Contents

1 Abstract

The identification and prediction of structural breaks or anomalies in time series data have gained significant importance in various fields. While several methods, including the Chow test, Cumulative Sum (CUSUM) test, and Bayesian Change Point Analysis among others, exist for this purpose, they have drawbacks such as being highly sensitive to structural break specification, requiring the user to specify suspected break points and sometimes being computationally intensive. The aim is to introduce a non-parametric approach that does not require prior parameters for structural break prediction.

With this research, there is a proposal for a unique method (with a variant) based on the two-sample Kolmogorov-Smirnov test to detect and predict structural breaks in time series data. The approach uses sub-samples of features from the dataset to conduct the Kolmogorov-Smirnov test and then forecasting is done on stock data with a simple Vector Autoregression as well as an LSTM model to compare differences. By utilizing these non-parametric methods, the aim is to open up new possibilities for enhanced decision-making and forecasting in areas where time series analysis plays a crucial role.

2 Introduction

2.1 Time Series

Time series are a fundamental part of research, offering valuable insights into the dynamics and patterns of data collected over the course of a time period. It provides a framework for analyzing and modeling data points that are observed at discrete time intervals. Time series data often exhibits time based dependencies, which means that the value of a data point at a given time is influenced by its past values. Understanding and leveraging these dependencies is crucial for accurate forecasting and identifying structural breaks, which are significant changes in the underlying process of generating the data.

Univariate time series focuses on a single variable observed over time, aiming to capture its inherent patterns, trends, and fluctuations. Various statistical techniques, such as autoregressive integrated moving average (ARIMA) models and exponential smoothing methods, are commonly employed to model univariate time series data. These models consider the previous values of the variable and incorporate them into a forecast, allowing for the prediction of future values.

In contrast, multivariate time series deals with multiple variables observed over time, exploring the relationships and dependencies among them. It allows for a more comprehensive understanding of the underlying dynamics by considering the simultaneous interactions between variables. Vector autoregressive (VAR) models, state space models and LSTM models are commonly used to analyze multivariate time series data. These models enable the estimation of multivariate relationships and allow for forecasting of future values based on the inter-dependencies identified.

Accurate forecasting relies on the assumption that the underlying process generating the time series data remains stable over time. However, structural breaks can significantly impact the dynamics of the data and invalidate this assumption, this will be further explained.

Whether analyzing univariate or multivariate time series, various statistical techniques can be employed to model and forecast future values. However, it is essential to account for the possibility of structural breaks, as they can substantially alter the dynamics of the data and impact the accuracy of forecasts. By considering both the inherent patterns and potential structural breaks, time series analysis can contribute valuable insights to econometric research and decision-making processes.

2.2 Structural Breaks

Structural breaks, also referred to as regime shifts or anomalies, are critical events that lead to a significant alteration in the underlying structure of a time series dataset. These breaks often result from changes in economic policies, technological advancements, shifts in market dynamics, or other exogenous factors that impact the data-generating process. Understanding and detecting structural breaks is essential for accurate forecasting, and decision-making in various fields.

This shift introduces a new regime with different parameters and properties, potentially leading to changes in the statistical properties of the time series data. Detecting structural breaks can be challenging due to their non-uniform patterns and diverse underlying causes.

Understanding the timing and impact of structural breaks is important for developing accurate forecasting models. Ignoring structural breaks can lead to misleading predictions, or inaccurate assessments of performance. By accounting for structural breaks, researchers can enhance the reliability and robustness of their analyses.

2.3 Relevance of the Research

Structural break analysis in time series data has gotten significant attention in the field of finance and has emerged as an important tool for decision-making. The detection and prediction of structural breaks in financial time series are of utmost importance for investors, and policymakers, as they can signify changes in market dynamics, shifts in underlying relationships, and potential investment opportunities.

Traditional methods have been extensively used in the financial domain to identify structural breaks. However, these methods often assume specific distributional assumptions and can be computationally intensive while may still fail to capture certain relationships inherent within the data. As a result, the need for alternative methods becomes evident.

Here, a new approach (with a variant) for structural break prediction in time series analysis is introduced. The method builds upon the two-sample Kolmogorov-Smirnov (KS) test, a non-parametric statistical test that enables the identification of differences in underlying distributions. By adapting this test to analyze sub-samples of features from datasets, the aim is to capture and predict structural breaks fairly accurately.

The relevance of the proposed method lies in its ability to provide valuable insights into the dynamics of the dataset by testing on the distribution itself without any need of prior parameters and being computationally simpler. By detecting structural breaks, different strategies and decisions can be employed. Moreover, the information gained from structural break prediction can be used to assess the effectiveness of taken measures and design appropriate responses in accordance to changing conditions.

By leveraging the potential of this method in structural break analysis, this research contributes to the advancement of structural break prediction. The insights gained from the proposed methods can possibly provide a deeper understanding of the underlying dynamics of financial markets.

In conclusion, the application of the two-sample KS test for structural break prediction in time series analysis offers a new avenue for research and practical applications.

3 Literature Review

3.1 Structural Break Detection

The detection of structural breaks in time series data has garnered significant attention in recent years around the world. Structural breaks, also known as anomalies, refer to abrupt changes in the underlying structure of a time series, which can arise from shifts in economic policies, changes in market dynamics, or other exogenous factors. Accurate identification and analysis of structural breaks are essential for reliable forecasting, and decision-making. This section aims to provide an overview of the existing research on structural break detection techniques.

Firstly there is the Chow test, proposed by Gregory Chow in 1960, is a widely used method for detecting structural breaks in regression analysis. It involves estimating separate regression models for different sub-periods and comparing the model fit statistics to assess if there is a significant change in the relationship between the variables. The test evaluates whether the coefficients of the regression models differ significantly before and after the suspected break point. If the test statistic exceeds a critical value, it indicates evidence of a structural break. The Chow test is particularly useful in understanding changes in the relationship between dependent and independent variables over time. However the Chow test requires the user to predefine a specific structural break point meaning it isn't suitable for detecting structural breaks at unknown/multiple points in time series. This also means that the test is highly sensitive to the choice of break point making it difficult to identify true structural breaks. The test statistic is given by:

$$F = \frac{(RSS_R - RSS_U)/m}{RSS_U/(T-2k)} \tag{1}$$

where RSS_R and RSS_U are the residual sum of squares for the restricted and unrestricted models, respectively, m is the number of restrictions imposed, T is the total number of observations, and k is the number of estimated parameters.

Next is the Cumulative Sum (CUSUM) test, introduced by Brown et al. in 1975, it is another widely employed method for detecting structural breaks. It involves cumulatively summing the differences between the observed values and the expected values under a given model. A significant upward or downward trend in the CUSUM statistic suggests the occurrence of a structural break. The CUSUM test is useful when the timing of the break is unknown and can detect both single and multiple breaks in a time series. The CUSUM test suffers from the same issue as the Chow test with regards to sensitivity of choice of break point, not only that the test is primarily designed to detect a single structural break. The CUSUM statistic is computed as follows:

$$S_t = \max\left(0, \sum_{i=1}^{t}(x_i - \hat{x}_i)\right) \tag{2}$$

where x_i is the observed value at time i and $\hat{x}_i$ is the expected value at time i based on the estimated model.

Next is the Bai-Perron test, developed by Bai and Perron in 1998, is a method for detecting multiple structural breaks in time series data. It uses a sequential testing procedure based on likelihood ratio tests to determine the optimal number and timing of breaks. The test involves estimating regression models with different break points and comparing the likelihood ratios to identify significant structural breaks. By iteratively adding breaks, the Bai-Perron test captures the changes in the time series more accurately than other tests that assume a fixed number of breaks. The issue regarding this test is that it is designed to detect structural breaks that involve change in mean of time series, as well as being dependent on number of breaks specified by user, if number specified is too low the test will fail to identify all breaks. The test statistic is computed as follows:

$$\lambda_k = \frac{T \cdot \hat{S}_k}{1 + \frac{2}{T} \sum_{j=1}^{k} \log(1 - \hat{\alpha}_j)} \tag{3}$$

where T is the total number of observations, $\hat{S}_k$ is the sum of squared residuals for the k-break model, k represents the number of breaks, and $\hat{\alpha}_j$ is the estimated break date.

Finally there is the Bayesian change point analysis, it is a statistical framework that utilizes Bayesian methods to detect structural breaks in time series data. It allows for the incorporation of prior information and uncertainty in the analysis. This approach involves specifying prior distributions for the break points and using Markov Chain Monte Carlo (MCMC) methods to estimate the posterior distribution of the break points. By sampling from the posterior distribution, one can identify the most likely break points and quantify the uncertainty associated with them.

However, the method faces the same issues as with Chow test and CUSUM test as well as being computationally intensive.

The Chow test, CUSUM test, Bai-Perron test, and Bayesian change point analysis are among the prominent approaches employed in the field. These techniques facilitate the identification of structural breaks in the underlying structure of time series data, enabling researcher to better understand and model complex time related phenomena.

Finally regarding the proposed method, the two-sample KS test is a non-parametric statistical test that can compare two samples to determine whether they come from the same distribution or not. By applying this test to the time series data, it can be used to identify structural breaks by detecting significant differences between distributions of two consecutive segments of the data.

Let $X_1, X_2, ..., X_n$ be time series data, where n is the total number of observations. To detect structural breaks, we divide the data into two segments: $X_{1:m}$ and $X_{m+1:n}$, where $1 \leq m < n$. Then the two-sample KS test is applied to these two segments to test the null hypothesis H_0 that they come from the same distribution.

The two-sample KS test statistic is given by:

$$D_m = \sup_x |F_{1,m}(x) - F_{2,n-m}(x)| \tag{4}$$

where $F_{1,m}(x)$ and $F_{2,n-m}(x)$ are the empirical distribution functions of $X_{1:m}$ and $X_{m+1:n}$, respectively. The empirical distribution function, denoted as $F_n(x)$, is a step function that gives the proportion of observations in a sample that are less than or equal to a specified value x. The formula for the empirical distribution function of a sample $X_1, X_2, \ldots, X_n$ is:

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x) \tag{5}$$

where $I(X_i \leq x)$ is an indicator function that equals 1 if $X_i \leq x$ and 0 otherwise.

In the context of the two-sample KS test, $F_{1,m}(x)$ and $F_{2,n-m}(x)$ are the empirical distribution functions of the first and second samples, respectively, up to a certain point m in the data for the first sample, and from the point $m+1$ to the end of the data for the second sample. The formula for the empirical distribution function of the first sample up to point m is:

$$F_{1,m}(x) = \frac{1}{m}\sum_{i=1}^{m} I(X_i \leq x) \tag{6}$$

And the formula for the empirical distribution function of the second sample from point $m+1$ to the end of the data is:

$$F_{2,n-m}(x) = \frac{1}{n-m}\sum_{i=m+1}^{n} I(X_i \leq x) \tag{7}$$

Under the null hypothesis H_0, D_m follows a distribution that depends on the sample size m and $n-m$. Further, the calculation of the p-value of the test as the probability of observing a test statistic as extreme as D_m or more extreme, under H_0 can be done. If the p-value is below a pre-determined significance level, the null hypothesis is rejected with a False value and it can be concluded that there is a structural break between the two segments.

By repeating this procedure for all possible values of m between 1 and $n-1$, the location of a difference in distribution within the time series data can be identified.

This in turn can be used to show the locations of structural breaks, through the use of a classifier and testing between sliding windows of data

3.2 Multivariate Time Series Forecasting

Multivariate time series forecasting plays an important role in a wide range of fields. Unlike univariate time series forecasting, where only a single variable is considered, multivariate time series forecasting involves predicting the future values of a variable using the other variables present within the time period as well. This section aims to provide an overview of the existing research and methodologies employed in the field of multivariate time series forecasting.

The Vector Autoregressive (VAR) model is one of the most commonly used and basic methods for multivariate time series forecasting. VAR models capture the interdependencies among the variables by representing each variable as a linear combination of its own past values and lagged values of other variables in the system. The parameters of the VAR model can be estimated using various techniques, such as ordinary least squares (OLS) or maximum likelihood estimation (MLE).

The VAR(p) model represents a multivariate time series as a linear combination of its own past values and lagged values of other variables in the system.

$$X_t = c + A_1 X_{t-1} + A_2 X_{t-2} + ... + A_p X_{t-p} + e_t \tag{8}$$

where X_t is a vector of multivariate time series variables at time t, c is a constant term, $A_1, A_2, ..., A_p$ are coefficient matrices capturing the lagged effects, and e_t is a vector of error terms.

Long Short-Term Memory (LSTM) models have gained significant popularity in the field of multivariate time series forecasting. LSTM is a type of recurrent neural network (RNN) that can effectively capture long-term dependencies and patterns in sequential data. Unlike the VAR model, LSTM models excel at handling the complex temporal relationships present in multivariate time series data. LSTM models are well-suited for this task as they can automatically learn and incorporate these dependencies in their predictions.

At the core of an LSTM model are memory cells that store and propagate information over time. These memory cells are designed to handle the vanishing or exploding gradient problem, which can hinder the learning process in deep neural networks. By utilizing gates, such as input, forget, and output gates, LSTM models selectively remember or forget information from previous time steps, enabling them to capture long-term dependencies effectively. This makes LSTM models particularly valuable when dealing with high-dimensional and complex data.

To train an LSTM model, historical data is divided into input sequences and corresponding target sequences where each observation consists of the values of multiple variables at a particular time step. The LSTM model is then trained on these sequential inputs and processes it, updating its memory cells and making predictions at each time step to predict the future values of the target variables given the historical input. During training, the model's parameters are optimized using gradient-based methods, such as back-propagation through time, which iteratively adjust the weights to minimize the difference between predicted and actual values.

Evaluating the accuracy and performance of multivariate time series forecasting models is important. Various evaluation metrics have been employed, including mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and mean absolute percentage error (MAPE). These metrics help assess the accuracy, bias, and efficiency of the forecasted values compared to the actual observations.

Mean Absolute Error (MAE):

$$MAE = \frac{1}{n} * \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{9}$$

Mean Squared Error (MSE):

$$MSE = \frac{1}{n} * \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{10}$$

Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{(}MSE) \tag{11}$$

Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100}{n} * \sum_{i=1}^{n} |\frac{y_i - \hat{y}_i}{y_i}| \tag{12}$$

where y_i represents the actual observed values, $\hat{y}_i$ represents the forecasted values, and n is the total number of observations.

Despite the progress made in multivariate time series forecasting, several challenges remain. Some of the key challenges include handling high-dimensional data, capturing long-term dependencies, addressing non-stationarity and structural breaks, and incorporating exogenous variables effectively. Future research efforts could focus on developing methodologies that address these challenges, exploring different approaches that combine different modeling techniques, and investigating interpretability and uncertainty quantification in multivariate forecasting models.

4 Methodology

4.1 Problem Statement

Despite the vast array of existing methodologies for detecting structural breaks, current approaches often suffer from several limitations. Many established techniques rely on restrictive assumptions, such as specific distributional assumptions or being highly sensitive to structural break specification, i.e. relying on previously specified break points from the user themselves, which may not be suitable for capturing the complexity of real-world data.

Therefore, there is a need for an approach that addresses these limitations and provides a framework for predicting structural breaks in time series data. This proposed method should overcome the shortcomings of existing techniques by incorporating the two sample Kolmogorov Smirnov test. Moreover, it should exhibit the capability to leverage the power of cross-sectional data.

Additionally, the new methodology should strive to enhance accuracy by incorporating adaptive modeling techniques that account for changing dynamics and evolving conditions in the time series data. By detecting and signaling upcoming structural breaks in a timely manner.

Therefore, the goal is to develop and validate a method for predicting structural breaks in time series data that aims to surpass the limitations of existing techniques. Through empirical evaluations and comparisons, this research aims to establish the effectiveness and practicality of the proposed approach as well as provide valuable insights and tools for various domains, enabling users to make informed decisions and adapt to changing conditions.

4.2 Objectives

- Choose two or more datasets to work and experiment upon
- Create a data streamer class to take a dataset and iterate through the dataset giving blocks of a specified size
- Perform the two sample Kolmogorov Smirnov column-wise on consecutive blocks
- Extract features of each column each block and its consecutive block
- Create new datasets with extracted features of both blocks with a new column containing test result
- Use datasets to create classification models for structural break prediction
- Forecast with Vector Autoregressive (VAR) model
- Forecast with Long short-term memory (LSTM) model
- Experiment, compare and contrast

4.3 Proposed Architecture

Before any structural break prediction or forecasting can be done, it is first necessary to be able to effectively feed data to the proposed model in order to get required results, For that a data streamer class is to be created, which takes any given dataset and allows the user to call upon blocks of data based on the user's specified size, as well as some other utility functions. These functions are ones that checks if it's still possible to take a block of same size from dataset, resetting the streamer back to the start of the dataset, returning the full stream of data, returning the block size, and finally returning the dataset size.

Next what would be required to be made is a preprocessing function for any block called or used by the user, this function works column-wise on each block of data passed through it. In order to prepare the block for the two sample Kolmogorov Smirnov test, any Date and/or Time columns would have to be temporarily dropped, any missing values would be replaced with the previous row value and if the missing value is at the beginning of the block, zero would be used instead. Finally, each column feature is separated via key value pairs in dictionaries in order to be later passed through to the test function and the dictionary is returned.

After that a function to the actual test is required, this function takes two dictionaries (returned by the preprocessing function above) from one block and the next consecutive block. The function then iterates through keys of both dictionaries (the keys are the same as they have the same column name) and performs the two sample Kolmogorov Smirnov test on each key's values using a significance value of 0.01 while storing the resulting Boolean values within another dictionary and returning it.

In order to make the classification (explained further on) training computationally easier, a statistics extractor function is created. In this function, statistics such as Mean, Median, Mode, Standard Deviation and Kurtosis are calculated for both dictionaries passed to it (both features from both blocks of data) and returned as a dataframe.

Now it is possible to create the training sets to train classification models to predict structural breaks for the user. Using a loop, empty dataframes are created for each feature in the dataset in order to store consecutive feature statistics. Then using a loop to iterate through the data streamer, two blocks are taken, passed through the preprocessing function, then the test function. Now the test results can be added as a new column to each statistics dataframe with each row value corresponding to two blocks of the feature. Then the initial block is replaced with the next block and the next consecutive block is called to replace the next block and is repeated till the end of the dataset.

With all dataframes for each feature at hand, classifiers can be built for predicting structural breaks for all features. For this either the Random Forest Classifier is used or One-Class SVM depending on a user specified threshold of presence for the presence of structural break class. Then classifiers are trained on newly created dataframes for all features from the original dataset and stored in a dictionary corresponding to each feature.

Finally forecasting can be done, for the sake of seeing multiple effects of structural break on forecasting, multivariate forecasting is done. As a benchmark, the Vector Autoregressive (VAR) and LSTM models can be used on the original dataset and resulting metrics for evaluation are recorded. Then using the newly proposed approach, prediction of presence of structural break on each time period via testing of consecutive rolling windows of specified block size, this essentially means that if the classifier were to use, for example: with a block size of 100, [0:99] and [1:100] rows and the classifier returns False (null hypothesis is rejected $\implies$ change in distribution), then there is only one term which causes the structural break that being the 100^{th}. From this, structural breaks starting from the row after the block size for each are predicted until the end and added as new columns, while those from zeroth index until the block size index are assumed to have no structural break. With this newly created dataset, the Vector Autoregressive (VAR) and LSTM models can be again used to forecast and record metrics.

A variant is also created where instead of adding all structural break predictions as separate columns, the majority boolean value of each row can be taken and made into a single separate column and added to the dataset and finally be used to forecast upon the the Vector Autoregressive (VAR) and LSTM models.

5 Experiments and Results

5.1 Datasets

The prediction and forecasting of structural breaks are performed using financial time series data from three prominent companies in the technology and entertainment sectors: Netflix (NFLX), Amazon (AMZN), and Alphabet Inc. (GOOGL), the parent company of Google. These companies were chosen due to their significance in their industries and the availability of high-quality historical data. The selection of these specific datasets provides a diverse representation of different sectors, allowing for a comprehensive analysis of structural break prediction and forecasting.

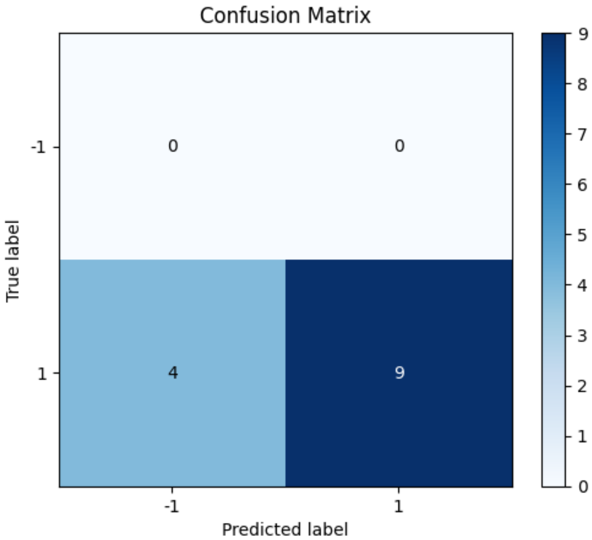
Netflix, an American media services provider, is widely recognized for its streaming platform offering a vast collection of movies, TV shows, and original content. The NFLX dataset comprises of daily Open, High, Low, Close, Adjusted Close prices and Volume traded, spanning from the 23^{rd} of May 2002 up until 1^{st} of April 2020. The availability of comprehensive financial information makes the NFLX dataset valuable for studying structural breaks and their impact on stock prices in the entertainment industry.

Google, a global technology corporation, has a strong presence in various sectors, including software development, cloud computing, and hardware devices. The GOOGL dataset also comprises of daily Open, High, Low, Close, Adjusted Close prices and Volume traded, spanning from the 13^{th} of March 1986 up until 1^{st} of April 2020, reflecting on Google's market performance over a significant time period. This dataset provides insights into the behavior of a leading technology company and its susceptibility to structural breaks. Analyzing the GOOGL dataset contributes to understanding the impact of economic events and industry trends on the stock prices of major technology firms.

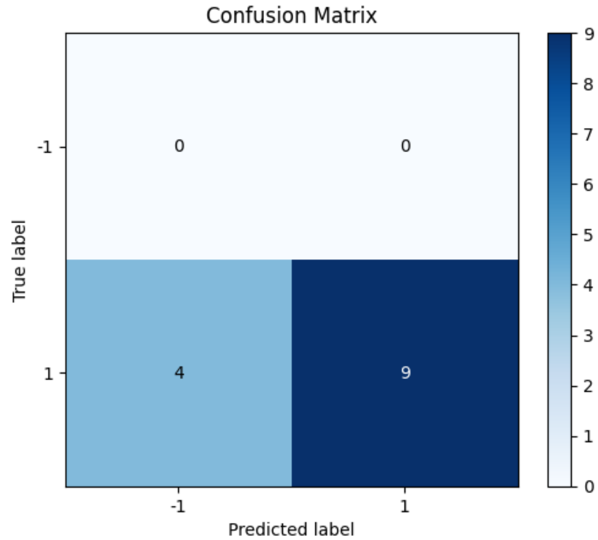
Amazon Inc, is a multinational conglomerate recognized for its dominance in the e-commerce and digital-streaming industry. The AMZN dataset encompasses daily Open, High, Low, Close, Adjusted Close prices and Volume traded, spanning from the 15^{th} of May 1997 up until 1^{st} of April 2020. The dataset reflects the influence of factors such as market competition, regulatory changes, and technological advancements on the stock prices of a prominent technology-based giant. The AMZN dataset enables the examination of structural breaks in a technology-driven sector, possibly shedding light on the relationship between market dynamics and stock price fluctuations.

5.2 Evaluation of Classification Models

Considering the classification model, one can visualise the confusion matrices of each feature from the three datasets in order to evaluate them.

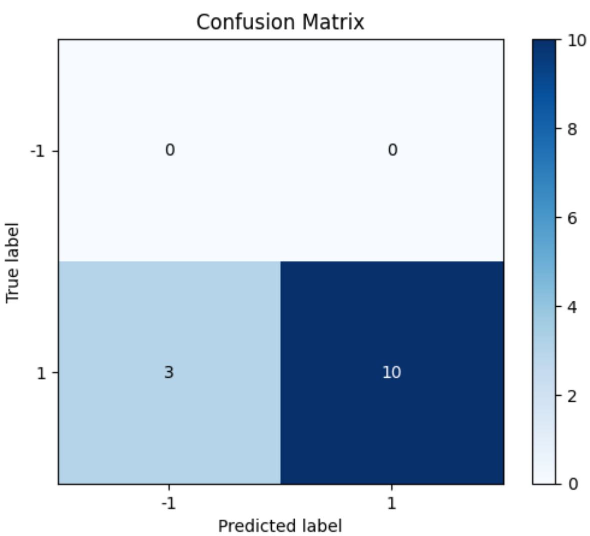


(a) NFLX Open Price

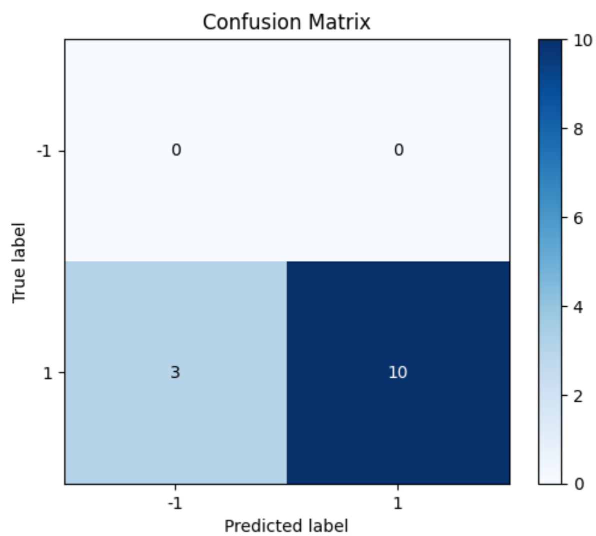


(b) NFLX High Price

Figure 1: NFLX's Open and High prices Confusion Matrices

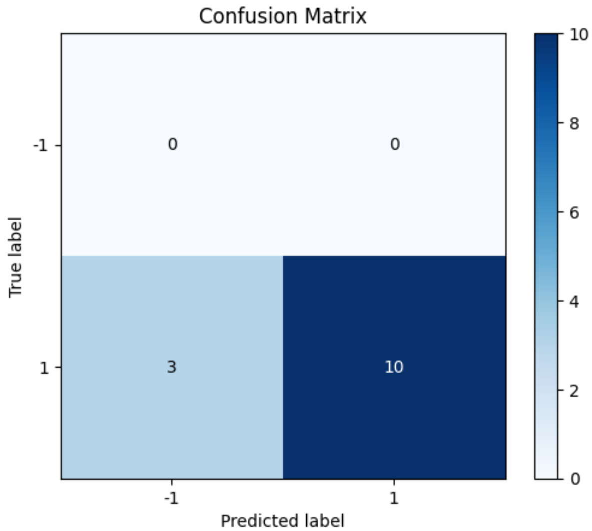


(a) NFLX Low Price

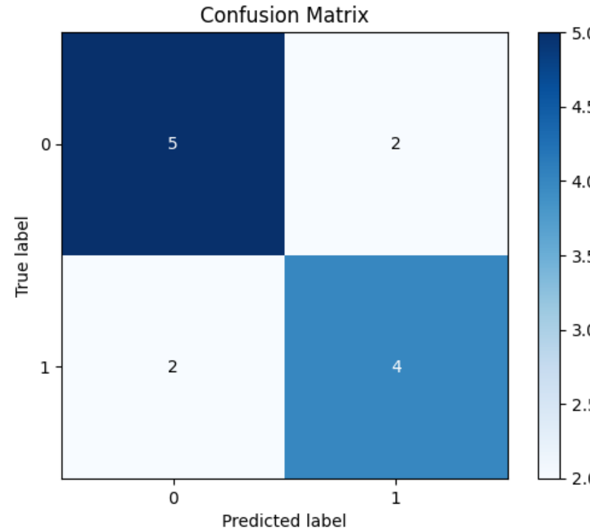


(b) NFLX Close Price

Figure 2: NFLX's Low and Close prices Confusion Matrices

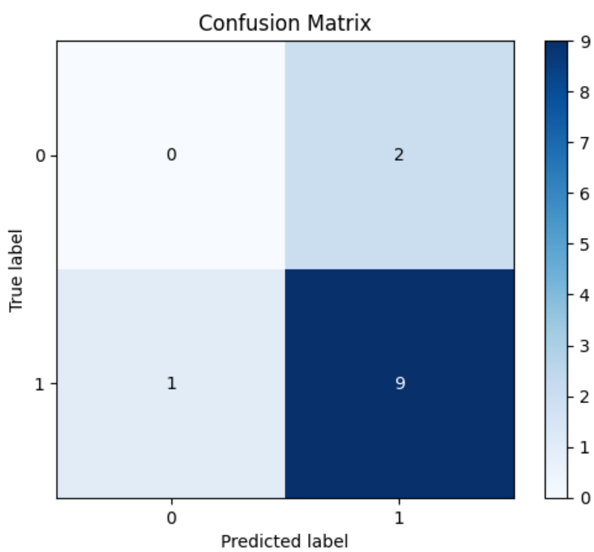


(a) NFLX Adj. Close Price

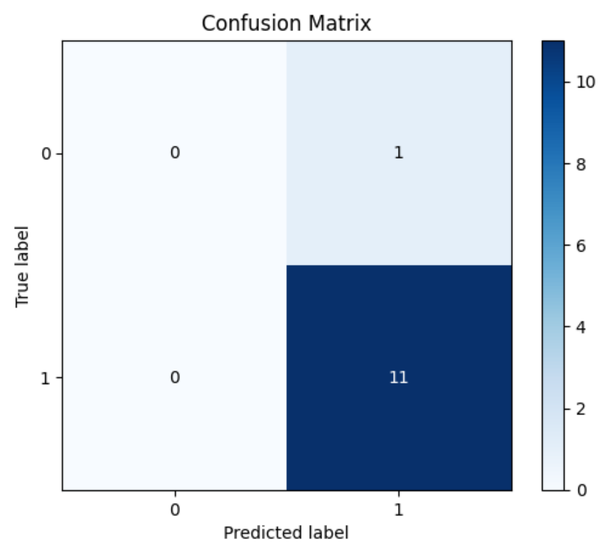


(b) NFLX Volume

Figure 3: NFLX's Adj. Close Price and Volume Confusion Matrices

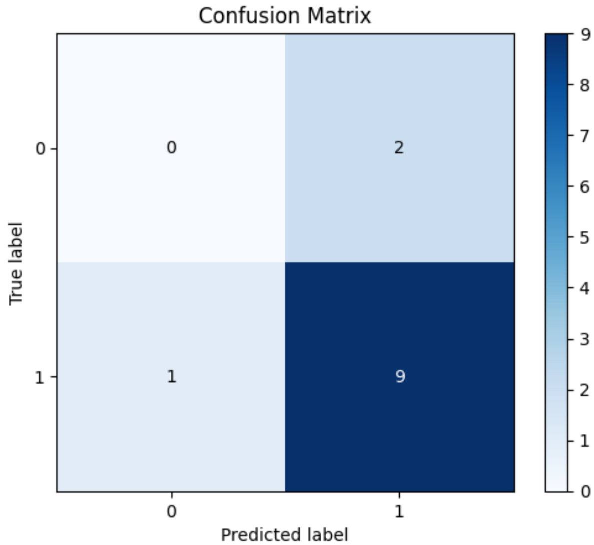


(a) GOOGL Open Price

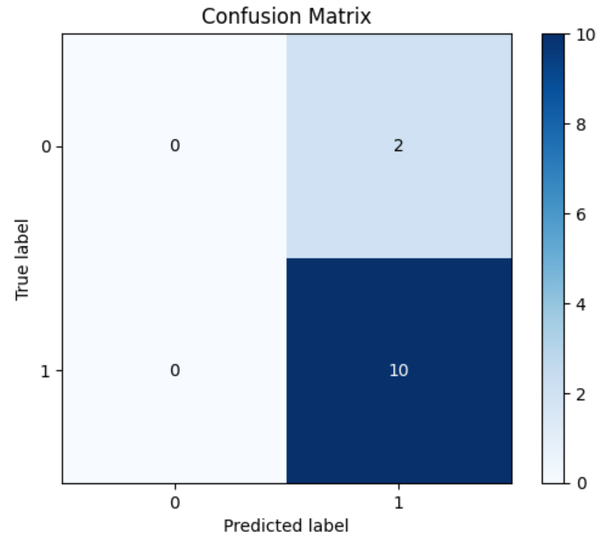


(b) GOOGL High Price

Figure 4: GOOGL's Open and High prices Confusion Matrices

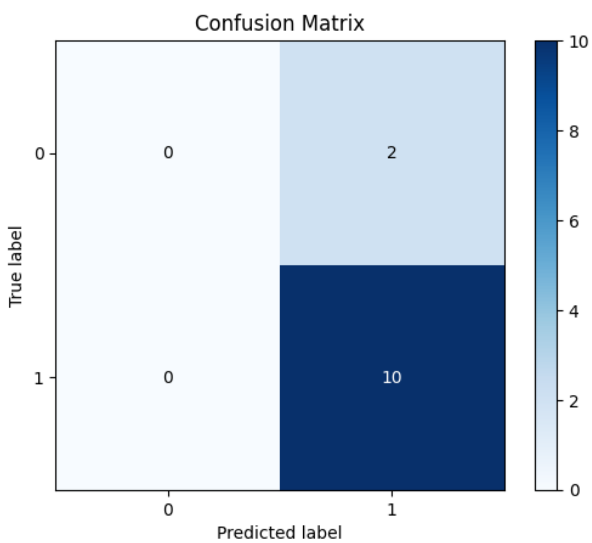


(a) GOOGL Low Price

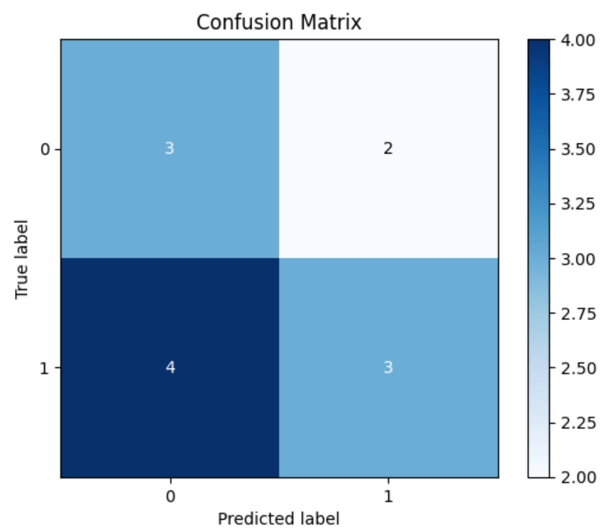


(b) GOOGL Close Price

Figure 5: GOOGL's Low and Close prices Confusion Matrices

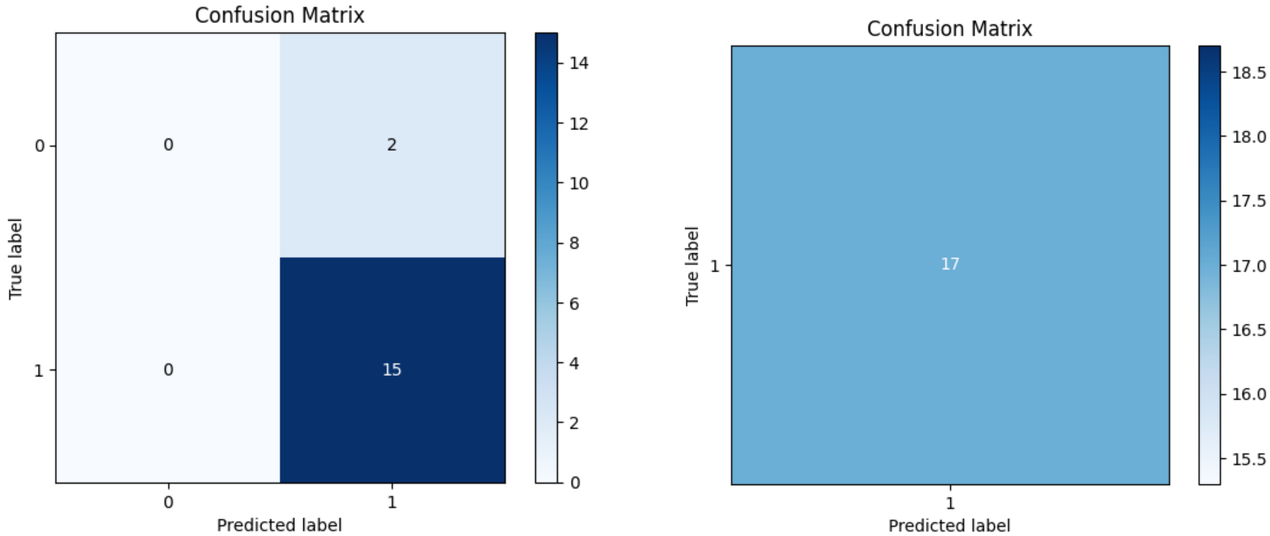


(a) GOOGL Adj. Close Price



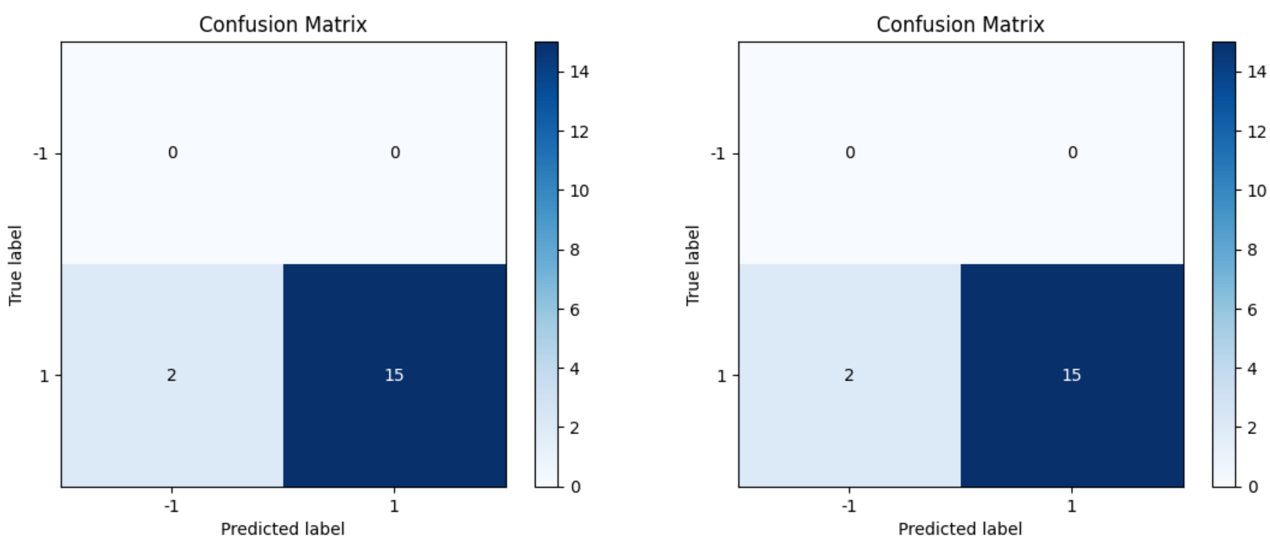
(b) GOOGL Volume

Figure 6: GOOGL's Adj. Close Price and Volume Confusion Matrices



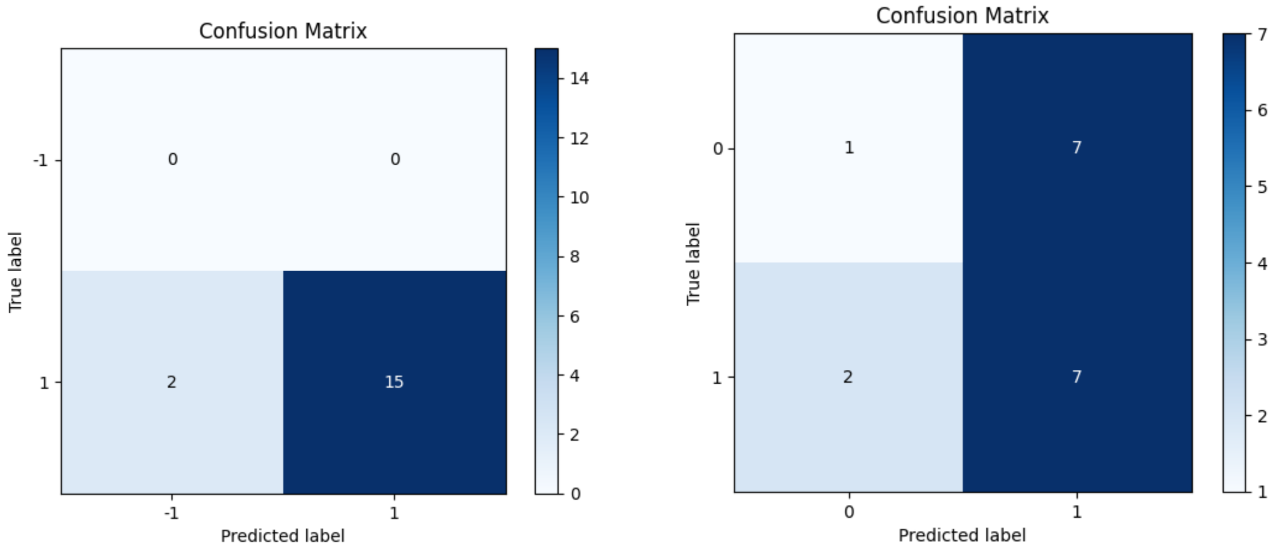
(a) AMZN Open Price (b) AMZN High Price

Figure 7: AMZN's Open and High prices Confusion Matrices



(a) AMZN Low Price (b) AMZN Close Price

Figure 8: AMZN's Low and Close prices Confusion Matrices



(a) AMZN Adj. Close Price

(b) AMZN Volume

Figure 9: AMZN's Adj. Close Price and Volume Confusion Matrices

Based on the analysis of the various confusion matrices, it can be deduced that the observed high accuracy is primarily attributed to the presence of a significant class imbalance. Notably, when examining the AMZN high price matrix depicted in Figure 7 (b), it becomes evident that it is particularly affected. Despite the utilization of a user threshold, the matrix still employs the Random Forest Classifier and fails to predict any structural breaks. To address this issue, attempts have been made to mitigate it through the implementation of the One-Class SVM approach wherever applicable. However, there is room for further enhancement by exploring alternative models and employing more refined thresholds to enable the utilization of different classification methods and models.

As anticipated, the data exhibits a substantial number of True values, indicating that the null hypothesis holds true for the majority of price observations. Conversely, there are very few instances of False values serving as true labels, mainly concerning the prices. This outcome aligns with expectations, given the inherent nature of financial markets. Nevertheless, it is important to acknowledge that such a significant class imbalance within the dataset may introduce concerns regarding forecasting on both models and must kept note.

5.3 Evaluation of Forecasting Models

Considering that there are three datasets to work on each with three different variants including the original benchmark dataset without including information on structural breaks, and two forecasting models using different evaluation metrics explained in earlier sections. Starting with the Vector Autoregression (VAR) Model:

Table 1: Benchmark VAR NFLX Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	97.97	22300.52	149.33	0.85
High	99.44	22957.18	151.52	0.85
Low	96.39	21575.87	146.89	0.85
Close	97.97	22284.39	149.28	0.85
Adj. Close	97.97	22284.39	149.28	0.85
Volume	12230162.71	498976975003176.4	22337792.53	0.67

Table 2: Variant 1 VAR NFLX Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	97.81	22265.57	149.22	0.84
High	99.27	22921.04	151.4	0.84
Low	96.23	21542.07	146.77	0.84
Close	97.81	22249.39	149.16	0.84
Adj. Close	97.81	22284.39	149.16	0.84
Volume	12220326.62	492985290180333.25	22203272.06	0.68

Table 3: Variant 2 VAR NFLX Forecast

	MAE	MSE	RMSE	MAPE
Open	97.88	22279.83	149.26	0.85
High	99.34	22935.79	151.45	0.85
Low	96.3	21555.83	146.82	0.85
Close	97.88	22263.66	149.21	0.85
Adj. Close	97.88	22263.66	149.21	0.85
Volume	12219294.82	493812730090611.44	22221897.54	0.68

First considering the NFLX dataset, comparing only the metrics on all features except Volume traded. It can be seen that both variants of the proposed model improve the forecasting power of Vector Autoregression (VAR) Model based on the MAE by an average of 0.16%, with MSE and RMSE it is an average of 0.07%. However with MAPE, the performance drops as the error increases by 1.2%. Overall it can be seen that the first variant of model is slightly better than the second.

Table 4: Benchmark VAR GOOGL Forecast

	MAE	MSE	RMSE	MAPE
Open	410.7	281708.1	530.76	0.53
High	413.2	285713.36	534.52	0.53
Low	408.22	277872.58	527.14	0.54
Close	410.95	282128.78	531.16	0.53
Adj. Close	410.95	282128.78	531.16	0.53
Volume	12294693.19	132805992131357.45	11524148.22	5.4

Table 5: Variant 1 VAR GOOGL Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	408.34	279948.78	529.1	0.53
High	410.83	283930.12	532.85	0.53
Low	405.89	276141.88	525.49	0.53
Close	410.95	282128.78	531.16	0.53
Adj. Close	408.6	280368.26	529.5	0.53
Volume	11452003.28	136170918842803.94	11669229.57	5.46

Table 6: Variant 2 VAR GOOGL Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	408.34	279817.51	528.98	0.53
High	410.83	283796.34	532.73	0.53
Low	405.89	276014.31	525.37	0.53
Close	408.6	280236.97	529.37	0.53
Adj. Close	408.6	280236.97	529.37	0.53
Volume	11453585.26	136252496521005.73	11672724.47	5.47

Next considering the GOOGL dataset and all of the features except for Volume traded, the MAE metric shows that forecasting power has improved across both variants by an average of 0.57%, the MSE and RMSE values show an improvement of 0.31% on average. However with the MAPE metric, no change is to be seen across both models as it stays the same at 0.53%. Overall, it can be seen that the second variant of the model is slightly better than the first.

Table 7: Benchmark VAR AMZN Forecast

	MAE	MSE	RMSE	MAPE
Open	578.1	672219.43	819.89	0.83
High	582.94	683506.88	826.74	0.83
Low	572.35	658480.17	811.47	0.83
Close	577.99	671622.5	819.53	0.83
Adj. Close	577.99	671622.5	819.53	0.83
Volume	5754513.21	3895457422838.54	6241359.97	1.73

Table 8: Variant 1 VAR AMZN Forecast

	MAE	MSE	RMSE	MAPE
Open	576.61	670431.69	818.8	0.83
High	581.42	681661.89	825.63	0.82
Low	570.89	656750.74	810.4	0.83
Close	576.5	669834.24	818.43	0.83
Adj. Close	576.5	669834.24	818.43	0.83
Volume	5855229.11	40224788171114.36	6342301.49	1.76

Table 9: Variant 2 VAR AMZN Forecast

	MAE	MSE	RMSE	MAPE
Open	577.66	671658.57	819.55	0.83
High	582.49	682927.21	826.39	0.83
Low	571.91	657938.64	811.13	0.83
Close	577.55	671061.77	819.18	0.83
Adj. Close	577.55	671061.77	819.18	0.83
Volume	5832277.5	39881377205378.58	6315170.4	1.76

Finally considering the AMZN dataset and all features except Volume traded. Based on MAE metric, the forecasting power has improved by an average of 0.25%. Regarding the MSE and RMSE metrics, they show an improvement of 0.13% on average. Finally with MAPE metric, there are no changes except for High price MAPE metric improving by 1.2% on the first variant model. Overall, again it can be seen that the first variant of the model is slightly better than the second.

It can be seen that the calculated metrics all stay around the same range for Open, High, Low, Close and Adj. Close prices for each respective stock except for their Volumes which have outstandingly high errors. This can be expected due to the much higher randomness of Volume of stock being traded, it is also clear that the proposed models do not necessarily improve the forecasting power for volume traded and do show increase in error metrics.

Before moving on to the LSTM forecasting model, it could be said that the LSTM models would most likely show better metrics than the Vector Autoregression (VAR) model. That being said, the metrics for all three datasets using the LSTM model are given below:

Table 10: Benchmark LSTM NFLX Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	40.22	3375.72	58.1	0.12
High	56.64	5747.42	75.81	0.18
Low	42.36	3761.94	61.33	0.13
Close	38.46	2528.98	50.29	0.13
Adj. Close	93.27	15548.52	124.69	0.3
Volume	21967673.63	626313053656373.9	25026247.29	3.18

Table 11: Variant 1 LSTM NFLX Forecast

	MAE	MSE	RMSE	MAPE
Open	103.92	26255.24	162.03	0.34
High	36.15	2575.16	50.75	0.14
Low	64.5	7415.2	86.11	0.25
Close	70.14	14927.33	122.18	0.22
Adj. Close	69.96	10550.01	102.71	0.24
Volume	8016639.73	142787256934606.9	11949362.2	1.07

Table 12: Variant 2 LSTM NFLX Forecast

	MAE	MSE	RMSE	MAPE
Open	14.55	439.15	20.96	0.06
High	35.19	2186.55	46.76	0.12
Low	29.27	1346.66	36.7	0.11
Close	34.95	1887.04	43.44	0.13
Adj. Close	33.21	2025.23	45.0	0.12
Volume	4105227.9	39698379369395.07	6300664.99	0.44

As mentioned before the benchmark LSTM model does in fact perform drastically better than the Vector Autoregression (VAR) models based on all metrics (for NFLX). Looking at the metrics from all three tables, what is interesting to note is that the first variant does not improve upon the benchmark's High, Low ,Close Prices and Volume traded of NFLX. The performance for Open price forecasting drops by around 299.6%, for Low price it drops by around 70.52% and, for Close price by 196.2% all of them on average across all four metrics. The second variant on the other hand, does not have the same issues and improves forecasting power all across the table being the best out of all models on the NFLX dataset. Here the second variant is much better than the first in terms of metric performance. Hence with structural break information on NFLX dataset the performance of forecasting has improved compared to benchmark dataset with no structural break information.

Table 13: Benchmark LSTM GOOGL Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	73.41	9012.97	94.94	0.06
High	41.14	3716.92	60.97	0.03
Low	21.76	805.17	28.38	0.02
Close	18.96	650.42	25.5	0.02
Adj. Close	17.06	580.66	24.1	0.01
Volume	738487.65	1209946808823.04	1099975.82	0.42

Table 14: Variant 1 LSTM GOOGL Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	118.76	96400.15	310.48	0.1
High	66.11	17606.74	132.69	0.05
Low	56.8	8070.93	89.84	0.05
Close	103.17	51211.99	226.3	0.08
Adj. Close	67.23	6590.94	81.18	0.06
Volume	867804.47	2397660236252.58	1548438.0	0.49

Table 15: Variant 2 LSTM GOOGL Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	24.93	1397.64	37.38	0.02
High	38.83	2173.38	46.62	0.03
Low	63.23	6495.95	80.6	0.05
Close	28.87	1400.12	37.42	0.02
Adj. Close	57.96	4753.2	68.94	0.05
Volume	940482.35	3210385495174.08	1791754.86	0.57

When looking at variant 1 metrics, it appears that the forecasting power has fallen all across the table as the benchmark still outperforms it based on the metrics. However, for the second variant the forecasting power for Open and High prices has improved, whereas it has fallen for the others. On average across all metrics, Open price forecasting has improved by 69.45% and for High price by 17.67%. Overall, the second variant performs better than the first, however compared to benchmark with no structural break information it only improves upon Open and High prices on the GOOGL dataset.

Table 16: Benchmark LSTM AMZN Forecast

	MAE	**MSE**	**RMSE**	**MAPE**
Open	2672.9	12578790.46	3546.66	1.73
High	1850.59	6447117.28	2539.12	1.16
Low	2960.36	15483929.89	3934.96	1.94
Close	4549.98	40048334.8	6328.38	2.83
Adj. Close	4753.1	41285421.54	6425.37	3.03
Volume	52845213.44	5085140007910850.0	71310167.63	13.89

Table 17: Variant 1 LSTM AMZN Forecast

	MAE	MSE	RMSE	MAPE
Open	424.96	231877.25	481.54	0.36
High	345.52	155718.26	394.61	0.27
Low	692.68	646392.94	803.99	0.61
Close	118.0	27683.95	166.38	0.08
Adj. Close	876.1	965133.93	982.41	0.74
Volume	1954896.31	7488492442383.01	2736511.0	0.47

Table 18: Variant 2 LSTM AMZN Forecast

	MAE	MSE	RMSE	MAPE
Open	542.7	334228.68	578.13	0.48
High	286.27	99163.13	314.9	0.26
Low	585.22	475711.82	689.72	0.44
Close	230.8	119958.98	346.35	0.15
Adj. Close	421.64	206925.41	454.89	0.37
Volume	11574287.88	240038034531021.8	15493160.9	2.94

At first glance, it can be seen that the benchmark dataset on LSTM model performs far worse than the on the Vector Autoregression (VAR) model. Again to be noted is that forecasting on both variants for Low Price is worse than that of the Vector Autoregression (VAR) model at any variant. The first variant shows better forecasting power on the Open and Close prices, but overall the second variant is better on the AMZN dataset.

6 Conclusion

In general, it can be asserted that the proposed models for detecting structural breaks exhibit notable enhancements in the context of both forecasting models when compared to the original dataset. While certain features may demonstrate sub-optimal performance in specific variants and models, an overall improvement in performance is observed in both variants, albeit in varying degrees.

The classifiers employed for predicting structural breaks in the dataset encounter a challenge of overfitting, primarily attributed to the presence of class imbalance. Therefore, it is important to delve into additional strategies to enhance the classifiers' performance, such as incorporating ensemble methods like bagging and boosting, thereby addressing the issue of overfitting and improving the accuracy of the predictions.

Regarding the basic Vector Autoregression (VAR) model, it becomes apparent that the first variant outperforms the second across both variants, leading to improved forecasting outcomes for features within both the NFLX and AMZN datasets.

Conversely, when considering the LSTM model, a contrasting pattern emerges. The second variant consistently demonstrates superior performance compared to the first across all three datasets. Notably, even in the AMZN dataset, the second variant exhibits enhanced forecasting capabilities for three features, surpassing the performance achieved by the first variant, which only improves forecasting for two features.

These findings highlight the intricate nature of the proposed structural break models and their impact on forecasting accuracy. It is imperative to delve further into the underlying reasons for the observed performance disparities such as to why the first variant performs better on a basic forecasting model while the second variant taking majority of all structural breaks of all features performs better on a Neural Network based model. Thorough analysis must be done on the contributing factors that led to the contrasting outcomes in order to gain a comprehensive understanding of the models' behavior and further refine their efficiency and accuracy.

7 Bibliography

References

[1] Hossein Abbasimehr and Reza Paki. Improving time series forecasting using lstm and attention models. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–19, 2022.

[2] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric reviews*, 29(5-6):594–621, 2010.

[3] Khaled A Althelaya, El-Sayed M El-Alfy, and Salahadin Mohammed. Stock market forecast using multivariate analysis with bidirectional and stacked (lstm, gru). In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–7. IEEE, 2018.

[4] Elena Andreou and Eric Ghysels. Structural breaks in financial time series. *Handbook of financial time series*, pages 839–870, 2009.

[5] Alexander Aue and Lajos Horváth. Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16, 2013.

[6] Satyanarayana Reddy Beeram and Swarna Kuchibhotla. Time series analysis on univariate and multivariate variables: a comprehensive survey. *Communication Software and Networks: Proceedings of INDIA 2019*, pages 119–126, 2020.

[7] Reza Ghanbari and Keivan Borna. Multivariate time-series prediction using lstm neural networks. In *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, pages 1–5. IEEE, 2021.

[8] Chonghui Guo, Menglin Lu, and Jingfeng Chen. An evaluation of time series summary statistics as features for clinical prediction tasks. *BMC medical informatics and decision making*, 20(1):1–20, 2020.

[9] John L Kling and David A Bessler. A comparison of multivariate forecasting procedures for economic time series. *International Journal of Forecasting*, 1(1):5–24, 1985.

[10] Suzan Köknar-Tezel and Longin Jan Latecki. Improving svm classification on imbalanced time series data sets with ghost points. *Knowledge and information systems*, 28:1–23, 2011.

[11] Hari S Luitel and Gerry J Mahar. A short note on the application of chow test of structural break in us gdp. *International Business Research*, 8(10):112, 2015.

[12] P Muthuramu, T Uma Maheswari, et al. Tests for structural breaks in time series analysis: A review of recent development. *Shanlax International Journal of Economics*, 7(4):66–79, 2019.

[13] John W. Pratt and Jean D. Gibbons. *Kolmogorov-Smirnov Two-Sample Tests*, pages 318–344. Springer New York, New York, NY, 1981.

[14] Francis EH Tay and Lijuan Cao. Application of support vector machines in financial time series forecasting. *omega*, 29(4):309–317, 2001.

[15] Sven Thies and Peter Molnár. Bayesian change point analysis of bitcoin returns. *Finance Research Letters*, 27:223–227, 2018.

[16] Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. *Modeling financial time series with S-PLUS®*, pages 385–429, 2006.