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

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.

Introduction:

**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**

**1.4 Thesis Organization**

**1.4.1 Framework of methodology**

Data Sources

Data Collection and Preprocessing

Data Description

Handling Missing Values and Outliers

Data Preprocessing

Introduction

Time Series Forecasting

ARIMA, SARIMA, GARCH, LSTM, Neural Prophet

**A screenshot of a graph

Description automatically generated**

Models comparison

Prophet Model

Training Parameters

Hyperparameter Tuning

Mean Absolute Error (MAE)

Mean Squared Error (MSE)

Root Mean Squared Error (RMSE)

Mean Absolute Percentage Error (MAPE)

R-squared (R2)

Performance Metrics

Anomaly Detection

Introduction

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

Models comparison

**A graph of anomaly detection

Description automatically generated**

Prophet Model

Precision, Recall and F1-score, ROC curve, AUC-ROC, Precision-Recall Curve (AUC-PR)

Performance Metrics

Integrated Analysis of best prophet library performances

1.4.2 Literature Review:

2.1 Time Series Forecasting Techniques

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

2.1.2 Time Series Decomposition and Forecasting Model (Prophet)

2.1.3 Deep learning model (LSTM, NeuralProphet,)

2.2 Anomaly Detection Approaches

2.2.1 Deep learning model (NeuralProphet, LSTM, Autoencoder)

2.2.2 Ensemble Anomaly Detection (Isolation Forest)

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

2.2.4 Time Series Decomposition and Forecasting Model (Prophet)

2.3 Related Studies in Finance Market Analysis

2.4 Overview of the Prophet Model

2.4.1 Traditional

2.4.2 Neural Prophet

**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 provide a reference point for evaluating the performance of individual assets or portfolios over time. By comparing the behavior of an asset or portfolio to the corresponding index, one can identify anomalies or deviations from the expected market trends.
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.

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 Euronext**

Euronext is a major stock exchange operator spanning several European cities, including Amsterdam, Brussels, Dublin, Lisbon, Milan, Oslo, and Paris. It ranks among Europe's largest stock exchange operators with benefits:

Regional Scope: Euronext offers a comprehensive view of European stock markets, making it a valuable resource for individuals interested in European stocks.

Market Data: It supplies real-time and historical market data, encompassing stock prices, indices, and trading volumes.

Company Insights: Users can access extensive details about listed companies, including financial statements, news, and corporate actions.

Trading Tools: Euronext provides tools and platforms for trading, which can be beneficial for those looking to execute trades on European markets.

Regulatory Information: It provides access to regulatory information and filings, aiding users in staying compliant with market rules.

News and Analysis: Euronext frequently publishes news and analytical insights pertaining to European financial markets, offering valuable information for decision-making.

**3.1.3 Yahoo Finance**

Yahoo Finance is a comprehensive financial platform and website operated by Yahoo. It delivers a wide range of financial data and tools to assist users in making informed investment choices with the following benefits:

Global Reach: Yahoo Finance covers not only U.S. markets but also global markets, spanning stocks, indices, currencies, commodities, and more.

Free Accessibility: Most of Yahoo Finance's offerings are available free of charge, including real-time stock quotes, news, and basic portfolio tracking.

Company Data: Users can access comprehensive information about publicly traded companies, including financial reports, analyst ratings, news updates, and historical data.

Interactive Charts: Yahoo Finance supplies interactive and customizable charts for technical analysis and visualizing historical performance.

Watchlists and Portfolios: Users can create watchlists and portfolios to monitor their investments and set up notifications for price changes and news.

Educational Resources: The platform offers educational content such as articles and videos to help users comprehend financial concepts and market trends.

**3.1.4 Sources**

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

Finance.Yahoo: allows to download several financial indexes’ datasets at <https://finance.yahoo.com/quote/RYAAY/history?p=RYAAY>,

Euronext: allows to download several financial indexes’ datasets in a time range ([https://live.euronext.com/markets/dublin/equities/list?\_gl=1\*1fnak24\*\_ga\_WYRYLMR662\*MTY5MDg5NzMzNi4xLjAuMTY5MDg5NzMzNi4wLjAuMA..\*\_ga\_PMEFBR6CSF\*MTY5MDg5NzMzNi4xLjAuMTY5MDg5NzMzNi4wLjAuMA](https://live.euronext.com/markets/dublin/equities/list?_gl=1*1fnak24*_ga_WYRYLMR662*MTY5MDg5NzMzNi4xLjAuMTY5MDg5NzMzNi4wLjAuMA..*_ga_PMEFBR6CSF*MTY5MDg5NzMzNi4xLjAuMTY5MDg5NzMzNi4wLjAuMA)..)

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 Data Description**

**3.2.1 Basic statistics**

**A close up of a number

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

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.

**3.2.2 Data distribution**

**A table with numbers and a few words

Description automatically generated with medium confidence**

**A graph with many blue squares

Description automatically generated with medium confidence**

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.

**3.3 Data Preprocessing Steps**

3.3.1 Data Cleaning

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 modeled 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 Hyperparameter Tuning**

**4.3 Performance Evaluation Metrics**

4.3.1 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,

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

4.3.2 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. 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

4.3.3 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

4.3.4 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

4.3.5 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 Results and Discussion**

**4.4.1 Comparing Prophet with Traditional Forecasting Methods**

**4.4.2 Analyzing Forecasting Accuracy and Robustness**

**4.5 Time Series Analysis (Code Section)**

4.5.1 Descriptive Analysis

- Data distribution

A graph of blue bars

Description automatically generated with medium confidence

Fig 1. Distribution of close prices Iseq20

From Fig.1, 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.

A green rectangular object with white lines

Description automatically generated with medium confidence

Fig 2. Box Plot Iseq 20

From Fig. 2:

Range of Prices: The 'Close' prices vary between a minimum of approximately 1063.24 and a maximum of around 1545.07. This suggests a considerable range of stock prices over the given time period.

Central Tendency: The median (Q2) of approximately 1342.58 is between the 1st quartile (Q1) and the 3rd quartile (Q3), indicating a roughly symmetrical distribution. This suggests that the central tendency of the 'Close' prices is around 1342.58.

Spread of Data: The interquartile range (IQR), which is the difference between Q3 and Q1, is approximately 218.67 (1451.91 - 1233.24). This indicates the spread of the middle 50% of the data. A larger IQR suggests a wider range of prices within this central range.

Outliers: It's worth noting that there may be outliers in the data as well. Outliers are data points that fall significantly below Q1 or above Q3 and are represented as individual points on the box plot. These outliers can potentially indicate extreme price fluctuations.

In summary, the box plot statistics provide valuable insights into the distribution and variability of the 'Close' prices. The data appears to have a somewhat symmetrical distribution with a central tendency around 1342.58, but there are also outliers suggesting occasional extreme price movements.

- Density Plot

A graph of a person with a graph

Description automatically generated with medium confidence

Fig.3 Density Plot

The density estimates provide insights into the distribution and likelihood of 'Close' prices within the dataset:

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.

4.5.2 Time Series Analysis

- Timeline plot

A graph showing a line

Description automatically generated with medium confidence

Fig. 4 Timeplot Iseq20

- Autocorrelation and partial autocorrelation

A graph of a function

Description automatically generated

Fig. 5. ACF

The ACF values decline as the lag between data points increases. 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. 6 PACF

PACF values exhibit a more intricate pattern compared to ACF. Several PACF values significantly differ from zero.

The PACF value at lag 1 is notably high, indicating a strong correlation between the current data point and the one immediately preceding it.

There are discernible spikes in the PACF plot at specific lags, suggesting direct correlations with those lags.

Conclusion from both plots:

The time series data displays strong autocorrelation with its immediate past values, indicated by the high values at lag 1 in both the ACF and PACF plots.

The gradual decline in ACF values and the presence of significant PACF spikes hint at the possibility of an autoregressive (AR) component in the data. In AR models, the current value is influenced by its past values.

**4.6 Modeling and Optimization (Code Section)**

4.6.1 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 computer screen shot of numbers and symbols

Description automatically generated

Table 1 – Results of Bayesian optimization Iseq20

Hyperparameters: The chosen hyperparameters for your model indicate the prior scale for changepoints and holidays. These values have been determined through model tuning or optimization.

Model Performance:

The MAE (Mean Absolute Error) of 227.66 suggests that, on average, your model's predictions are off by approximately 227.66 units from the actual values. Lower MAE values are desirable.

The MSE (Mean Squared Error) of 62,832.77 indicates the average squared difference between predictions and actual values. It's a larger value due to squaring, and lower values are preferred.

The RMSE (Root Mean Squared Error) of 250.66 is the square root of MSE and provides a measure of the model's prediction error in the original units. Smaller RMSE values are better.

The MAPE (Mean Absolute Percentage Error) of 17.36% indicates that, on average, your model's predictions have an error of 17.36% relative to the actual values. Lower MAPE values are desirable.

The R-squared (R2) of -2.64 suggests that the model is not performing well. R-squared should ideally be close to 1, indicating a good fit. A negative R-squared indicates that the model's predictions are worse than simply using the mean of the observed values.

Overall, the model's performance, as indicated by the provided metrics, suggests that it may not be a good fit for the data. Further model refinement or alternative modeling approaches may be necessary to improve prediction accuracy. Additionally, the negative R-squared suggests that the model is performing worse than a simple mean-based model, indicating the need for significant improvement.

4.6.2 Prophet Model Forecast with Confidence Intervals (plot)

A graph showing a line

Description automatically generated with medium confidence

Fig. 8 Iseq20 Prophet Model Forecast with Confidence Intervals

Original Data:

The original dataset is the ISEQ20 historical index from August 24, 2020, to August 22, 2023, with a use of 764 data points.

In this period, the index fluctuates from a low of 1132.07 to a high of 1461.70.

Forecasted Values:

The forecasted values extend beyond the original dataset, covering the period from August 24, 2020, to August 21, 2024, with a total of 1129 data points.

These forecasted values, indicated by the 'yhat' column, start at 1124.90 and show an upward trend, eventually reaching approximately 1900.27 by August 21, 2024.

This suggests that the model predicts an increase in stock prices over the forecasted period.

Upper Confidence Interval:

The upper confidence interval (CI) values, represented in the 'yhat\_upper' column, provide an upper limit for the forecasted stock prices.

Beginning at 1164.76, these upper CI values rise over time, indicating the highest potential stock price levels.

The upper CI offers insight into the possible upper boundary of stock price growth, suggesting that prices may reach as high as 1164.76 by August 24, 2020, and potentially higher in the forecasted period.

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 1083.71, these lower CI values increase over time, indicating the lowest potential stock price levels.

The lower CI helps define a range within which stock prices are likely to remain, suggesting that prices may not drop below 1083.71 by August 24, 2020, and potentially higher in the forecasted period.

Overall Conclusion:

In summary, the forecasts, along with upper and lower confidence intervals, provide valuable information regarding expected trends and the potential range of stock prices.

The forecasts point toward an upward trend in stock prices, while the confidence intervals acknowledge uncertainty and potential price variability.

Investors and stakeholders should consider these forecasts and confidence intervals when making investment decisions, weighing the upward trend against potential price fluctuations.

**4.7 Decomposition and Smoothing (Code Section)**

4.7.1 Trend analysis

A graph showing a line

Description automatically generated with medium confidence

Fig. 9 Iseq 20 trend

The trend component reveals the underlying, longer-term pattern in the data. Starting from August 27, 2020, there is a noticeable upward trend in the closing prices of the ISEQ20 index, with an increase of approximately 1142.93 points. Prior to this date, the trend component is marked as "NaN," indicating that there might not have been a clear trend during that period.

4.7.2 Seasonal decomposition

A blue and white barcode

Description automatically generated

Fig. 10 Iseq 20 Seasonal decomposition

The seasonal component 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.

4.7.3 Residual component

A graph showing a sound wave

Description automatically generated

Fig. 11 Iseq 20 Residual component

The residual component 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.

4.8 Performance Evaluation (Code Section)

A black text on a white background

Description automatically generated

Table 2 – Results of ISEQ20 Performance Metrics

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.

**4.9 Model Comparison (Code Section)**

**4.9.1 Models compared to Prophet Model**

4.9.1.1 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.

4.9.1.2 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.

4.9.1.3 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.

4.9.1.4 **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.

4.9.1.5 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)).

**4.9.2** **characteristics comparison**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aspect | ARIMA | SARIMA | LSTM | Neuralprophet | GARCH |
| Model Type | Statistical | Statistical | Neural Network | Hybrid (NN+TSF) | Statistical |
| Application Area | General time series | Seasonal time series | General time series | General time series | Financial volatility |
| Handling Missing Data | Requires imputation | Requires imputation | Can handle missing data | Can handle missing data | Requires complete data |
| Stationarity | Requires Differencing | Requires Differencing | Data Transformation | Handles Non-Stationarity | Not Applicable |
| Handling Seasonality | Manual specification | Automatic incorporation | Requires feature engineering | Automatic detection | Not applicable |
| Long-term Dependencies | Limited | Limited | Effective | Effective | Limited |
| Handling Irregular Data | Not suitable | Not suitable | Suitable | Suitable | Not suitable |
| Interpretability | High | High | Moderate | Moderate | Low |
| Computational Complexity | Low | Moderate | High | Moderate | Low |
| Parameter Tuning Required | Yes | Yes | Yes | Minimal | Yes |
| Forecast Horizon | Short to medium-term | Short to medium-term | Short to long-term | Short to medium-term | Short to medium-term |
| Strengths | - Simple to implement <br> - Interpretable <br> - Suitable for stationary data | - Captures seasonal patterns <br> - Handles seasonal data | - Captures complex patterns <br> - Suitable for long sequences | - Handles irregular data <br> - Automatically detects seasonality | - Models financial volatility |
| Weaknesses | - May require manual specification <br> - Limited handling of seasonality | - Requires knowledge of seasonal periods <br> - Limited handling of irregular data | - Complex architecture <br> - Computationally intensive | - May not work well with very short time series | - Limited application outside finance |
| Software Libraries (Examples) | statsmodels, forecast, auto\_arima | statsmodels, forecast | TensorFlow, PyTorch, Keras | NeuralProphet | rugarch, fGarch |

4.9.3 Prophet Model vs. Other Algorithms

A number and numbers on a white background

Description automatically generated

Table 3 – Results of Prophet Model VS other models

4.9.4 Scatter plot comparing Prophet with other algorithms.

A screenshot of a computer

Description automatically generated

Fig. 14 Performance Metrics Scatter plot

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 forecasting task. If minimizing absolute or percentage errors is crucial, ARIMA may be preferred. If you prioritize ease of use and interpretability, Prophet might be a suitable choice despite its slightly higher errors compared to ARIMA and LSTM.

**4.10 Index Comparison (Code Section)**

**A graph of a graph

Description automatically generated with medium confidence**

Fig. 15 Iseq20 vs other European Indexes

Key Observations:

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.

4.10.1 Basic Statistics european indexes

**A screenshot of a white sheet with numbers

Description automatically generated**

Table 1 - Summary statistics

Analyzing 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:

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).

4.10.2 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.

3.3.2 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.

4.10.2 Scatter plot comparing Iseq20 vs. Other Indexes.

- Plot with Normalized data for comparison

A graph of a stock market

Description automatically generated with medium confidence

Fig. 16 Normalized Close Prices

- Heatmap for performance comparison

A screenshot of a graph

Description automatically generated

Fig. 17 Heatmap for performance comparison

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.

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

4.10.3 Irish Vs German Index

4.10.3.1 Scatter Plot

A graph showing a number of dots

Description automatically generated with medium confidence

Fig. 18 Irish Vs German Index

4.10.3.2

A graph showing a green line

Description automatically generated

Fig 19. Irish Vs German Index

Conclusion from both Fig. 18 and Fig. 19

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.

- Daily returns comparison

A graph showing a number of data

Description automatically generated with medium confidence

Fig. 20 Daily returns comparison (german vs Irish indexes)

Closing Prices: The DAX index consistently maintains higher closing prices compared to the ISEQ20 index. This implies that, on average, the German stock market (DAX) features stocks with higher values than the Irish stock market (ISEQ20). Daily Returns:

DAX Performance: Daily returns for the DAX index fluctuate across the timeframe. Positive returns signal days when the DAX index increased in value, while negative returns denote days of decline. Overall, the DAX portrays a mix of positive and negative daily returns, indicating a combination of upward and downward market movements. ISEQ20 Performance: The ISEQ20 index also displays fluctuations in daily returns, mirroring the pattern observed in the DAX. Similar to the DAX, the ISEQ20 showcases both positive and negative daily returns over the studied period.

Comparative Analysis: Volatility: Both indices demonstrate volatility, evident from the oscillations in daily returns. Investors might note that market volatility can present opportunities for profit while simultaneously carrying heightened risk. Correlation: To delve into the relationship between these indices, one can conduct a correlation analysis on their daily returns. A positive correlation indicates that when one index rises, the other often follows suit, whereas a negative correlation suggests the opposite behavior.

Color Indicator: The "Color" column signifies whether the daily return for each index is positive (labeled as "Positive") or negative (labeled as "Negative"). This feature aids in swiftly identifying the direction of daily movements. Recent Performance:

As of August 22, 2023, the DAX index rests at 15,705.62, while the ISEQ20 index stands at 1,461.70. This mirrors the historical trend where the DAX consistently upholds a higher value than the ISEQ20. In summary, the DAX index persistently outperforms the ISEQ20 index in terms of closing prices. Both indices showcase volatility and fluctuating daily returns, rendering them susceptible to market dynamics and external influences. To make well-informed investment decisions, investors should weigh factors such as their risk tolerance, investment timeline, and portfolio diversification. Scrutinizing the correlation between these indices and monitoring their performance in the context of broader economic and geopolitical events can yield valuable insights for constructing effective investment strategies.

- Cumulative returns comparison

A graph showing the stock market

Description automatically generated with medium confidence

Fig. 21 Cumulative returns comparison

**Anomaly Detection:**

**5.1 Introduction to Anomaly Detection Methods**

**A graph with red and blue lines

Description automatically generated**

Fig. 22 Iseq 20 Closing Prices with Anomalies

**A graph showing the growth of a stock market

Description automatically generated**

Fig 23 actual vs predicted

A graph with a line going up

Description automatically generated

Fig. 24 Residuals

**5.2 Implementing Anomaly Detection Models**

5.2.1 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.

5.2.2 **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.

* + 1. **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.

**5.2.4 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:** Similar to 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 behaviors compared to the original Prophet model.

**5.2.5 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 analyzed.

5.3 Performance Evaluation Metrics for Anomaly Detection

**5.3.1 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.

**5.3.2 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.

**5.3.3 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.

**5.4 characteristics comparison**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Isolation Forest | One-Class SVM | Prophet | NeuralProphet | LSTM |
| Model Type | Ensemble | Support Vector | Statistical | Hybrid | Deep Learning |
| Suitable for Time Series | Yes | Yes | Yes | Yes | Yes |
| Handling Seasonality | No | No | Yes | Automatic | Depends on Implementation |
| Handling Outliers | Yes | Yes | Robust to outliers | Robust to outliers | May require preprocessing |
| Handling Missing Data | Yes | No | Robust to missing data | Robust to missing data | May require preprocessing |
| Computational Complexity | Moderate | Moderate | Low | Moderate | High |
| Adaptability to Data Types | Versatile | Versatile | Time Series | Time Series | Time Series |
| Anomaly Explanation | No | No | No | No | No |
| Suitable for Univariate Data | Yes | Yes | Yes | Yes | Yes |
| Suitable for Multivariate Data | Yes | Yes | No | No | Yes |
| Handling Non-Stationarity | Yes | Yes | Limited | Limited | Yes |
| Sensitivity to Outliers | Moderate | High | Limited | Limited | High |
| Scalability | High | Moderate | Moderate | Moderate | Moderate |
| Interpretability | Limited | Limited | Yes | Limited | Limited |
| Handling Missing Data | Yes | No | No | Yes | No |
| Performance | Fast and efficient | Fast | Moderate | Moderate | Computationally Intensive |

**5.5 Results and Discussion**

5.5.1 Comparing Anomaly Detection Methods

A screenshot of a computer screen

Description automatically generated

Table 4 – Results of Model Comparison in Anomaly detection

5.5.2 Analysing the Prophet Model's Anomaly Detection Capability vs Other models

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.

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

A blue rectangles on a white background

Description automatically generated

Fig. 26 Model comparison anomaly detection

A screen shot of a graph

Description automatically generated

Fig. 27 Model comparison anomaly detection

Integrated Analysis:

**6.1 Combining Time Series Forecasting and Anomaly Detection**

**6.2 Identifying Anomalous Market Behavior in Forecasted Data**

**6.3 Decision-Making Implications for Investors and Financial Institutions**

**Conclusion:**

**7.1 Summary of Findings**

**7.2 Contributions and Implications**

**7.3 Limitations and Future Directions**

**References:**

Time Series Forecasting Using FB-Prophet by Kirti Sharma, , Rajni Bhalla ,and Geetha Ganesan (August 2022)

Integrating Navier-Stokes equation and neoteric iForest-BorutaShap-Facebook’s prophet framework for stock market prediction: An application in Indian context by Indranil Ghosh, Tamal Datta Chaudhuri (2022)

Vishwakarma, A., Singh, A., Mahadik, A., & Pradhan, R. (2020). Stock Price Prediction Using Sarima and Prophet. International Journal of Advanced Research in Science, Communication and Technology, 9(1). Retrieved from <http://ijarsct.co.in/Paper315.pdf>

Appendices:

**9.1 Detailed Model Configurations**

9.2 Additional Figures and Tables