust data-driven tools and techniques.

Time series analysis has long been an essential component of financial market research, enabling the examination of historical data to uncover patterns, trends, and anomalies. Anomaly detection holds significant relevance for early warning systems in risk management and fraud detection. Additionally, time series forecasting is fundamental for making informed investment decisions, optimizing portfolio strategies, and developing policies to navigate the volatile nature of financial markets.

The Prophet model, developed by Taylor and Letham (2017), has gained popularity for its versatility in capturing various time series patterns and seasonality. This model, based on a decomposable time series with three main components - trend, seasonality, and holiday effects - has demonstrated remarkable success in forecasting applications across various domains. However, its performance in the specific context of the Irish finance market remains largely unexplored.

This thesis aims to bridge this gap by focusing on two key objectives. First, it seeks to evaluate the efficacy of the Prophet model in anomaly detection within the Irish finance market. Anomalies in financial time series can signal critical market events, such as economic crises or trading irregularities, and detecting them promptly can be a decisive factor in risk mitigation. Second, this research aims to assess the Prophet model's forecasting performance in the Irish finance market, offering insights into its potential applications for investors and financial institutions. (rever com os meu objectivos 1.2)

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.

In the following chapters, we will delve into the theoretical foundations of time series analysis and the Prophet model, followed by a comprehensive empirical analysis of the Irish finance market data. Through rigorous evaluation, this research endeavours to provide a nuanced understanding of the Prophet model's capabilities and limitations within the Irish financial landscape, ultimately aiding stakeholders in their pursuit of enhanced market intelligence.

**1.1 Background and Motivation**

**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.

**1.3 Scope and Limitations**

Scope:

1. **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.
2. **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.
3. **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.
4. **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.
5. **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.
6. **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:

1. **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.
2. **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.
3. **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.
4. **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.
5. **Model Hyperparameters:** The performance of the Prophet model may depend on its hyperparameters.
6. **Interpretability:** While Prophet is known for its ease of use, its inner workings may not be as interpretable as traditional statistical models.
7. **Data Privacy:** The use of financial data raises privacy concerns. This study complies with data privacy regulations and guidelines.
8. **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 For Time Series Forecasting Techniques

2.1.1 Prophet Model

Epidemiological Forecasting:

In the context of epidemiology, the Prophet model has been utilized for disease outbreak prediction. Khayyat et al. (2021) applied the Prophet model to predict COVID-19 outbreaks, demonstrating its effectiveness in time series forecasting of daily cases. Similarly, Mahmud (2020) conducted time series analysis of daily COVID-19 cases in Bangladesh using the Prophet model, highlighting its suitability for epidemiological data.

Environmental Forecasting:

Time series forecasting has also been instrumental in environmental studies. Samal et al. (2019) employed the Prophet model for air pollution forecasting, demonstrating its effectiveness in predicting air quality based on historical data. Additionally, Gupta et al. (2022) used the Prophet model to forecast solar power generation, emphasizing its relevance in renewable energy applications.

Financial Forecasting:

Financial markets often involve complex time series data, and the Prophet model has found applications in this domain as well. Fang et al. (2019) combined the Prophet model with LSTM and BPNN for forecasting financial markets, showcasing its adaptability in predicting stock prices. Mizuta et al. (2022) explored financial market instability and investment strategies using an agent-based model, highlighting the importance of accurate time series forecasting.

Hybrid Forecasting Approaches:

Researchers have also integrated the Prophet model with other techniques to enhance forecasting accuracy. Arslan (2022) developed a hybrid forecasting model combining LSTM and Prophet for energy consumption, demonstrating the potential for combining multiple methods to improve predictions. Additionally, Navratil and Kolkova (2019) decomposed and forecasted time series data in business economics using the Prophet model.

Conclusion:

In conclusion, the Prophet model has emerged as a powerful tool for time series forecasting in various domains. Its flexibility, simplicity, and adaptability make it suitable for a wide range of applications, including epidemiological predictions, environmental forecasting, and financial market analysis. Researchers continue to explore and expand the capabilities of the Prophet model, making it an essential tool in the field of time series forecasting.

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

*ARIMA and SARIMA Models:*

ARIMA and SARIMA models are widely employed for univariate time series forecasting due to their simplicity and effectiveness in capturing various patterns.

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 (Makatjane & Moroke, 2021).

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 (Adineh et al., 2021).

Sirisha et al. (2022) conducted a comparison study between ARIMA, SARIMA, and LSTM models, offering valuable insights into the relative performance of these models in profit prediction (Sirisha et al., 2022).

Pandey (2021) performed a comparative analysis between the LSTM network and ARIMA model, focusing on the forecasting of non-stationary financial time series, shedding light on the strengths and limitations of each model (Pandey, 2021).

Cheng et al. (2020) applied ARIMA models to financial time series in stocks, demonstrating the versatility of ARIMA in modeling and forecasting stock prices (Cheng et al., 2020).

Maskey (2022) used the ARIMA model to predict the NEPSE Index, showcasing the model's applicability in forecasting stock market indices (Maskey, 2022).

Zhang (2023) proposed a financial time series frequent pattern mining algorithm based on the ARIMA model, offering a unique perspective on extracting patterns in financial time series data (Zhang, 2023).

*GARCH Models:*

GARCH models are widely used to capture volatility patterns in financial time series data.

Khan et al. (2023) applied GARCH models to investigate the impact of the COVID-19 pandemic on financial market volatility, highlighting the importance of modeling volatility during crisis periods (Khan et al., 2023).

Fatima and Uddin (2022) 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 (Fatima & Uddin, 2022).

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 (Nguyen et al., 2020).

He (2020) explored stylized facts of financial time series using GARCH and stochastic volatility models, contributing to our understanding of financial market dynamics (He, 2020).

Aghabazaz et al. (2022) introduced a time-varying GARCH mixed-effects model for isolating high- and low-frequency volatility and co-volatility, emphasizing the flexibility of GARCH in capturing different volatility components (Aghabazaz et al., 2022).

*Conclusion:*

ARIMA, SARIMA, and GARCH models play crucial roles in forecasting financial time series. ARIMA and SARIMA models are suitable for capturing trend and seasonality patterns in univariate time series data, while GARCH models excel in modeling and forecasting volatility. Researchers and practitioners can choose these models based on their specific forecasting needs and the characteristics of the financial time series data.

2.1.3 Deep learning model (NeuralProphet, LSTM)

*NeuralProphet*

Financial Time Series Forecasting with NeuralProphet:

The NeuralProphet model, an extension of the popular Prophet model, has garnered interest due to its application in financial forecasting. Yang et al. (2018) proposed an intelligent and hybrid weighted fuzzy time series model based on empirical mode decomposition for financial markets forecasting. While they used a different approach, the hybrid concept aligns with the NeuralProphet's adaptability to different domains.

Multivariate Forecasting:

Widiputra et al. (2021) introduced a multivariate CNN-LSTM model for predicting multiple parallel financial time series. Although they did not use NeuralProphet explicitly, their approach aligns with the idea of leveraging neural networks for multivariate financial forecasting, which is a hallmark of the NeuralProphet model.

Outlier Detection:

In financial time series, outlier detection is crucial. Loperfido (2020) presented a kurtosis-based projection pursuit method for outlier detection in financial time series. While they did not employ NeuralProphet, this study highlights the importance of preprocessing and identifying anomalies, which can complement NeuralProphet's forecasting capabilities.

Time Series-to-Image Encoding:

Barra et al. (2020) explored deep learning and time series-to-image encoding for financial forecasting. While they focused on a different approach, the use of deep learning in financial forecasting aligns with the capabilities of NeuralProphet in handling complex financial data.

Short-Term Trend Prediction:

Özorhan et al. (2019) addressed short-term trend prediction in financial time series data. While their method differs from NeuralProphet, the concept of capturing short-term trends resonates with NeuralProphet's ability to capture daily patterns and seasonality in financial data.

Impact of COVID-19 on Financial Markets:

Sansa (2020) investigated the impact of COVID-19 on financial markets. Although this study did not use NeuralProphet, it underscores the relevance of timely and accurate forecasting tools in reacting to unforeseen events—a capability NeuralProphet can provide.

New Entropic Measures:

Lerner (2023) introduced a new entropic measure for the causality of financial time series. While not directly related to NeuralProphet, this research emphasizes the importance of developing innovative metrics to enhance financial forecasting.

*Conclusion:*

In conclusion, financial time series forecasting remains a dynamic field where innovation and adaptability are paramount. The NeuralProphet model, as an extension of the Prophet model, has shown promise in handling complex financial data. While the specific application of NeuralProphet in financial forecasting literature is limited, its capabilities align with the evolving needs of financial analysts and decision-makers. Researchers continue to explore and adapt neural forecasting models to capture the intricacies of financial markets, offering potential improvements in forecasting accuracy.

*LSTM*

LSTM in Financial Time Series Forecasting:

The use of LSTM models in financial time series forecasting has garnered significant attention. Researchers have explored various aspects of LSTM-based forecasting, including architecture, feature engineering, and performance evaluation.

Multimodal Forecasting:

Zhao et al. (2023) proposed a combination model, Adaboost-KNN-LSTM, for financial time series data prediction. While combining different models, LSTM serves as a crucial component, highlighting its effectiveness in multimodal forecasting.

GAN-Based Approaches:

Li et al. (2021) introduced GGM-GAN for financial time series prediction. Although this study incorporates GAN (Generative Adversarial Network), LSTM is utilized for the modeling part, emphasizing its role in capturing complex dependencies in financial data.

Transaction Volume Estimation:

Bozkan et al. (2023) focused on transaction volume estimation in financial markets using LSTM. This study demonstrates LSTM's ability to handle diverse financial data types, including transaction volumes.

Cryptocurrency and Portfolio Strategies:

Studies such as Zou et al. (2022) explored LSTM-based strategies for Bitcoin and gold investments, while Wang et al. (2021) applied LSTM to stock forecasting. These investigations emphasize LSTM's versatility in analyzing different financial assets and formulating investment strategies.

Optimization and Noise Reduction:

Qian (2023) reviewed stock price prediction methods based on LSTM, highlighting its applications in optimizing trading strategies. Karimi Dastgerdi and Mercorelli (2022) investigated LSTM models with Kalman filtering for noise elimination, underscoring LSTM's adaptability to address real-world data challenges.

Ensemble Approaches:

He et al. (2023) developed a deep learning ensemble model for financial time series forecasting, showcasing LSTM's role within ensemble frameworks.

Incorporating News Sentiment:

Ray et al. (2021) adopted a hybrid approach, combining Bayesian Structural Time Series with LSTM, to analyze the influence of news sentiment on stock price forecasting. This study demonstrates LSTM's integration with other methods to enhance predictive accuracy.

Comparative Analysis:

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.

Financial Market Analysis:

LSTM models have also been used in financial market analysis (Makarov et al., 2021; Ni et al., 2023). These studies underscore LSTM's application in solving classification problems and analyzing multidimensional time series data.

*Conclusion:*

In conclusion, LSTM models have become indispensable tools in financial time series forecasting and analysis. Researchers continue to explore their capabilities in modeling complex dependencies, handling diverse data types, and improving trading and investment strategies. As the field of financial data analysis evolves, LSTM models remain at the forefront of innovation, providing valuable insights and predictions for financial markets.

**2.2 For Anomaly Detection Approaches**

2.2.1 Prophet Model

Anomaly detection techniques using prophet model:

Prophet Model: A Robust Approach

The Prophet model, introduced by Taylor and Letham (2018), has gained prominence as a versatile framework for time series forecasting and anomaly detection. Its adaptability to missing data, incorporation of holiday effects, and handling of special events make it a valuable tool in identifying anomalies within temporal data.

Graph Neural Networks for Financial Markets

Costa (2023) proposed an innovative approach to anomaly detection in global financial markets using Graph Neural Networks (GNNs) and nonextensive entropy. This study underscores the importance of GNNs in capturing intricate relationships within financial data, contributing to effective anomaly detection.

Darknet Markets and Anomalies

Ursani et al. (2021) explored the impact of adverse events in darknet markets using anomaly detection techniques. By analysing transaction data, the study demonstrates the efficacy of anomaly detection in identifying suspicious activities within darknet markets.

Collusion Detection in Stock Markets

Madurawe et al. (2021) introduced a collusion set detection method within the stock market context. Through graph clustering and anomaly detection, this research identifies potentially collusive behavior among market participants, highlighting the role of anomaly detection in preserving market integrity.

Anomaly Detection in Financial Data

Jain et al. (2021) conducted a comprehensive study on anomaly detection algorithms in financial data. Their work provides an extensive overview of techniques employed for identifying anomalies in financial datasets.

STAD-GAN: Unsupervised Anomaly Detection

Zhang et al. (2022) presented an unsupervised anomaly detection method, STAD-GAN, designed for multivariate time series data. This approach demonstrates the effectiveness of generative adversarial networks in identifying anomalies within complex temporal datasets.

Local Anomaly Detection in Multivariate Time Series

Benkabou et al. (2021) proposed a local anomaly detection method based on temporal dependency using a Poisson model. Their research contributes to the development of techniques focused on detecting anomalies in multivariate time series data.

Manipulation Detection in Cryptocurrency Markets

Kampers et al. (2022) addressed manipulation detection in cryptocurrency markets, underscoring the importance of anomaly detection in maintaining the integrity of digital financial ecosystems.

Prophet and LSTM for Financial Forecasting

Fang et al. (2019) explored the combination of the Prophet model with Long Short-Term Memory (LSTM) networks for forecasting financial markets, specifically the Morgan Taiwan Index. This research showcases the integration of traditional and deep learning-based time series forecasting techniques.

Detecting Illicit Financial Flow

Opeyemi et al. (2022) proposed a Gaussian multivariate anomaly detection model for detecting illicit financial flows. Their study highlights the relevance of anomaly detection in identifying suspicious financial activities.

Real-Time Anomaly Detection Using Facebook Prophet

Nithish et al. (2021) applied the Facebook Prophet method to real-time anomaly detection. This research emphasizes the practicality of the Prophet model in monitoring and flagging anomalies in time series data streams.

Conclusion

In conclusion, anomaly detection in time series data is a critical field with applications spanning multiple domains. The Prophet model, along with emerging techniques such as graph neural networks, generative adversarial networks, and local anomaly detection methods, contributes to the growing arsenal of tools for identifying anomalies in temporal data. These advancements are crucial for maintaining the integrity and security of various systems and datasets.

2.2.2 Ensemble Anomaly Detection (Isolation Forest)

The Isolation Forest model, introduced by Liu et al. (2008) as a machine learning-based anomaly detection technique, has gained popularity for its effectiveness in isolating anomalies within complex datasets. This literature review explores the application of the Isolation Forest model in financial markets and related fields.

*Foundations of Isolation Forest*

The Isolation Forest algorithm, originally proposed by Liu et al. (2008), is based on the principle that anomalies are rare and tend to have shorter paths in a binary tree structure. This approach offers several advantages for anomaly detection:

Efficiency: The Isolation Forest model is computationally efficient, making it suitable for large-scale datasets (Liu et al., 2008).

Scalability: It is capable of handling high-dimensional data effectively (Liu et al., 2008).

*Applications in Financial Markets*

Financial Fraud Detection: Kumar et al. (2021) applied the Isolation Forest algorithm to detect financial fraud in plastic payment card transactions, showcasing its effectiveness in identifying anomalous transactions.

Stock Market Anomaly Detection: Isolation Forest has been utilized for anomaly detection in stock market data (Wang et al., 2023; Holmér, 2019). These studies highlight its potential to identify unusual trading behaviours and market irregularities.

Cryptocurrency Price Analysis: Researchers have employed Isolation Forest for predicting and detecting anomalies in cryptocurrency price data (Tanwar & Kumar, 2022; Priyanto et al., 2021).

*Beyond Financial Markets*

The versatility of the Isolation Forest model extends beyond financial markets, as it has found applications in various domains:

IoT Security: AbuAlghanam et al. (2022) introduced a fusion-based anomaly detection system for the Internet of Things (IoT), leveraging a modified Isolation Forest algorithm to enhance IoT security.

Machine Monitoring: Li et al. (2021) proposed a similarity-measured Isolation Forest for anomaly detection in machine monitoring data, underlining its applicability in industrial settings.

*Comparative Studies*

Several studies have compared the Isolation Forest model with other anomaly detection methods:

Smolen and Benova (2023) conducted a comparative analysis between Isolation Forest and autoencoders for network anomaly detection.

Fan et al. (2021) performed a comparative study between Isolation Forest and LOF (Local Outlier Factor) in data mining applications.

*Future Directions and Challenges*

While the Isolation Forest model has demonstrated its effectiveness in anomaly detection across various domains, ongoing research is needed to address specific challenges, such as parameter tuning and interpretability. Furthermore, combining Isolation Forest with other machine learning techniques, such as LSTM autoencoders (Priyanto et al., 2021), offers exciting opportunities for improving anomaly detection in financial markets and beyond.

*Conclusion*

the Isolation Forest model has proven to be a highly effective tool for identifying anomalies in various fields, ranging from financial markets to IoT security and machine monitoring. Its efficiency and scalability, coupled with its capability to detect anomalies even in complex, high-dimensional datasets, make it a preferred choice for both researchers and professionals. As technology continues to advance, it is likely that the Isolation Forest and similar anomaly detection methods will continue to be essential for maintaining the security and reliability of financial systems and other critical applications.

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

A Brief Overview

the One-Class SVM, introduced by Schölkopf et al. (2001), stands out for its ability to detect outliers in unimodal datasets. In the context of financial markets, it has been widely applied to identify extreme pricing anomalies (Chan, 2022).

*Applications in Financial Markets*

Machine learning methods, including the One-Class SVM, have found diverse applications in financial markets. Stojanović et al. (2021) explored machine learning for fraud detection in fintech applications, demonstrating the effectiveness of these methods in detecting fraudulent transactions. 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.

*Challenges and Future Directions*

Despite significant progress, challenges persist in anomaly detection within financial markets. High-frequency trading data poses unique challenges due to its volume and velocity, requiring scalable solutions. Moreover, the detection of market manipulation and subtle anomalies remains an ongoing research area (Li et al., 2023).

*Conclusion*

anomaly detection methods, including the One-Class SVM model, have become indispensable in 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.

2.3.4 Deep learning model (NeuralProphet, LSTM)

*NeuralProphet*  
A Brief Overview

The NeuralProphet model, introduced by Johnson et al. (2020), represents a notable example of machine learning techniques applied to time series data, which is prevalent in financial markets.

*Applications in Financial Markets*

Machine learning methods, including the NeuralProphet model, have found diverse applications in financial markets. Chan (2022) proposed "DeepTrust," a framework for explaining extreme pricing anomalies, showcasing the use of deep learning in understanding financial anomalies. Stojanović et al. (2021) employed machine learning for fraud detection in fintech applications, emphasizing the importance of anomaly detection in safeguarding financial transactions.

*Challenges and Future Directions*

Despite the promising results, challenges persist in anomaly detection within financial markets. High-frequency trading data presents unique challenges due to its volume and speed, necessitating scalable solutions. Additionally, the detection of market manipulation and subtle anomalies remains an ongoing research area (Li et al., 2023).

*Conclusion*

In conclusion, anomaly detection methods, including advanced machine learning models like the NeuralProphet model, play a pivotal role in preserving the integrity of financial markets. These methods offer robust solutions to identify extreme pricing anomalies, fraudulent activities, and market irregularities. As financial markets continue to evolve, further advancements in machine learning techniques are expected to contribute to more effective anomaly detection and risk management.

*LSTM*

*An Overview*

LSTM models, a type of recurrent neural network (RNN), have gained popularity for their ability to capture sequential dependencies in time series data. LSTM's ability to model long-term dependencies makes it well-suited for anomaly detection in financial time series.

*Applications in Financial Markets*

Machine learning models, especially LSTM-based approaches, have found diverse applications in financial markets. For instance, Ahmed et al. (2017) applied anomaly detection on big data in financial markets using LSTM, demonstrating its effectiveness in identifying unusual market behaviors. Similarly, Mitiche et al. (2021) utilized LSTM auto-encoders for data-driven anomaly detection in high-voltage transformer bushings.

*Challenges and Future Directions*

Challenges in anomaly detection in financial markets with LSTM models include handling high-frequency trading data, model scalability, and detecting subtle market manipulations. Ongoing research, such as Feng et al. (2021), explores unsupervised anomaly detection techniques to address these challenges.

*Conclusion*

In conclusion, LSTM models have emerged as valuable tools for anomaly detection in financial markets due to their ability to model complex temporal dependencies. These models have demonstrated effectiveness in detecting anomalies in various financial datasets, contributing to market stability and security. As financial markets continue to evolve, further research and advancements in LSTM-based anomaly detection methods are expected.

**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:

1. 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 S&P 500 frequently acts as a standard to gauge the performance of the United States 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.
2. 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.
3. 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.
4. 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:

1. 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.
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.

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.

**3.1.2 Sources**

**All the datasets used for this study are available for public domain**:

downloaded several financial indexes’ datasets at Finance. Yahoo and Euronext.

For the study, it was created six datasets BEL20 (Belgium), FTSE100 (UK), CAC40 (France), ISEQ20 (Ireland), DAX40 Germany), PSI20 (Portugal), ISEQ20 (Ireland) from both sites.

The datasets 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 (fig. 01)

# **3.2.2.2 Outliers**

Empty DataFrame

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 (fig 02), however, we’ll see later on our anomaly detection section, we’ll find several.

**3.2.2 Basic statistics**

**A close up of a number

Description automatically generated**

***Fig. 03 Basiq statistics from Iseq20***

From Fig 03, there are 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 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.

The maximum 'Close' price observed in the dataset is roughly 1545.07, indicating the highest value during the analysed time-period.

In conclusion: The 'Close' prices in the iseq20\_df 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 a few words

Description automatically generated with medium confidence**

***Fig. 04 Data distribution with 10 bins from Iseq20***

**A graph with many blue squares

Description automatically generated with medium confidence**

***Fig. 05 Frequency distribution with 10 bins from Iseq20***

From Fig 04 and 05, 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**

3.2.3.1 Box Plot

A green rectangular object with white lines

Description automatically generated with medium confidence

***Fig. 06 Box Plot from Iseq20***

From Fig. 06, there are 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 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.

The maximum 'Close' price observed in the dataset is roughly 1545.07, indicating the highest value during the analysed time-period.

In conclusion:

The 'Close' prices in the iseq20\_df 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.

3.2.3.2 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. 07 Density Plot from Iseq20***

The density estimates provide insights into the distribution and likelihood of 'Close' prices within the dataset (fig.07):

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.

3.2.3.3 Timeline plot

A graph showing a line

Description automatically generated with medium confidence

***Fig. 08 Time Plot from Iseq20***

From Fig. 08, we can reach the following conclusion:

***Periods of Index Growth:***

***Overall Uptrend (01/07/2021 - 10/10/2022)***: During this period, the index showed consistent growth, starting at around 1449 points and peaking at 1576 points. This was marked by a positive sentiment in the market and investor confidence in the economy.

***Post-COVID Recovery (03/11/2020 - 17/02/2021)***: After the initial impact of the COVID-19 pandemic, the index experienced significant growth, rising from roughly 1200 points to approximately 1473 points. This period signified optimism about economic rebound.

***Steady Growth (29/03/2021 - 06/05/2022):*** From March 2021 to May 2022, the index displayed steady growth, going from 1257 points to 1243 points, indicating a stable and positive market environment.

***Periods of Index Decrease:***

***Market Correction (07/05/2022 - 21/06/2022):*** During this phase, the index underwent a correction, falling from about 1243 points to 1089 points. This could be attributed to concerns such as inflation and potential interest rate hikes.

***Volatility (22/06/2022 - 10/10/2022):*** Following the correction, the index experienced increased volatility, fluctuating between 1089 and 1219 points. This period was marked by uncertainty about economic policies and global events.

***Downturn (08/03/2022 - 15/03/2022)***: In early March 2022, the index sharply declined from around 1218 points to 1144 points within a short timeframe, which may have been influenced by concerns over interest rates and geopolitical tensions.

***Consolidation (16/03/2022 - 30/06/2022)***: From mid-March to the end of June 2022, the index experienced a consolidation phase, with limited growth. It ranged between 1144 and 1247 points during this period, indicating a cautious market sentiment.

*General Observations:*

The index's performance is influenced by a variety of factors, including economic data, global events, and market sentiment. Growth periods usually align with positive economic indicators, while downturns are associated with uncertainties or corrections. The index's movements include steady growth, sharp declines, and consolidation phases, reflecting the cyclical nature of financial markets.

3.2.3.4 Autocorrelation and partial autocorrelation

A graph of a function

Description automatically generated

***Fig. 09 Autocorrelation Plot from Iseq20***

The ACF values decline as the lag between data points increases (fig 09). 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. 10 Autocorrelation Plot from Iseq20

PACF values exhibit a more intricate pattern compared to ACF. Several PACF values significantly differ from zero (fig. 10). 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 computer code

Description automatically generated

Fig. 10 Bayesian optimization from Iseq20

From Fig. 10, we have:

Best Hyperparameters: The Bayesian optimization process determined the optimal hyperparameters for the Prophet model. These hyperparameters are crucial for the model's performance, and they were found to be changepoint\_prior\_scale = 0.1372 and holidays\_prior\_scale = 0.7634.

Model Fit: The Prophet model was trained on the provided time series data using the best hyperparameters. It's important to note that the model's performance heavily depends on the quality and nature of the underlying data.

*Performance Metrics:*

MAE (Mean Absolute Error): This metric indicates that, on average, the model's predictions deviate from the actual values by approximately 227.79 points. Lower MAE values are desirable, but the interpretation should consider the scale of your data.

MSE (Mean Squared Error): With an MSE of 62856.42, the model's errors are squared before averaging. This value quantifies the average squared difference between predicted and actual values.

RMSE (Root Mean Squared Error): The RMSE of 250.71 represents the square root of the MSE. It provides a measure of the average magnitude of errors in the same unit as the target variable. Lower RMSE values indicate better model performance.

MAPE (Mean Absolute Percentage Error): This metric measures the percentage difference between predicted and actual values. An MAPE of 17.37% means, on average, the model's predictions have an absolute percentage error of 17.37%. Lower MAPE values indicate better accuracy.

R-squared (R2): The value of -2.65 indicates that the model does not fit the data well. R-squared measures the proportion of the variance in the dependent variable that is predictable from the independent variables. A negative value suggests that the model's predictions are worse than simply using the mean of the target variable.

*Overall Assessment:*

The Bayesian optimization process helped in finding hyperparameters for the Prophet model. However, the model's performance, as indicated by the performance metrics, seems suboptimal. The negative R-squared suggests that the model might not be suitable for capturing the underlying patterns in your data. It's important to consider other factors, such as data quality, feature engineering, and model selection, to potentially improve forecasting accuracy. Additionally, further analysis may be needed to understand the reasons behind the poor model fit and explore alternative modeling approaches.

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. 11 Prophet Model Forecast with Confidence Intervals on optimized hyperparameters from Iseq20.

From Fig. 11, we have:

Original Data:

Forecasted Values: The forecasted values span from August 24, 2020, to August 22, 2023, comprising 764 data points. Starting at 1142.59 and trending upwards, they reach around 1556.67 by August 22, 2023, indicating a positive trend.

Upper Confidence Interval: The upper confidence interval (CI) values, represented in the 'yhat\_upper' column, provide an upper limit for the forecasted stock prices. These values start at 1188.04 and show a potential upper boundary for stock prices, indicating a maximum of around 1602.52 by August 22, 2023.

Lower Confidence Interval: Conversely, the lower confidence interval (CI) values, shown in the 'yhat\_lower' column, provide a lower limit for forecasted stock prices. Starting at 1097.14, these values indicate the lowest potential stock price levels, suggesting prices may not drop below 1513.04 by August 22, 2023.

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. 12 Trend Analyse from Iseq20.

The explanation is in 3.2.3.3 Timeline plot.

4.3.2 Seasonal decomposition

A blue and white barcode

Description automatically generated

Fig. 13 Seasonal decomposition Analyse from Iseq20.

The seasonal component (fig. 13) 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. 14 Residual component Analyse from Iseq20.

The residual component (fig. 14) 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 August 28, 2020, there is a small negative residual value of approximately -0.08 points, suggesting a minor decrease in closing prices beyond expected patterns.

**4.4 Performance Evaluation**

**4.4.1 Performance Evaluation Metrics**

4.4.1.1 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 [4].

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

4.4.1.2 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

4.4.1.3 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

4.4.1.4 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 you 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

4.4.1.5 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 black text on a white background

Description automatically generated

Fig. 15 Performance Evaluation Metrics from Iseq20.

From Fig 15, 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.

i's approximately 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 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.

The MAPE of approximately 17.11% indicates that, on average, the model's percentage errors are moderate.

Overall, there may be room for improvement in the model's accuracy and explanatory power. Further model refinement, feature engineering, or parameter tuning may be necessary to enhance its forecasting performance.

**4.5 Model Comparison**

**4.5.1 Models compared to Prophet Model**

4.5.1.1 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.

4.5.1.2 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.

4.5.1.3 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.

4.5.1.4 **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.

4.5.1.5 GARCH (Generalized Autoregressive Conditional Heteroskedasticity)

**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** **characteristics comparison**



Fig. 16 Prophet Model compared to other models (characteristics) (rever<br>)

4.5.3 Prophet Model vs. Other Algorithms

A number and numbers on a white background

Description automatically generated

***Fig. 17 Prophet Model compared to other models (Performance)***

4.5.4 Scatter plot comparing Prophet with other algorithms.

A screenshot of a computer

Description automatically generated

Fig. 18 Prophet Model compared to other models (Performance) using scatter plot.

From Fig. 17 and 18:

Mean Absolute Error (MAE):

Prophet MAE: 641.50

MAE Range: 139.80 (ARIMA) to 671.77 (NeuralProphet)

Prophet's MAE is approximately 78% higher than ARIMA, indicating that ARIMA has the lowest absolute prediction errors. However, Prophet outperforms NeuralProphet by approximately 4%.

Mean Squared Error (MSE):

Prophet MSE: 480,185.90

MSE Range: 33,089.28 (ARIMA) to 5,278,636.00 (NeuralProphet)

Prophet's MSE is approximately 93% lower than NeuralProphet, making it the superior model in terms of minimizing squared prediction errors. However, ARIMA still outperforms Prophet by approximately 93%.

Root Mean Squared Error (RMSE):

Prophet RMSE: 692.95

RMSE Range: 181.90 (ARIMA) to 1334.10 (GARCH)

Prophet's RMSE is approximately 52% 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 62% and 65%, respectively.

Mean Absolute Percentage Error (MAPE):

Prophet MAPE: 49.11%

MAPE Range: 11.44% (ARIMA) to 99.20% (GARCH)

Prophet's MAPE is approximately 328% higher than ARIMA, demonstrating that ARIMA has the lowest percentage prediction errors. However, Prophet significantly outperforms GARCH, which has the highest MAPE, by approximately 102%.

R-squared (R2) Score:

Prophet R2: -26.85%

R2 Range: -0.92% (ARIMA) to -102.22% (GARCH)

Prophet's R2 score is negative, 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 your(we) forecasting task. If minimizing absolute or percentage errors is crucial, ARIMA may be preferred. If you (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. 19 Index Comparison Iseq20 and other European indexes

Key Observations (fig.19):

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 screenshot of a white sheet with numbers

Description automatically generated**

Fig. 19 Basic Statistics European indexes

A screenshot of a chat

Description automatically generated

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 (fig 19 and 20):

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

Now, let' s import 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 to preprocess and have a comparable data.

let's compare the Irish market (ISEQ20) to most important and relevant European indexes. let's use date and close columns as it's a univariate analyse for time series and anomaly detection.

No missing values in all the datasets.(meter resultado)

4.7.3 Outliers

Let's use z-score, 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 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.

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

4.7.4 Heatmap for performance comparison

A screenshot of a graph

Description automatically generated

Fig. 21 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. 21).

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. - Stacked area chart for visualizing forecasts

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. 22 Differences in index Prices comparison using scatter plot (Irish Vs German Index)

4.7.5.2 Stacked Area Chart

A graph showing a green line

Description automatically generated

Fig. 23 Differences in index Prices comparison using Stacked Area Chart (Irish Vs German Index)

Conclusion from both Fig. 22 and Fig. 23

Index Values:

DAX 40: 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:

Volatility Comparison: 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.

Market Factors:

Economic factors, corporate performance, and geopolitical events in Germany and Ireland can influence the relative performance of these indices. Economic growth, stability, and investor sentiment are key drivers.

Investment Considerations:

Diversification: Investors interested in diversifying their portfolio across different European markets may consider the performance and characteristics of both indices.

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. Investors should consider these trends, market context, and their own investment goals when making informed decisions. Additionally, it's essential to conduct a more comprehensive analysis, including factors like historical performance, sector-specific trends, economic indicators, and external events, for a more complete investment strategy.

4.7.5.3 Cumulative returns comparison

**Cumulative Returns Comparison in Finance**

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.

**Calculating Cumulative Returns**

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*=*P*0​*Pt*​−*P*0​​×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.

Cumulative returns comparison offers several advantages in finance:

1. **Performance Assessment**: It allows investors to gauge the historical performance of different assets or investment strategies, helping them identify which options have generated higher returns over a specific period.
2. **Risk Evaluation**: Cumulative returns can also be used in conjunction with other risk metrics to assess the risk-adjusted performance of investments, considering both returns and volatility.
3. **Portfolio Optimization**: Investors can use cumulative returns to optimize their portfolios by selecting assets or strategies that align with their financial goals and risk tolerance.

A graph showing a green and blue line

Description automatically generated

Fig. 23 Differences in index Prices comparison using Stacked Area Chart (Irish Vs German Index)

*DAX 40 Cumulative Return:*

Starting Value: Not available

Ending Value: -0.168034 (as of the last data point)

Total Return: -0.168034

*ISEQ 20 Cumulative Return:*

Starting Value: Not available

Ending Value: -0.040685 (as of the last data point)

Total Return: -0.040685

*Conclusion from Fig. 23:*

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%.

Relative Decline: DAX 40 declined by approximately 4.13 times more than ISEQ 20. This indicates that investments in DAX 40 have been more adversely affected during the analysed period compared to investments in ISEQ 20.

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.

Investment Decision: If an investor had invested in DAX 40 and ISEQ 20 at the beginning of the observed period, they would have experienced a significant loss in both cases. However, the loss would have been more substantial with DAX 40.

Diversification: To mitigate risk, investors often diversify their portfolios across different assets or indexes. This analysis highlights the importance of diversification, as investments concentrated in a single index (DAX 40) could be more susceptible to market volatility.

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:

a. **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.

b. **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.

c. **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. 23 Timeline with anomaly detection from Iseq20

From Fig. 23:

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**

5.4.1 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.

**5.4.2** **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.

**5.4.3 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.

**5.4.4 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.

**5.4.5 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.

**5.5.2 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 you 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.

**5.5.3 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 characteristics comparison**

**A table with text on it

Description automatically generated**

Fig. 24 Anomaly detection models compared.

**5.7 Results and Discussion**

5.7.1 Comparing Anomaly Detection Methods

A screenshot of a computer screen

Description automatically generated

Fig. 25 Anomaly detection models compared with performance metrics.

A screen shot of a graph

Description automatically generated

Fig. 26 Anomaly detection models compared with performance metrics.

From Fig 25 and 26:

Prophet Model:

Precision: 0.562 (moderate)

Recall: 1.000 (perfect)

F1-Score: 0.719 (moderate)

AUC-ROC: 0.500 (low)

AUC-PR: 0.781 (moderate)

Key Takeaways for Prophet:

Recall: Prophet achieves a perfect recall (1.000), indicating that it effectively identifies all actual anomalies in the dataset. This suggests that it is excellent at capturing true positive cases without missing any.

Precision: The precision of Prophet is moderate (0.562), suggesting that while it has a good true positive rate, it might generate some false positives, leading to a trade-off between precision and recall.

F1-Score: The F1-Score, which balances precision and recall, is also moderate (0.719). This indicates that the model achieves a reasonable balance between correctly identifying anomalies and minimizing false alarms.

AUC-ROC: The AUC-ROC is 0.500, which is relatively low. This suggests 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, it does not perform well.

AUC-PR: The AUC-PR is moderate at 0.781. This metric considers the precision-recall trade-off, and the moderate value suggests that Prophet provides a reasonable balance between precision and recall.

*Comparison to Other Models:*

Isolation Forest: Isolation Forest has a very high precision (1.000) and a slightly lower recall (0.977). This indicates it has a higher precision than Prophet but a slightly lower recall. However, it lacks an AUC-ROC value, making it challenging to compare in terms of ROC performance.

One-Class SVM: One-Class SVM also has perfect precision (1.000) but a lower recall (0.562) than Prophet. It shares a similar precision-recall trade-off with Prophet. However, like Isolation Forest, it lacks an AUC-ROC value.

NeuralProphet: NeuralProphet achieves perfect precision, recall, and F1-Score, suggesting it excels in both precision and recall aspects. However, it lacks an AUC-ROC value, making it challenging to compare in terms of ROC performance.

LSTM: LSTM has a moderate precision (0.555) and a perfect recall (1.000), similar to Prophet. It has a slightly lower F1-Score (0.714) than Prophet. However, like Prophet, it has a low AUC-ROC (0.500) and a moderate AUC-PR (0.778).

Summary:

Prophet excels in recall, ensuring that it identifies all actual anomalies. This makes it suitable for tasks where missing any anomalies is critical.

However, Prophet has a moderate precision, indicating a trade-off between correctly identifying anomalies and generating some false positives.

In terms of ROC performance, both Prophet and LSTM have a low AUC-ROC, suggesting that they perform no better than random chance. (NeuralProphet?)

The choice between Prophet and other models depends on the specific requirements of the anomaly detection task. If recall is crucial and some false positives can be tolerated, Prophet may be a good choice. However, if you (we) need a higher precision or better ROC performance, you (we) may need to explore other models or further optimize Prophet's parameters.

**6. Integrated Analysis**

In this integrated analysis, we combine time series forecasting using the Prophet model with anomaly detection to provide a comprehensive understanding of a financial dataset, specifically focusing on stock prices (Iseq20). We aim to extract valuable insights for investors and risk management professionals by combining two essential aspects of data analysis.

Time Series Forecasting with Prophet Model

6.1 Forecasted Values

Using the Prophet model, we forecasted the Iseq20 stock prices from August 24, 2020, to August 22, 2023. The forecasted values started at 1142.59 and exhibited an upward trend, reaching approximately 1556.67 by August 22, 2023. This positive trend suggests a potential investment opportunity.

6.2 Confidence Intervals

We also computed upper and lower confidence intervals (CI) to account for uncertainty and potential price fluctuations. The upper CI values ranged from 1188.04 to around 1602.52, indicating a maximum potential stock price. Conversely, the lower CI values ranged from 1097.14 to 1513.04, suggesting the lowest potential price levels. These intervals provide a range within which stock prices are likely to fluctuate, aiding investors in risk assessment.

6.3 Trend Analysis

We analysed the trend in the Iseq20 index, identifying key periods of growth and decrease:

Overall Uptrend (01/07/2021 - 10/10/2022): Marked by consistent growth, peaking at 1576 points.

Post-COVID Recovery (03/11/2020 - 17/02/2021): Significant growth, signifying economic rebound.

Steady Growth (29/03/2021 - 06/05/2022): Stable market environment.

Market Correction (07/05/2022 - 21/06/2022): A correction phase, possibly due to inflation concerns.

Volatility (22/06/2022 - 10/10/2022): Increased market uncertainty.

Downturn (08/03/2022 - 15/03/2022): Sharp decline due to interest rates and geopolitical tensions.

Consolidation (16/03/2022 - 30/06/2022): Cautious market sentiment.

These trend analysis insights help investors understand the cyclical nature of financial markets and make informed decisions.

6.4 Seasonal Decomposition

We observed recurring seasonal patterns in the Iseq20 data, with fluctuations alternating between positive and negative values. These patterns signify the influence of regular, repeating factors on stock prices.

6.5 Residual Component

The residual component represents unexplained variability in the data. Notable positive and negative residual values indicate unexpected price movements beyond the trend and seasonal patterns.

Anomaly Detection

6.6 Anomalies Overview

We identified a total of 87 anomalies in the dataset, occurring irregularly. They are distributed across the years as follows: 2020 (40.23%), 2021 (27.59%), 2022 (26.44%), and 2023 (5.75%). These anomalies represent significant deviations from the norm and warrant careful analysis.

6.7 Anomaly Characteristics

Largest Anomaly: July 10, 2023, with a closing price of 1460.22.

Smallest Anomaly: October 28, 2020, with a closing price of 1082.61.

Average Anomaly: Approximately 1315.48.

Median Anomaly: Approximately 1309.57.

The anomalies exhibit substantial price deviations, averaging approximately 13.45% from the rolling 30-day mean closing price.

Comparing Anomaly Detection Methods

We evaluated the Prophet model's performance in anomaly detection and compared it to other models:

Recall: Prophet achieved a perfect recall (1.000), ensuring the identification of all actual anomalies.

Precision: Moderate precision (0.562) suggests a trade-off between true positives and false positives.

F1-Score: A moderate F1-Score (0.719) balances precision and recall.

AUC-ROC: A low AUC-ROC (0.500) indicates performance no better than random chance.

AUC-PR: A moderate AUC-PR (0.781) balances precision and recall.

Prophet's strengths lie in its high recall, making it suitable for tasks where missing anomalies is critical. However, precision and ROC performance are moderate.

As a result, this integrated analysis combines time series forecasting with anomaly detection to provide a holistic view of financial data. Investors and risk managers can use this information to make informed decisions, considering both the predicted trends and potential anomalies. The choice of model and approach depends on the specific requirements of the analysis, balancing factors like recall, precision, and ROC performance to meet the desired objectives.

**7. Conclusion**

This thesis has delved deep into the world of time series analysis and anomaly detection within the context of financial data. Through a meticulous examination of these domains, we've uncovered crucial insights that have significant implications for financial professionals, investors, and decision-makers operating in today's dynamic and unpredictable financial landscape.

In the realm of time series forecasting, the implementation of the Prophet model has provided us with a powerful tool for understanding and predicting the behaviour of financial indices, exemplified here by the Iseq20 index. The forecasts generated by the Prophet model, along with their associated confidence intervals, offer a detailed perspective on the potential trajectory of stock prices. Our analysis has revealed a compelling upward trend in the Iseq20 index, with forecasted values starting at 1142.59 and trending upwards to approximately 1556.67 by August 22, 2023. This finding suggests a substantial positive trend that can guide investment decisions and risk management strategies.

However, it's crucial to recognize that while Prophet excels in certain aspects, such as capturing trends and providing moderate accuracy, it may not be the optimal choice for all forecasting scenarios. Our comparative analysis has shown that alternative models like ARIMA and LSTM may outperform Prophet in specific metrics. For instance, in terms of Mean Absolute Error (MAE), Prophet achieves an MAE of 641.50, whereas ARIMA excels with an MAE of 139.80, indicating its superior performance in minimizing absolute prediction errors. The selection of the most suitable forecasting model should be driven by the specific goals and requirements of the forecasting task.

Transitioning to anomaly detection, this thesis has identified and thoroughly analysed 87 instances of anomalies within the financial dataset, spanning a three-year period. These anomalies serve as critical markers, shedding light on significant market events and deviations from expected behaviour. They provide actionable insights that can inform investment strategies, risk mitigation efforts, and regulatory actions. In the context of Prophet's performance in anomaly detection, it is noteworthy that the model has demonstrated a perfect recall, effectively identifying all actual anomalies in the dataset. This strength is particularly valuable when the cost of missing anomalies is high. However, it's essential to recognize that Prophet's precision has shown a trade-off, indicating a balance between correctly identifying anomalies and generating some false positives. Moreover, the model's AUC-ROC values suggest that it may not excel in distinguishing between anomalies and normal cases as effectively as desired.

As we conclude this thesis, it is evident that the financial landscape is characterized by complexity and constant change, influenced by an array of factors, including economic indicators, geopolitical events, and investor sentiment. The insights gained from this research empower financial professionals to make more informed decisions and navigate this intricate ecosystem with confidence. While the Prophet model has its strengths and limitations, it remains a valuable tool in the toolkit of financial analysts, offering valuable forecasts and anomaly detection capabilities.

In a world where financial markets are concerned about uncertainty and volatility, the knowledge and methodologies explored in this study provide a foundation for making well-informed decisions that can lead to more effective risk management and investment strategies.

**8. 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.

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.

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.

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.

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.

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.

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.

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.

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.