**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.

**1. 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

**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

3.2 Data Description

3.3 Data Preprocessing Steps

3.4 Handling Missing Values and Outliers

**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)):

1. 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.
2. 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.
3. 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.
4. 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)):

1. 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.
2. 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.
3. 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.
4. 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.
5. 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)

4.3.2 Mean squared error (MSE)

4.3.3 Root Mean Squared Error (RMSE)

4.3.4 Mean Absolute Percentage Error (MAPE)

4.3.5 R-squared (R2)

4.4 Results and Discussion

4.4.1 Comparing Prophet with Traditional Forecasting Methods

4.4.2 Analysing Forecasting Accuracy and Robustness

**5. Anomaly Detection:**

5.1 Introduction to Anomaly Detection Methods

5.2 Implementing Anomaly Detection Models

5.2.1 Isolation Forest for Anomaly Detection

5.2.2 One-Class SVM for Anomaly Detection

5.3 Performance Evaluation Metrics for Anomaly Detection

5.3.1 Precision and Recall and F1-score

5.3.4 Receiver Operating Characteristic (ROC) Curve

5.3.5 Area Under the ROC Curve (AUC-ROC)

5.3.6 Precision-Recall Curve (AUC-PR)

5.4 Results and Discussion

5.4.1 Comparing Anomaly Detection Methods

5.4.2 Analysing the Prophet Model's Anomaly Detection Capability

**6. Integrated Analysis:**

6.1 Combining Time Series Forecasting and Anomaly Detection

6.2 Identifying Anomalous Market Behaviour in Forecasted Data

6.3 Decision-Making Implications for Investors and Financial Institutions

**7. Conclusion:**

7.1 Summary of Findings

7.2 Contributions and Implications

7.3 Limitations and Future Directions

**8. 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>

Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. THE AMERICAN STATISTICIAN, 72(1), 37-45. doi:10.1080/00031305.2017.1380080

**9. Appendices:**

9.1 Detailed Model Configurations

9.2 Additional Figures and Tables