UNIVERSITY OF CAPE TOWN
DEPARTMENT OF COMPUTER SCIENCE

Computer Science Honours
Final Paper
2018

Title: The Notion of Concept Drift and The Prediction of The Stock Market

Author: Andre Venter

Project Abbreviation: STRMLN

Supervisor(s): Professor Deshen Moodley

Category	Min	Max	Chosen
Requirement Analysis and Design	0	20	0
Theoretical Analysis	0	25	0
Experiment Design and Execution	0	20	20
System Development and Implementation	0	15	5
Results, Findings and Conclusion	10	20	20
Aim Formulation and Background Work	10	15	15
Quality of Paper Writing and Presentation	10		10
Quality of Deliverables	10		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	0	10	
Total marks		**80**	

The Notion of Concept Drift and The Prediction of The Stock Market

Andre Venter
University of Cape Town
Student Number: VNTAND006

ABSTRACT

When trying to make predictions on real-time data that changes overtime and can be influenced by outside factors measures have to be taken in order to identify these changes and adapt adequately to them. In this paper two methods namely, Early Drift Detection Method and Page-Hinkley method, for identifying these changes and three methods, Sliding window, Adjusting Sliding Window, and an Ensemble approach for dealing with these changes will be discussed. Findings on the effectiveness of these drift detectors in collaboration with the drift mitigation methods will be explored and key findings and reasoning behind these results will provided.

KEYWORDS

Concept Drift, Machine Learning, Stock market, EDDM, Page-Hinkley, Sliding window, Adjusting Sliding window, Ensemble

1 INTRODUCTION

Machine learning based time series analysis has become an increasingly interesting problem due to the increase in data accessibility and availability. The increase in available computing power has allowed for more complex machine learning techniques to be explored and more accurate predictions to be made. When trying to make prediction with streamed time series data using machine learning predictions may become less and less accurate as time goes on [1]. One reasoning for this is that the concept of the underlying data has changed and the model has not or is not able to take this into account, the model may also be to fit to a specific concept and is not able to learn a new one or identify when the concept has changed. This is what is called concept drift. There has been methods created to in order to detect these changes along with methods to mitigate their impact. This paper discusses two methods used for detection and three methods used for mitigation of concept drifts [1].

The focus of the research was on the effect of concept drift on the closing prices of shares. There are multiple factors that can affect share prices, such as supply and demand, political turmoil in a country or even war. Most machine learning methods currently cannot detect or deal with these various underlying reasons for change. In hopes to solve some of these problems methods were explored and combine to see if they would lessen the impact of these underlying changing concepts.

The major aim of this project was to find an effective way of dealing with the concept drifts when they were detected or to mitigate them before they could impact the machine learning. By doing this it was hoped to see an improvement in prediction accuracy of the machine learning models and in doing so being able to expand this research into other fields that could use a more accurate predictor that deals with multiple changing concept.

2 Background

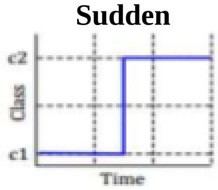
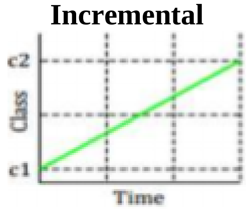
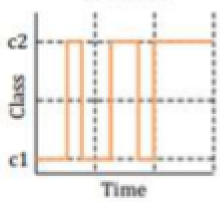
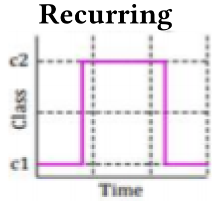
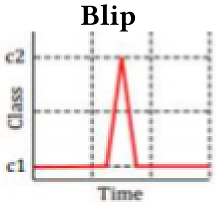
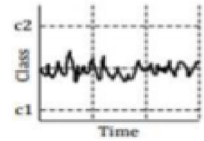
2.1 What is Concept Drift

The underlying concept of data is changing over time constantly and this concept may rely on some hidden context that is not a key predictive feature. A well-known and obvious example of this is weather forecasting, simply relying on past data is not an effective way of predicting future weather patterns as this may vary depending on seasons or climate change [2]. Another example is customers spending patterns, looking at previous spending patterns won't necessarily let you predicted their next purchase, this may depend on the month, interest rate, exchange rate, availability of substitutes etc. and often the reason for their change in spending patterns is hidden and not part of the learning task making it more difficult to predict with machine learning [2]. This is concept drift a change in the underlying data for some unknown reason. A machine learning model should be able to identify concept drifts and adapt or mitigate their effects, while distinguishing between noise and true drift. Being over sensitive and adapting to drifts is just as bad as adapting to late an effective model should be able to differentiate between noise and actual drifts but sensitive enough not to miss drifts. There are also recurring concept drifts what can be due to cyclic events such as different seasons and holiday events e.g. Christmas and New year's [1]. Thus, an ideal concept drift handling system should be able to: (1) quickly adapt to concept drift; (2) be able to distinguish noise from concept drift; and (3) recognize and deal with recurring concept drifts [1].

2.2 Types of Concept Drifts

The types of concept drifts and short definitions can be seen in Table 1. However the three most prevalent in the real world are. (1) Sudden (abrupt) concept drift, (2) gradual concept drift, and (3) recurring concept drift [3]. An example of (1) is someone graduating from university will now have different monetary concerns, whereas an example of (2) is the use of a piece of factory equipment will gradually change the quality of its output [4], and an example of (3) would be seasonality effects such as people spending more during Christmas but afterwards the spending goes back to previous levels [4]. The hidden changes in context may cause a change of target concept and also may cause a change in the underlying data distribution, both types indicate that the model needs updating.

Table 1: Types of Concept drifts and definitions Extracted from [3] page 2.

Concept Drift With Figure	Definition
Sudden 	Abrupt changes that irreversibly change the variables class assignment. E.g. Seasonal changes on sales.
Incremental 	Variables slowly change their values over time. E.g. price growth due to inflation.
Gradual 	Variables slowly change their class distribution over time. E.g. output of factory equipment.
Recurring 	Changes that are occurring are not permanent and are reverted after some time.
Blip 	Random, and can be ignored as it can't be adequately monitored e.g. brief increase in stock price.
Noise 	Not a concept drift as it is does not heavily affect data but must be able to tell the difference between drift and noise.

2.3 Concept Drifts and the JSE

Due to the nature of change of data over time the underlying concepts of the data is bound to change due to some unforeseen circumstances. Concept drift as explained and explored in section 2.1, is when the properties of the target variable that a machine learning model is trying to predict changes over time for some unforeseen and unknown reason. This is a problem with in time series data and the data may face one or many different types of drifts mentioned in section 2.3. The financial stock market is no exception to this case and the underlying concepts could be effected due to things such as seasonality or inflation. This is why when making prediction of the finical stock market that there are measures in place in order to detect and deal with concept drifts when they are present.[5,6].

The Johannesburg Stock Exchange (JSE), is the largest South African financial stock market [7]. When making predictions on the JSE there will be a constantly inflow of new and updated data that the machine learning model can use in real time. This data may be influenced by a varying amount of different concept drifts and the underlying data changing for some unknown or unforeseen reason, such as political instability that is common in South Africa.

2.4 Related Works

Harries and Horn [8] discuss the idea of dealing with concept drift by only providing a prediction when the next information e.g. share price, asset share value is similar to that of the training data, this was done by associating a set of permitted attribute ranges with each leaf of a decision tree. To provide a prediction the attributes of the information must be within the range of the training data. Their method was tested on the Sydney Futures Exchange (SFE) and it was seen that there was an 8 percent increase in accuracy after drift detection and avoidance methods were put in place [8]. In the paper by Bruno Silva et al. [9] they examine the effect of using neural networks to detect concept drift in financial markets. To do this they go through two modules, the first module uses an artificial neural network that takes incoming streams and produces aggregations, and compressing the data while retaining the relationships within the data. The output of this module goes into a second module that uses a fixed set of these aggregations and produces an output that determines if there is concept drift present or not [9]. The method was tested on the Dow Jones Industrial index (DJI) using the daily share index prices for test data.

3 Evaluation Metrics

To evaluate the performance of the models, the data will be split up into training data and test data. 90 data points for training, 10 data points for validating and 20 data points for testing. E.g. data points 1-90 was used to train the model, data points 90-100 was used for validating the model and data points 100-120 was used for testing the model on unseen data. The model then would move on to data points 20-110, 110-120, and 120-140 for training, validating and testing respectively. It would do this until it had used all the data points it had available. The test data was used to assess the three measures below,

i Percent error: tool for determining the precision of your calculations.((PredictedValue-ActualValue)/ActualValue)*100

ii Model accuracy: Proportion of the number of correct predictions compared to the total

iii.Drifts detected: How may drifts did the drift detectors identify.

4 Methods for Detecting Concept Drift

As the types of drifts that would be encountered were unknown two of the most common drifts were chosen namely gradual and abrupt concept drifts and two of the most prominent drift detectors for these two types of drifts were chosen, Early Drift Detection Method(EDDM) and the Page-Hinkley Drift Detection Method, for gradual and abrupt drift detection respectively.

4.1 Early Drift Detection Method

The Early Drift Detection Method (EDDM), was created to improve open detection on gradual concept drift while at the same time keeping a good performance at detecting abrupt concept drifts. It's serves to detect whether or not your model itself has started to drift instead of the data. The idea is to identify the distance between the errors not only how many errors there are. While the machine learning method is learning it will get more predictions correct and the distance between the errors will increase. The average distance between two errors (p) and the standard deviation (s) is stored and when p + 2(s) reaches the maximum that is when the distance between the errors is max and the standard deviation is max this indicates that a drift has been detected [2,10].

4.2 Page-Hinkley Method

The Page-Hinkley test (PHT) is a sequential analysis technique typically used for monitoring change detection. It allows for abrupt drift detection. The PH test is a sequential adaptation of the detection of an abrupt change in the average of a Gaussian signal. Considers a cumulative variable m(t), defined as the cumulated difference between the observed values and their mean till the current moment [2,11]. Unlike the EDDM method it is used to check whether the data has drifted and not if your model has.

Algorithm 1 Page-Hinkley drift detection algorithm

1. **for** $t > 0$

    1.1 Computes

    $$\bar{x_T} = 1/T \sum_{t=1}^{t} x_t$$
    $$U_T = \sum_{t=1}^{T} (x_t - \bar{x}_T - \delta)$$
    $$m_T = min(U_t, t = 1 \dots T)$$

    If $PH_T = U_T - m_T > \lambda$

        return and report a change at time t_{PH}

    else

        return to 1.1

5 Methods for Mitigating Concept Drift

5.1 Sliding Window Method

The problem is phrased so that multiple recent time steps can be used to make the prediction for the next time step, the window size can be tuned for each problem. For example using a window of size 2, given a time (t) we want to be able to predict the next value at the next time stamp (t+1), we use the current time (t) along with two prior times (t-1 and t-2). When phrased as a regression problem the input variables are t-2, t-1, t and the output variable is t+1 [12, 13].

5.2 Adjusting Sliding Window Method

The window size works the same as with the sliding window, given a time (t) we want to be able to predict the next value at the next time stamp (t+1), we use the current time (t) along with two prior times (t-1 and t-2). However how it is different to a normal sliding window is that the size of window changes depending on if a drift is detected or not. If a drift is detected the window size will be halved and the model will be retrained on this smaller window size and predictions will carry on where it left off. However if (n) correct predictions are made without a drift the window size will return to its original size [12,13].

5.3 Ensemble Method

Multiple models are all trained on different parts of data and then these different models are given a time (t) and they each make a prediction of what they calculate to be (t+1). All these predictions are then summed together and divided by the amount of models used to get an average and this average is the final prediction [14]. The idea is by training multiple models on different parts of the data if multiple concepts are overlapping it will have a lesser impact on the final prediction.

6 Implementation

In order to measure the amount of drifts along with incorporating in drift mitigation methods into the machine learning the drift detection methods had to be in implemented into the machine learning process. Also further for the EDDM method to work more efficiently a range in which a prediction was considered correct or not was needed which will be latter discussed in the results section. All models were trained and tested the same way as explained in section 3,

Baseline: A Neural network model in the form of a Multi-layer Perceptron (MLP). A MLP is a feedforward artificial neural network it creates a set of out puts from a set of inputs. It has several layers of input nodes connected as a directed graph between the input and output layer and makes use of backpropagation for training. The input that the neural net takes is a single value which is the previous time stamp (t-1) actual weekly closing price at that point. The output is a prediction for the expected weekly closing price at time stamp (t). In order to count the amount of drifts the EDDM drift detector was used with it, every time the model made a prediction and it was right it would tell the EDDM True, an when it was incorrect it would tell the EDDM False. When the EDDM detects a drift it will add it to a counter that tracks the total amount of times the machine learning has drifted.

Sliding Window: Similar to the baseline the EDDM detection method was incorporated into the learning. At the bases of the method the machine learning technique is also a multi-layer perceptron neural net with an adjustable lookback that is used to set the size of the window used to train and predict. The input is a set of values from (t-1...t-n) where n is determined by the look back or in other words the size of the window. These values are the weekly closing prices at each of these time stamps. The output of the neural net is a single value the weekly closing price at time (t). As the windowed neural net learns and predicts it tell the EDDM whether it had gotten the prediction correct (True) or wrong (False) and when the EDDM determines there was a drift detected it updates the drift counter.

Adjusting Sliding Window: This method works much like the normal Sliding Window method, the inputs and outputs are the same. However while training and predicting, the window size changes and the neural net would use this new window size on its next predictions. This method used both the Page-Hinkley and EDDM method and also used a combination of the two. When Page-Hinkley detected a drift in the data the window sized would halve and the model would be retrained on this smaller window size. After (n) correct predictions without a drift the window size was set back to the original size. The EDDM worked the same if there was a drift in the learning the window size would be halved and after (n) correct predictions without a drift the window size would be set back to the default. In the final implementation the EDDM and PH methods were used together, when either one of them detected a drift the window size would be halved, and after (n) correct predictions without detecting a drift from either drift detectors the window size would be set back to its original size.

Ensemble Method: In This method the training data is split up into four parts and four different models are trained on one of these parts each. These models each make their own prediction and then these prediction are added together and averaged to give the final prediction. Each of these models is a neural network model in the form of a Multi-layer Perceptron, the same as the sliding window methods and baseline. As an input they each take in the data point at time (t-1) and make a prediction of the value at time (t).

7 Results and Discussion

For data weekly closing prices from the JSE was used, and the most volatile, medium volatility, and least volatile, shares were chosen for testing. We regard volatility as low, medium or high. The higher the volatility the more liable to change and uncertainty the dataset. Naspers, a broad-based multinational internet and media group, for the most volatile dataset, Redfine, a Real estate investment trust company, for the least volatile dataset, and The Foschini Group (TFG) a South African clothing retail company was chosen for the medium volatility dataset. In order to better test the EDDM method and better compare the accuracy of the predication methods a prediction was considered correct if it was with-in 2 percent of the actual value. This value determined through experimentation to obtain a meaningful accuracy that could be measured and insuring the EDDM moved could effectively determine drifts or not. Further the percent error was calculated on each prediction and then averaged across the total predictions.

7.1 Datasets

Three datasets were chosen to be used for training, testing and evaluating based on their volatility.

• **Dataset 1**: Naspers was selected as a dataset as its daily closing price on the JSE is very volatile with a variance of 83783 and a mean of 52326 over a period of 9.5 years.

• **Dataset 2**: TFG was selected as a dataset as its daily closing price on the JSE is relatively stable with a variance of 5228 and a mean of 5991 over 9.5 years.

• **Dataset 3**: Redefine was selected as a dataset as its daily closing price on the JSE is very stable with a variance of 324 and a mean of 669 over 9.5 years.

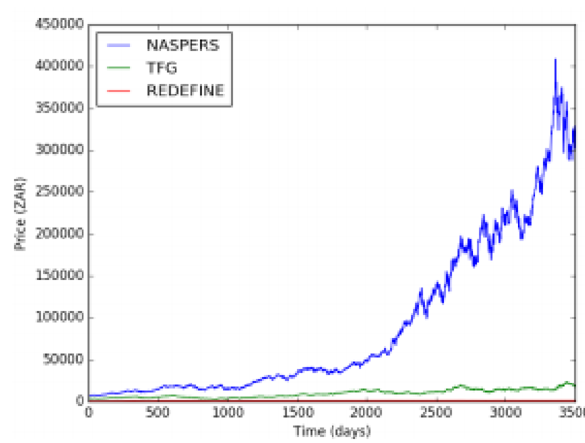


Figure 1: Datasets volatility

Table 2: Baseline Results

Dataset	Percent Error	Drift Count	Percent Correct Predictions
NaspersAVG	3.04%	0	42.6%
TFGAVG	3.29%	3	36.8%
RedfineAVG	2.28%	5	55.6%

7.2 Fixed Sliding Window

Looking at tables 3, 4 and 5 the sliding window methods results, it can be seen that only the TFG dataset performed better than the baseline with a window size of 5. However the Naspers' and Redefine datasets only did slightly worse than that of the baseline. Also from the results it can be seen as the window size gets larger and larger the accuracy starts declining. This could be due to the fact that as it is using the weekly average as more data is being looked at concepts begin to overlap. The least volatile dataset most likely preforms worse due to the fact that the subtle change in the data allows the baseline that provides almost the same value of (t-1) for its prediction of (t) to be right more often than not due to the little change from week to week. For a window size of 5 the TFG dataset out preforms the baseline method by 2%, but does worse with larger window sizes. This is most likely due to the fact that there is enough change in the data that the baselines prediction of almost the same value is not as accurate anymore but if the window becomes to large the volatility of the data begins to effect it again and multiple concepts are being used. This is also the reason why the volatile data preforms worse than the baseline as the constant fluctuation of the data makes using a sliding window to be effected by this more.

Table 3: Naspers weekly average sliding window method results

Window size	Percent Error	Drift Count	Percent Correct Predictions
5	3.12%	3	41.4%
10	3.16%	0	39.6%
15	3.73%	4	36.1%
20	3.82%	2	34.5%
30	3.89%	3	33.75%
40	4.98%	3	27.3%
50	6.58%	4	21.8%

Table 4: TFG weekly average sliding window method results

Window size	Percent Error	Drift Count	Percent Correct Predictions
5	3.27%	6	38.8%
10	3.48%	2	36.1%
15	3.52%	1	36.3%
20	3.86%	4	33.5%
30	4.39%	6	29.6%
40	6.53%	2	21.4%
50	8.07%	3	18.0%

Table 5: Redefine weekly average sliding window method results

Window size	Percent Error	Drift Count	Percent Correct Predictions
5	2.35%	4	53.8%
10	2.55%	4	49.5%
15	2.56%	6	49.8%
20	2.86%	4	47.6%
30	3.24%	5	43.4%
40	3.76%	5	38.5%
50	4.44%	4	31.9%

7.3 Adjusting Sliding Window

For the Adjusting sliding window the initial window size was set at 10 due to it allowing for adequate adjustments to the window size to be made without reaching the minimum lookback too quickly. When looking at tables 6, 7 and 8 the adjusting window method results for the least volatile dataset of Redefine, struggled compared to having a constant window size and also performed worse than the baseline, this seems to show that if the shares are less volatile that it becomes too difficult for these two drift detectors to pick up where the concept changes are. Looking at NaspersAVG results from the same tables it performed better than its baseline and fixed window comparison with the Page-Hinkley method. This could be due to the fact that the adjusting window allows it to capture more of the current concept when the shares are less volatile and using a smaller window when they are more volatile to cut out concepts. Looking at the TFG dataset it can also be seen that the Page-Hinkley drift detection worked best with it compared to the others, even outperforming the fixed window by 3.4% and the baseline by 5.4%. Similarly to the more volatile shares this could be due to the fact that when the data is less volatile a larger part of the concept can be captured and when the data starts becoming more volatile due to maybe multiple concepts being present it can use a smaller window size to focus on the correct concept while predicting.

For the Naspers and TFG datasets the Page-Hinkley drift detector allowed the adjusting window method work better than using only the EDDM method or using both the EDDM and Page-Hinkley method together. Due to the datasets being more volatile and having more abrupt changes the EDDM drift detector may be ineffective in determining drifts and when it is being used it picks up blips or noise telling the model to adjust the window size when it shouldn't. It can also be seen that the least volatile data when using the drift detectors separately that it performed better when using the EDDM method. This could be because there is not many abrupt changes in the data and the Page-Hinkley method isn't able to effectively identify all the drifts. Also the Page-Hinkley may be suffering more from blips and noise as most of the data is similar and gradually changing when there is a big jump for an instance it may determine this as a drift adjusting the window size when it is not needed.

Table 6: Adjusting sliding window method using EDDM method results

Dataset	Percent Error	Drift Count	Percent Correct Predictions
NaspersAVG	3.12%	2	39.3%
TFGAVG	3.34%	2	35.6%
RedfineAVG	2.49%	5	50.4%

Table 7: Adjusting sliding window method using Page-Hinkley method results

Dataset	Percent Error	Drift Count	Percent Correct Predictions
NaspersAVG	2.94%	45	44.3%
TFGAVG	3.11%	39	42.2%
RedfineAVG	2.87%	29	47.7%

Table 8: Adjusting sliding window method using Page-Hinkley and EDDM methods results

Dataset	Percent Error	Drift Count	Percent Correct Predictions
NaspersAVG	2.98%	47	42.1%
TFGAVG	3.28%	46	35.9%
RedfineAVG	2.44%	33	52.2%

7.4 Ensemble Method

Looking at table 9 it can be seen that the ensemble method preformed the best for the non-volatile dataset (Redefine) compared to all the other methods used for it including the baseline. A reason for this may be that due to the training data being split-up if there is a start or end of a different concept being used by the model its impact is lessened due to the final prediction being an average across the 4 different models. Handling the problem better than the adjusting sliding window that may not be able to identify all the concepts due to the minimal change. Also looking at the dataset that has a medium volatility (TFG) the ensemble method performed better than the baseline and the fixed sliding window. Similarly to the non-volatile dataset this is most likely due to the fact that when the data is less volatile and concept changes are difficult to identify and when multiple concept are being used it has a lesser impact on the final prediction due to being averaged across the 4 models. However the dataset that is the most volatile did not perform better than the baseline most likely due to the big changes in the data, so when the models are each trained on different parts of the data they are each capturing their own concept and averaging it has a lesser effect than on the other two datasets.

Table 9: Ensemble method results

Dataset	Percent Error	Drift Count	Percent Correct Predictions
NaspersAVG	3.01%	1	42.0%
TFGAVG	3.18%	1	39.3%
RedfineAVG	2.24%	5	56.7%

7.4 Overall Evaluation

In table 10 the overall best performing models can be seen for each dataset. From this table it can be seen that for the more volatile datasets (TFG and Naspers) the adjusting window with the Page-Hinkley drift detection method preformed the best and for the least volatile dataset the ensemble method preformed the best. It should also be noted that all the methods for the medium volatile dataset (TFG) out preformed the baseline method. The TFG dataset also had the best performance boost overall, having an accuracy increase by 5.4% and in decrease in the percent error by 0.18%. Whereas the volatile dataset (Naspers) had an accuracy increase of 1.7% and a decrease in the percent error of 0.1%. The best performance for the least volatile dataset (Redefine) has an accuracy increase of 1.2% and a percent error decrease of 0.04%. A reasoning for the lower effectiveness for the low volatility dataset (Redefine) could be because the baseline makes a prediction that only slightly different to what it takes as an input, and gets it correct more often than not as the data changes only slightly from point to point, were as the more volatile the data becomes the less effective this is as the changing patterns and concepts more heavily effect the data making the use of mitigation methods overall more effective.

Table 10: Comparison of all methods on each dataset

	Redefine	**TFG**	**Naspers**
Dataset Volatility	Low	Medium	High
MLP Baseline lookback of 1	Accuracy: 55.6% Percent Error: 2.28%	Accuracy: 36.8% Percent Error: 3.29%	Accuracy: 42.6% Percent Error: 3.04%
MLP: Fixed Sliding Window Lookback of 5	Accuracy: 53.8% Percent Error: 2.35%	Accuracy: 38.8% Percent Error: 3.27%	Accuracy: 41.4% Percent Error: 3.12%
MLP: Adjusting Sliding Window EDDM	Accuracy: 50.4% Percent Error: 2.49%	Accuracy: 35.6% Percent Error: 3.34%	Accuracy: 39.3% Percent Error: 3.12%
MLP: Adjusting Sliding Window PH	Accuracy: 47.7% Percent Error: 8.18%	Accuracy: **42.2%** Percent Error: **3.11%**	Accuracy: **44.3%** Percent Error: **2.94%**
MLP: Adjusting Sliding Window EDDM and PH	Accuracy: 52.2% Percent Error: 2.44%	Accuracy: 35.9% Percent Error: 3.28%	Accuracy: 42.1% Percent Error: 2.98%
MLP: Ensemble Method	Accuracy: **56.7%** Percent Error: **2.24%**	Accuracy: 39.3% Percent Error: 3.18%	Accuracy: 42.0% Percent Error: 3.01%

8 Conclusions and Future Work

This paper presented three methods along with sub methods for mitigating concept drift, Sliding window method, Adjusting Sliding window method and an Ensemble method. In order to improve their effectiveness they were used in conjunction with two different drift detection methods, The Early Drift Detection Method and the Page-Hinkley drift detection method. All the methods were tested and evaluated using the closing share price weekly average of Redfine, Naspers and The Foschini Group obtained from the Johannesburg stock exchange.

From experimental results of the non-volatile shares it can be seen baseline method outperformed both the window methods, getting a 55.6% accuracy and 2.28% percent error, but the Ensemble method did the best out of all the methods and did better than the baseline getting an accuracy of 56.7% and a percent error of 2.24%. For future works further exploration into the ensemble method seems to be the best place to start. Looking into incorporating drift detectors within each model to add a bias or weighting when making the final prediction. Also looking into using monthly averages may help as then the data could have a more significant change and the window methods may perform better.

For volatile shares the best performer was the Adjusting sliding window method with the Page-Hinkley drift detector, preforming better than the baseline getting an accuracy of 44.3% and a percent error of 2.94% compared to that of the baseline of an accuracy of 42.6% and percent error of 3.04%. The adjusting sliding window with the use of different drift detectors could be looked into in attempts to improve the accuracy of the method.

Looking at the medium volatile dataset the fixed sliding window with a window size of 5, the adjusting sliding window with the Page-Hinkley drift detection method and also the ensemble method performed better than the baseline with accuracies of 38.8%, 42.2% and 39.3% respectively and percent errors of 3.27%, 3.11% and 3.18% respectively. Were as the baseline method had an accuracy of 36.8% and a percent error of 3.29%. Since all these methods outperformed the baseline all of them provide avenues for future work. However as the ensemble method was looked into the least and further exploration into combining it with drift detectors should be a good place to start. The adjusting window could also be combined with different drift detectors in an attempt to improve its accuracy.

A key take away from the findings is that finding more drifts does not always indicate worse accuracy as can be seen in the results of the fixed sliding window. When the size of the window is 10 on the TFG dataset it had detected less drifts than when the window size was 5 however the smaller window size still had a higher accuracy that the bigger one. It can further be seen looking again at the TFG data set and tables 7 and 8 more drifts were picked up when using both the EDDM and Page-Hinkely detectors together but the model preformed worse most likely due to detecting noise or a blip and not true change.

For future works it could also be looked into using different drift detectors for the adjusting sliding window method and the ensemble method. There may be more effective drift detectors for these methods that could boost the performance for all the datasets. The datasets could also be transformed more or in different ways so that the current detectors work better, due to time constraints only weekly averages were used but monthly or bi-weekly averages may allow the methods to perform better. The ensemble method was also not fully explored and could be obtain better accuracy if the parameters were fine-tuned more and also if an effective way to incorporate the drift detectors into the individual models could be done a further parameter could be added so that models that are predicting incorrectly can be ignored more often and the final prediction could be more accurate.

REFERENCES

[1] K. Nishida. 2008. Learning and Detecting Concept drift. Ph.D. Dissertation. Hokkaido University, https://doi.org/10.1007/978-3-540-75488-6_27

[2] Y. Kadwe, V. Suryawanshi. 2015. A Review on Concept Dtift. IOSR Journal of Computer Engineering, (Jan. 2015), 20-26. DOI: https://doi.org/10.9790/0661-17122026

[3] P.K.Srimani, M. Patil. Mining Data Streams With Concept Drift In Massive Online Analysis Frame Work. Pages 133-142. WSEAS TRANSACTIONS on COMPUTERS. E-ISSN: 2224-2872. 2016

[4] A. Tsymbal Department of Computer Science Trinity College Dublin, Ireland, "The problem of concept drift: definitions and related work ", April 29, 2004.TCDCS-2004-15.

[5] I. Zliobaite. 2010. Learning under Concept Drift: an Overview. (22 Oct, 2010). arXiv:1010.4784v1

[6] R. Cavalcante, A. Oliveira. 2015. An approach to handle concept drift in financial time series based on Extreme Learning Machines and explicit Drift Detection. (01 Oct. 2015). DOI: https://doi.org/10.1109/IJCNN.2015.7280721

[7] JSE. Johannesburg stock exchange. JSE learning modules. 2013. Retrieved April 20, 2018 from https://www.jse.co.za/grow-my-wealth/what-are-financial-markets

[8] Michael Harries and Kim Horn. 1995. Detecting Concept Drift in Financial Time Series Prediction using Symbolic Machine Learning.

[9] Bruno Silva, Nuno Marques, and G Panosso. 2012. Applying neural networks for concept drift detection in financial markets. 960 (01 2012), 43–47.

[10] S. Yu, Z. Abraham, H. Wang, M. Shah and J. Principe. 2017. Concept Drift Detection and Adaptation with Hierarchical Hypothesis Testing. (25 Jul. 2017). arXiv:1707.07821v1. . Retrieved from https://arxiv.org/abs/1707.07821v1

[11] R. P. Bose, W. van der Aalst, I. Žliobaitė, and M. Pechenizkiy. 2013. Dealing With Concept Drifts in Process Mining. (16 Oct. 2013). DOI: https://doi.org/10.1109/TNNLS.2013.2278313

[12] Vahida Attar, Prashant Chaudhary, Sonali Rahagude, Gaurish Chaudhari, and Pradeep Sinha. 2012. An Instance-Window Based Classification Algorithm for Handling Gradual Concept Drifts. In Agents and Data Mining Interaction, Longbing Cao, Ana L. C. Bazzan, Andreas L. Symeonidis, Vladimir I. Gorodetsky, Gerhard Weiss, and Philip S. Yu (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 156–172.

[13] Yu Sun, Ke Tang, Zexuan Zhu, and Xin Yao. 2017. Concept Drift Adaptation by Exploiting Historical Knowledge. CoRR abs/1702.03500 (2017). arXiv:1702.03500 http://arxiv.org/abs/1702.03500

[14] J.W.Taylor, R. Bulzza. 2002. Neural network load forecasting with weather ensemble predictions. (Aug 2002). DOI: https://doi.org/10.1109/TPWRS.2002.800906

[15] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[16] Eric Jones, Travis Oliphant, Pearu Peterson, et al. 2001–. SciPy: Open source scientific tools for Python. (2001–). http://www.scipy.org/ [Online; accessed 11-07-2018].

[17] MOA. Massive Online Analysis. MOA Machine Learning for Streams. 2018. Retrieved April 21, 2018 from https://moa.cms.waikato.ac.nz/

[18] François Chollet et al. 2015. Keras. https://keras.io. (2015).