Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

AI in Financial Markets

Semester Thesis

Tim Gretler

tgretler@student.ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Ye Wang, Zhongnan Qu
Prof. Dr. Roger Wattenhofer

January 17, 2021

Abstract

The goal of financial market participants is to predict the future. Correctly predicting future prices translates to monetary gain. This work analyzes previous work that uses recurrent neural networks to predict stocks. These proven techniques are adapted to the bond market, which is the biggest financial market.
The performance achieved on the bond market is unsatisfactory because it can't beat simple prediction schemes such as predicting the last day for tomorrow. This shortcoming can also be shown in the previous work. The encountered difficulties can be explained with financial market theory concepts, such as the efficient market hypothesis.
Resulting from our experimental and theoretical work, we propose a rethinking of benchmarks in financial market prediction. The proposed suggestion try to make the development and comparison of models easier.

Contents

Chapter 1
Introduction

Financial Markets are a marketplace where many different agents can exchange different securities; these include stock shares and bonds. In 1971 the first automatic trading system was introduced, known as the National Association of Securities Dealers Automated Quotations - in short, NASDAQ. Today nearly all trades are made digitally on automatic trading systems. With the introduction of electronic trading systems, the rise of computer-based trading began. Currently, 35% of all public assets are managed by computers, and 80-90% off all stock share trades are done by an algorithm [1]. Most of the computer-managed assets are passively managed and follow a fixed strategy but there also exist funds that follow a dynamic computer-managed strategy. The funds are called quant funds and they manage approximately 2.5% of all total public assets. Quant funds use a variety of numerical strategies, which include neural networks.
This work consists of two parts. The first part looks at US government bond data and tries to transfer previous work [2] on stock market prediction to bond market prediction. As in the previous work, the method used for prediction is a LSTM neural networks. Using LSTM networks serves the goal of including the time dependencies of financial market data. The second part looks at the characteristics of financial markets and how they influence the evaluation and training of neural networks for financial markets. This part can be regarded as follow up work on the prediction issues encountered in the first part.

Chapter 2

Background

2.1 Financial Markets

Financial markets are a type of market where agents can meet to exchange securities. On an exchange, an agent can exchange money for a security. In the context of this work, two types of securities are of particular interest: Stocks and bonds.

2.1.1 Stock shares

Stock shares represent ownership in the capital stock of a company. These shares are traded in financial markets. There are different valuation methods for stocks. A commonly used method is the discounted cash flow method. This method discounts all future cash flows and the sum of the discounted cash flow is the valuation of the stock.

$$StockPrice = \frac{CF_0}{(1+r_0)} + \frac{CF_1}{(1+r_1)^2} + ... = \sum_{n=1}^{\infty} \frac{CF_n}{(1+r_n)^n} \tag{2.1}$$

$$r_i = \text{Discount rate i years from now}$$
$$CF_i = \text{Cashflow i years from now}$$

This represents the theoretical value of a stock but this method is not used in practice since there exist too many factors to estimate. One would have to estimate the discount factors and cash flows far into the future to get an accurate estimate. Typically more practical methods are used. In general, there exist two popular methods. The first is to do fundamental analysis and look at the economic data that influence the stock price. This means looking at the balance sheet and other factors to create an outlook for the future and estimate the stock's value according to the outlook. The second method is technical analysis. This method involves looking at past stock prices and trying to predict trends and future price changes. Technical analysis is discussed in more depth in section 2.1.5.

2.1.2 Government Bonds

In this work, US government bonds are considered since they are the most liquid and present the most significant market. Bonds are another popular type of security. Bonds are issued for a limited amount of time, and the issuer has to pay back the debt to the lender while paying interest during the lending period.

$$\begin{aligned} BondPrice &= \frac{C}{(1+r_1)} + \frac{C}{(1+r_{1.5})^{1.5}} + \frac{C}{(1+r_2)^2} + ... \\ &= \sum_{i=1}^{n} \frac{C}{(1+r_i)^i} + \frac{C}{(1+r_{i+0.5})^{i+0.5}} \end{aligned}$$

$$\begin{aligned} C &= \text{Coupon} \\ r_i &= \text{Discount rate i years from now} \end{aligned}$$

This formula is similar to the stock share valuation. The only difference is that the time frame for a bond is limited. Treasury bills, Treasury notes, Treasury bonds, and Treasury Inflation-Protected Securities (TIPS) are the four kinds of debt the US government issues to finance its spending. Treasury bills are short-term debt obligations with less than one year of maturity. Treasury notes are debt obligations with a duration of one to ten years. These kinds of bonds were used in this work. Treasury bonds are securities with a higher duration than treasury notes. All of these bonds are issued at a so-called issue date and end on a maturity date. Each bond is issued with a yield. The yield is the annual interest that the bond pays. Figure 2.1 shows the development of the 10 year treasury note yield over the last 30 years. US bonds typically pay interest twice a year.

2.1.3 Efficient Market hypothesis

The efficient market hypothesis is a hypothesis in the financial literature that states that the current price of the security reflects all available information. The idea of this hypothesis is that all current information is incorporated in the price because if it wouldn't be people would trade on it and move the price to the efficient price. This means that if a market is efficient, it is impossible to predict future prices according to the hypothesis [3].
For this, markets need to be efficient, and the following conditions should be met:

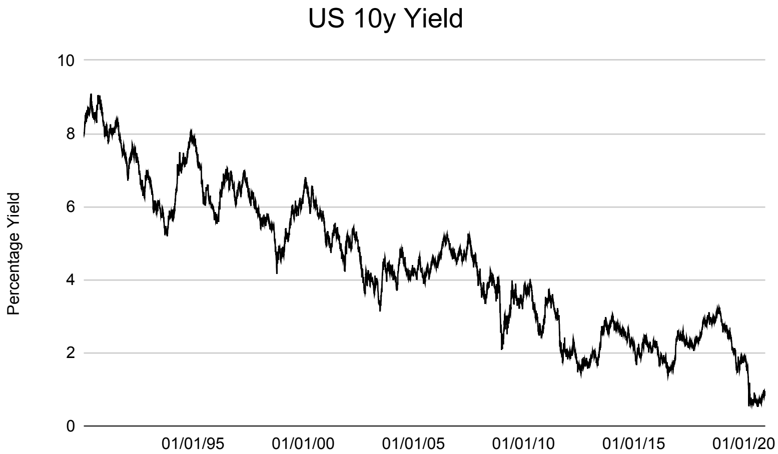


Figure 2.1: 10y yield

- Market participants obtain all necessary information regarding the relevant security.
- There is enough liquidity in the market. This means that it is possible to sell all assets for cash at any moment.
- There is low market friction cost. Trading fees are low enough to not provide a barrier to entry.

Because we assume that all participants have the same information available an equilibrium price is reached. This equilibrium is stable. Any new information is immediately incorporated into the price of the security. Therefore financial markets aggregate all available information into a single price according to the efficient market hypothesis.
At any point in time all information is available in the present price and therefore past prices are irrelevant for future price changes. This means the price changes behave in a random walk fashion. Specifically these random walks are Markov Processes since any further change only depends on the current state.
The connection between asset price and random walks can be traced back to the french mathematician Louis Bachelier [4] in 1900. His work was rediscovered in the 1960s by Eugene Fama and others. The contribution of Fama was that he stated three forms of efficient markets [5].

- *Weak Form:* Is also known as random walk theory and states that future prices are not influenced by past prices. It assumes that all currently available information is incorporated into the price.

- *Semi-Strong Form:* This form assumes that stock prices adjust fast to newly available information in the market. It suggests that neither technical nor fundamental analysis can be used to outperform the market. Only non-public information can be used.

- *Strong Form:* This is the most strict form of the efficient market hypothesis and states that all information is incorporated into the current price. Public and private information is included.

The efficient market hypothesis isn't without critics. In 2013 Schiller was awarded the noble prize in economics for showing that markets aren't efficient. Interestingly Fama also received his noble price in 2013 with Schiller for his contribution to the efficient market hypothesis [6].

2.1.4 Random Walk

The random walk theory is based on the assumption that log returns 2.2 are independent and normally distributed [7]. In comparison to the normal return the log returns are additive. This property is shown in 2.3 where the k period return is the sum of one period returns. We can now rearrange the log return formula and form the random walk model 2.4. This includes the assumption that the returns are Gaussian distributed $r_{i,t} \sim \mathcal{N}(\mu, \sigma^2)$. If we set an arbitrary starting point we have a geometric random walk. Figure 2.2 shows an example of such a process and one can see the similarities to a stock market ticker. In this case an initial Price of $P_0 = 100$ and the following Gaussian distribution $r_{i,t} \sim \mathcal{N}(\mu = 0, \sigma^2 = 0.03)$ are used.
Figure 2.3 shows the time series plot of the log returns of the S&P 500 over the last 25 years. One can see that the return are nearly uncorrelated and there exist clusters of volatility around certain events. To further show that there isn't much correlation between returns figure 2.4 plots the autocorrelation for the last ten days. For all delays the correlation is very small. This empirical evidence leads to the conclusion that stock markets indeed follow a nearly random pattern.

$$r_t = \log \frac{P_t}{P_{t-1}} \tag{2.2}$$

$$r_{k,t} = r_{1,t} + ... + r_{1,t-k+1} \tag{2.3}$$

$$\frac{P_t}{P_{t-k}} = \exp(r_{1,t} + ... + r_{1,t-k+1}) \tag{2.4}$$

$$P_t = P_0 * \exp(r_{1,t} + ... + r_{1,1}) \tag{2.5}$$

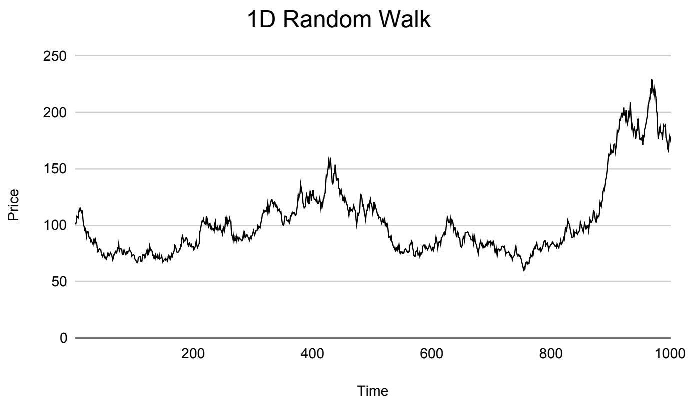


Figure 2.2: Geometric Random Walk

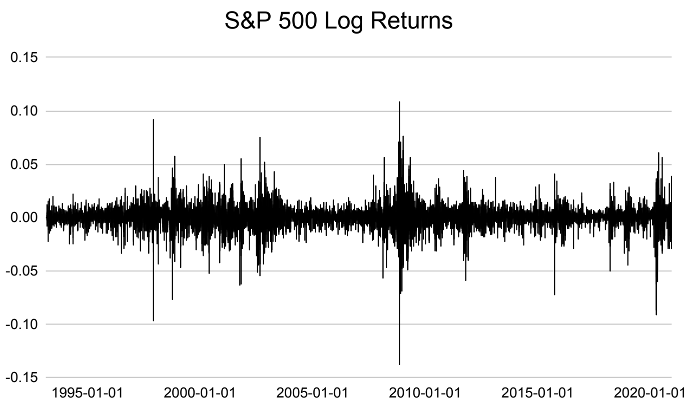


Figure 2.3: Log Returns

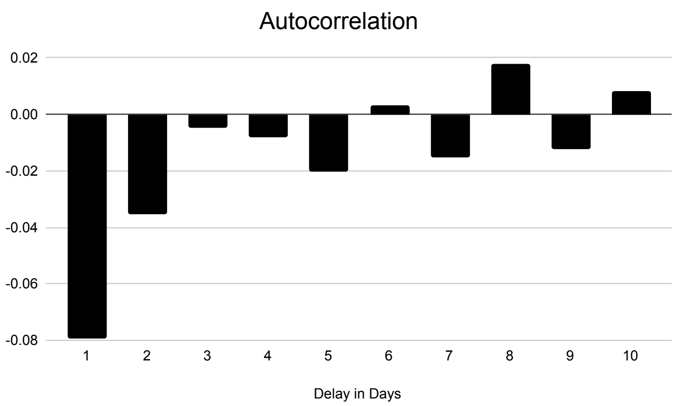


Figure 2.4: Autocorrelation

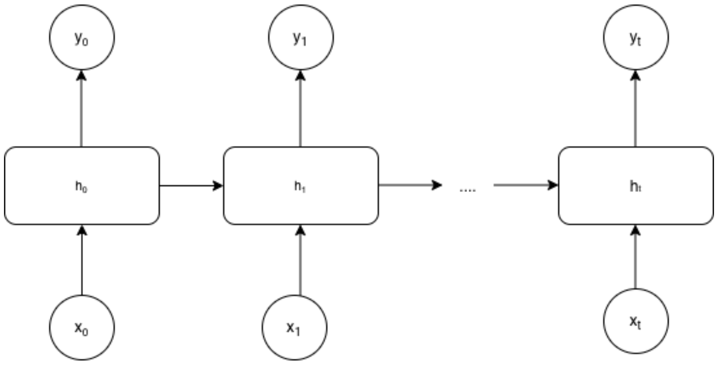


Figure 2.5: Recurrent Neural Network

2.1.5 Technical Analysis

Technical Analysis is a methodology to forecast future prices according to previous prices. The core assumption of technical analysis is that all fundamental data is already incorporated in the price. To predict future prices only trends and auxiliary technical data are used. If one assumes that the efficient market hypothesis holds technical analysis should not provide any useful results. This conclusion is supported by current studies on technical analysis, which provide mixed results [8].

2.2 Neural Networks

Neural Networks are inspired by the human brain. The idea is to connect many neurons and train the network of neurons on training samples. The most common neural network is a dense network where different layers of neurons are connected to form a DAG. These simple networks are able to approximate any function [9]. In this work, a special type of neural network was used, namely recurrent neural networks. The goal of recurrent neural networks is to capture time dependence in the data. This is especially useful when dealing with time series data.

2.2.1 Recurrent Neural network

Recurrent neural networks (RNN) are a special neural network where the calculation of the output does not only depends on the input but also on the output of the previous unit. An RNN produces an output at each time step. The advantages of an RNN are that the network has some information about the previous output. The information from the last time step can then be used for further calculations. This characteristic is useful in time series data where the value of the previous time step is useful for predicting the next time step.

2.2.2 LSTM

Long short-term memory (LSTM) is a special kind of recurrent neural and were first introduced in 1997 [10]. The goal of LSTM networks is to capture long and short time dependencies in the data. Compared to a regular RNN the LSTM also has information about the prediction of cells prior to the previous cell. LSTM networks transfer this information in a so-called hidden state. This hidden state is a combination of short and long time dependencies.

Chapter 3

Existing Work

In the following section, we analyze the results from a 2017 paper which proposes a state frequency memory recurrent neural network model for stock market prediction [2].

3.1 State Frequency Memory

State Frequency Memory models are a special variant of recurrent neural networks. The goal of this model is to capture multi-frequency trading patterns in the stock data. It works by decomposing the hidden states into different frequencies to better predict different time horizons. Figure 3.1 shows the architecture compared to a RNN and a LSTM cell.

3.1.1 Method

The paper uses other state-of-the-art methods to compare the prediction results of the state frequency memory model. Namely, they used autoregression and LSTM networks to compare the results. These models are used as a baseline for the proposed state frequency memory model.
The data set consists of 50 US blue chips companies from 2007 to 2016. To train the model the first 80% of the data is used. The remaining samples are used for testing.
The objective of the model is to predict a certain number of time steps, in this case days, into the future. In the paper one, three and five-day prediction is evaluated. The performance of the model is evaluated according to the mean squared error on the test set.

3.1.2 Results

Table 3.1 shows the paper's result with the additional result from the last day prediction. The results show that their proposed state frequency model does not

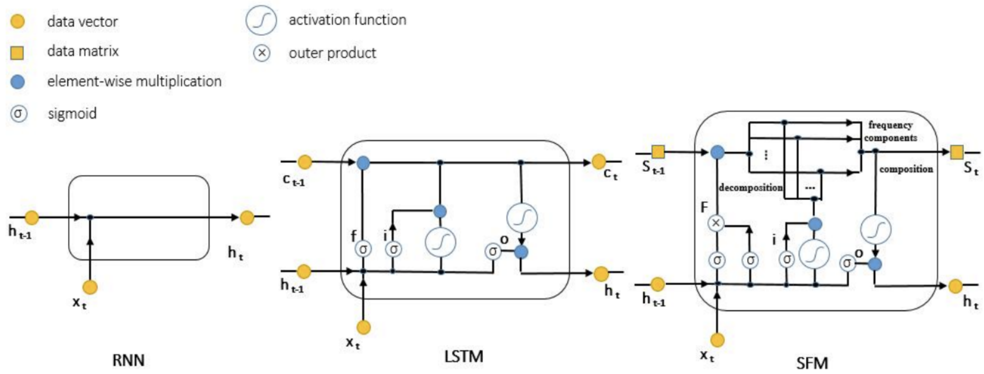


Figure 3.1: State Frequency Memory

	1-step	3-step	5-step
AR	6.01	18.58	30.74
LSTM	5.93	18.38	30.02
SFM	5.57	17.00	28.90
LAST DAY	5.50	16.77	27.58

Table 3.1: MSE

provide better performance than predicting yesterday's price for tomorrow. These findings align with our observation on bond data discussed in the next chapter and the discussion on the efficient market hypothesis in section 2.1.3. Since the state frequency model only looks at past prices, it is enough for the market to be weakly efficient. If the market is weakly efficient, the prices move in a random walk like fashion and therefore predicting the last day's price for tomorrow is the best guess. Overall predicting the price of the last day is better for all evaluated time step predictions. It is also better than both proposed baselines.
Figure 3.2 show the price prediction of the state frequency memory model for one day into the future. Only looking at the prediction of the model might lead to a false conclusion about the model's performance since the predictions look accurate. The right picture shows that predicting the last day leads to a similar, even better prediction graph.
The results from this work support the efficient market hypothesis since the model isn't able to outperform.

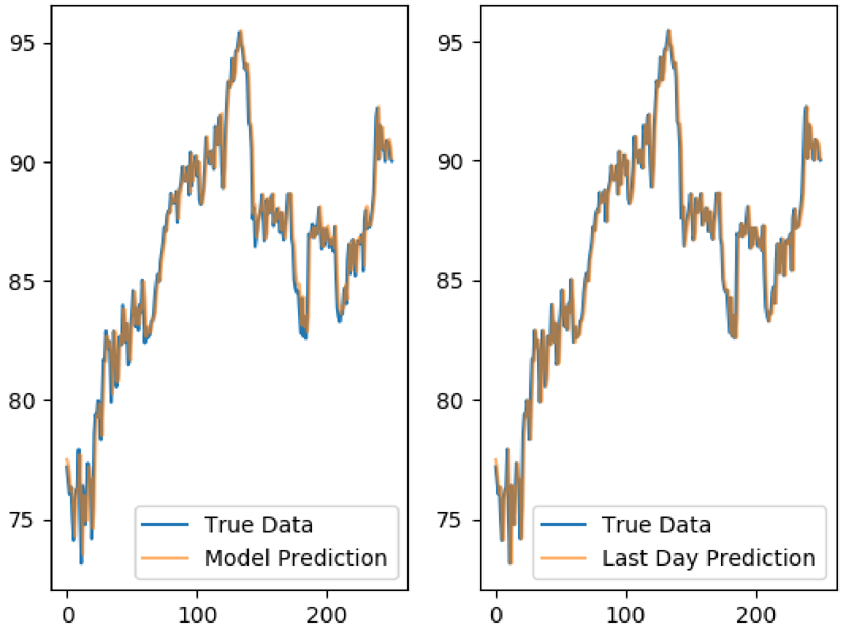


Figure 3.2: Prediction comparison

Chapter 4

Experiments

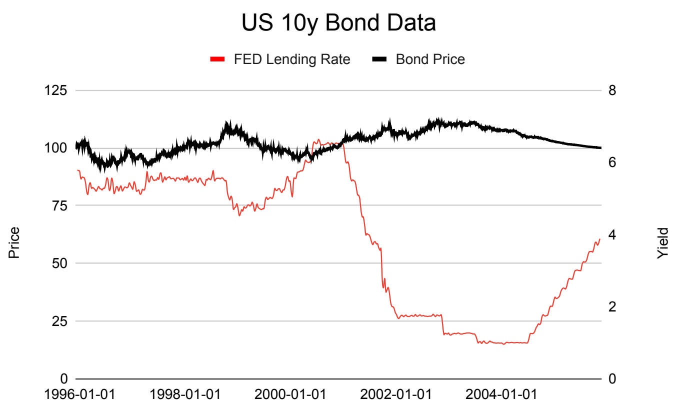
This chapter is about the US bond prediction model. The model is an LSTM model to predict US bonds. In our work, we tried to transfer the findings of [2] in the stock market to the US government bond market. The US bond market currently has a size of about 30 trillion dollars [11] and is larger than the stock market.

4.1 Method

As mentioned, the goal was to transfer the findings of prior work on the stock market to the bond market. For the experiments, US bond data was available from 1976 to 2018. This data included a variety of bonds with different maturity lengths. Each bond has price data for the time it is outstanding. Normally the bond price is calculated with formula 4.1. The coupon is determined by the yield of the bond and for the discount rate the FED lending rate is used. Figure 4.1 shows the price of a 10y bond and the FED lending rate during the duration of the bond.
The prediction method used is an LSTM network. This method is also used in the existing work. Table 4.1 shows the configuration used to train the LSTM model on the bond data. Furthermore, instead of predicting the price the goal was to predict the profit and loss value. The profit and loss value is a commonly used value in bond trading. By definition, the profit and loss is the difference in the value of the security between yesterday and today. Formula 4.1 shows the formula for calculating the PnL value. This has to be done because neural networks use normalized data. The reason for choosing the profit and loss value as a normalization is that if one can predict the PnL for tomorrow, it is easy to derive a profitable trading strategy.
Across our experiments, the last five PnL values were used to predict the PnL of tomorrow. This corresponds to the one-day prediction from the existing work.

Hidden state size	20
LSTM layers	10
Loss Function	MSE
Epochs	2000
Training Timeframe	01.05.1987 - 31.01.2008
Test Timeframe	01.01.2009 - 24.04.2018

Table 4.1: LSTM configuration



$$PnL = (p_t + i_t + c_t) - (p_{t-1} + i_{t-1}) \tag{4.1}$$

p_t = Clean Price at day t
i_t = Accrued interest at day t
c_t = Coupon payments if received any

4.2 Evaluation

To evaluate the model two main metrics were used. The first metric is the mean absolute error (MAE) because this metric can easily be interpreted. The second metric is an accuracy metric, which shows the percentage of correct price direction predictions. This accuracy metric is useful because it can be easily translated into a trading strategy. For example, an upward prediction for tomorrow can be used as a buy signal.
As a benchmark, the last day prediction was used. In the case of PnL prediction this means predicting a PnL of zero for tomorrow meaning no change in the value of the bond.

Duration	2y	2y	5y	5y	10y	10y
FED	false	true	false	true	false	true
MAE	0.248	0.379	0.348	0.4875	0.524	0.858
LAST DAY	0.232	0.232	0.123	0.123	0.118	0.118

Table 4.2: Results for LSTM model on bond prediction

Duration	2y	2y	5y	5y	10y	10y
FED	false	true	false	true	false	true
ACC	0.7257	0.64217	0.6511	0.608	0.587	0.623
LAST DAY	0.799	0.799	0.803	0.803	0.812	0.812

Table 4.3: Accuracy for LSTM model on bond prediction

4.3 Results

Table 4.2 and 4.3 show the results for the LSTM model. The mean absolute error is shown by table 4.2 and the accuracy is shown in table 4.3. The top two rows describe the input data used. Duration describes the maturity of the bonds used and FED indicates if the FED lending rate was used as an additional data point. The results show that the LSTM model does not perform well when compared against the benchmarks. There are a few possible explanations for this. As shown in section 3 other work suffers from similar issues and can't beat the benchmark. Predicting the last day is consistently better in both the existing work and in our work. This leads to the question of why that last day is such a good prediction for tomorrow. The stock market and the bond market are very liquid and well-researched markets. If one assumes that the efficient market hypothesis, which is discussed in section 2.1.3, is true, one can argue that the US government bond and stock market are efficient. Since our model and the model of the existing work perform an advanced form of technical analysis they don't incorporate any information that isn't available to the market and therefore according to the EMH can not outperform on a lasting basis.
If the results showed that you could outperform the bond market, it would render the efficient market hypothesis false.
Another observation is that adding additional information like the FED lending rate does not help. It even worsens the performance of the model. A possible reason for this is that this publicly well-known information and probably already incorporated in the price. Therefore it only adds complexity without providing any useful information to the model.

Chapter 5

Rethinking Benchmarks for Financial Models

In this chapter, benchmarks for financial prediction model are discussed. As seen in the prior two chapter benchmarks are crucial in evaluating the performance of financial prediction models. Three key aspects are discussed in depth. First, the selection of the test set such that it accurately represents the market. Secondly, there is a discussion about loss function for training and their impact on model performance. Finally, some evaluation metrics are studied, which can improve the decision-making process about a developed model.

5.1 Test data

Testing is a central point in machine learning. A common practice is to use a test set. A test set is a fixed fraction of the data which is solely used for evaluating the model at the end. It is good practice to not use the test set to evaluate your model during model iteration. During the model development a validation set is used.
In the context of the financial market, it is important to use a representative test set. As discussed in section 2.1.4 stock market behavior follows a random walk pattern and there exist clusters of higher volatility. Therefore one needs to place close attention to the dataset. Due to this observation, we discuss different methods of choosing a dataset in the next section.

5.1.1 Data

To illustrate the differences in the possible datasets we analyze the three following markets. First the stock market where the data from the existing work 3 is used. This data includes historical prices from 2007-2016 for 50 blue chips US stocks. The second market is the cryptocurrency market, where the bitcoin price is used from 2014-2020. The third market is the gold market from 2000-2020. To analyze the data, it is split into quarters such that different configurations can be tested. The following configurations are analyzed

- *Regular:* This is the regular form which is also the most common. Just use the last 20% of the dataset as test dataset. The evaluation results with this method are really dependent on the current time. For example using the last four quarters from the current time can lead to abnormal test set. This is visible when looking at the log returns in figure 2.3.
- *Random:* This is another method to choose a dataset. In our case 20% of the quarters are randomly selected. This can lead to a better distributed test set. A potential drawback is that it is possible to test on old data which might be outdated. An example for this would be major policy changes like the abandoning of the gold standard.
- *Last days of quarter:* The goal of this approach is to use the last few days of a quarter as a test set. Therefore the test set is distributed across the entire data. This should lead to better distribution since the test set includes data from all quarters in the dataset.

Table 5.1 shows the standard deviation and the normalized entropy for different test set configurations. The standard deviation is calculated across the entire data. For the normalized entropy calculation the formula 5.1 is used.
The evaluation shows that characteristics of test and training set can be vastly different between different methods. In general using the last days of each quarter as a test set leads to the most balanced train and test set according to our evaluation.

$$\eta(X) = \frac{H}{H_{max}} = -\sum_{i=1}^{n} \frac{p(x_i)\log(p(x_i))}{\log(n)} \tag{5.1}$$

5.2 Loss functions

An integral part of training neural networks is the loss function. The loss function is the objective of the optimization and takes the prediction and the ground truth as an input and outputs a real value. During training, a neural network is optimized such that the value of the loss function is minimized. Therefore the choice of the loss function can have a large influence on the behavior of the model. To evaluate the influence of the loss function on the performance, we trained the stock prediction model described in section 3 with different loss functions.

	Stocks		BTC		Gold	
	Train	Test	Train	Test	Train	Test
STD						
Regular	56.1	121.9	3622	1896	465	219
Random	81.3	39.5	4213	4375	477	431
Last days of quarter	75.1	76.3	4234	4588	476	472
Entropy						
Regular	0.976	0.957	0.917	0.996	0.982	0.998
Random	0.970	0.978	0.946	0.881	0.985	0.980
Last days of quarter	0.971	0.967	0.936	0.922	0.985	0.982

Table 5.1: Dataset metrics

- *Mean Squared Error:*

$$L(y, f(x)) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \tag{5.2}$$

- *Mean Absolute Error:*

$$L(y, f(x)) = \frac{1}{n}\sum_{i=1}^{n}|y_i - f(x_i)| \tag{5.3}$$

- *Huber Loss:*

$$L_\delta(y, f(x)) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases} 0.5 * (y_i - f(x_i))^2, & |y_i - f(x_i)| \leq \delta \\ \delta * |y_i - f(x_i)| - 0.5 * \delta^2, & otherwise \end{cases} \tag{5.4}$$

Figure 5.1 shows the mean squared error on the test data for four models trained with different loss functions. The results are similar between the different loss function but the MSE and the huber loss with a 0.1 delta perform best. A possible explanation for this is that these loss function match the distribution of the data, which is discussed in section 2.1.4.

5.3 Evaluation

Evaluating the model is an integral part of the model development process. To evaluate the performance a test set is used. Datapoints from the test set are not used during training. Therefore these are unseen examples for the model. The evaluation is usually done with evaluation functions to measure the performance. In the following section, different evaluation functions are discussed according to their advantages and disadvantages.

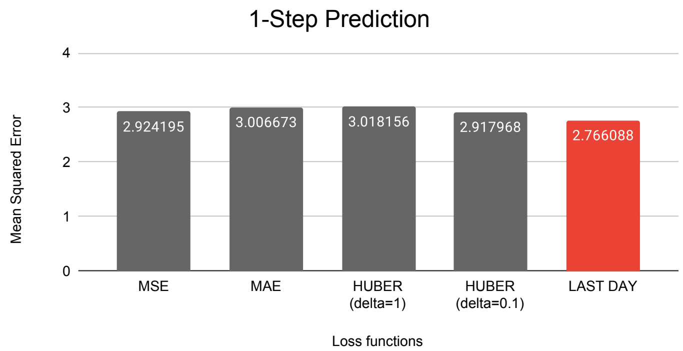


Figure 5.1: 1 Day Prediction Error

- *MSE:*
$$E(y_{pred}, y_{true}) = (y_{true} - y_{pred})^2 \tag{5.5}$$
This is a often used evaluation metric in finance since it relates to the distribution of the data and is simple to report. The lower the mean squared error the better the model. A score of 0 indicates a perfect model. A key characteristic is that bad predictions get a large error due to the square function.

- *MAE:*
$$E(y_{pred}, y_{true}) = |y_{true} - y_{pred}| \tag{5.6}$$
The advantage of this error is that it is easy to interpret. For example if the model predicts the stock price it can be translated to a monetary value.

- *Relative Error:*
$$E(y_{pred}, y_{true}) = \left|\frac{y_{pred} - y_{true}}{y_{true}}\right| * 100\% \tag{5.7}$$
The advantage of this error is that it is easy to interpret. It is measurement of precision and the error is relative to the actual value.

- *R2 Squared Error:*
$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{5.8}$$
R^2 is the normalized version of the MSE. Because it scales the result it makes it easier to compare different models across asset classes. For example a model predicting a asset class with a higher average price most likely has a higher MSE than a model predicting an asset with a lower priced assets. R^2 makes it possible to compare the models across asset classes.

- *Trading Strategy:* This is a more complex approach since it involved generating a trading strategy from the model which might be an involved process. The resulting trading strategy would then be tested and evaluated according to the return generated.

Chapter 6

Conclusion

This work explores prediction in the financial market. The main conclusion of this work is that it proves to be difficult to predict financial markets, especially in efficient markets. Results from our experiments on the bond market and results from existing work on the stock market show that these markets indeed appear to be efficient. To make money by predicting the financial market it is crucial to find inefficiencies. This can be done by incorporating not yet known information or by being faster than other trade participants.

The most promising direction for further work is to research smaller more niche markets with fewer participants. These markets might be less efficient and therefore it could be possible to implement successful trading strategies. Another possible direction is to make use of nonconventional information. This method is used in practice by certain investors [12].

Bibliography

[1] "The stockmarket is now run by computers, algorithms and passive managers," *The Economist*, Oct 2019.

[2] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 2141–2149.

[3] S. Lera and D. SORNETTE, "Constrained random walk models," Ph.D. dissertation, Master thesis, ETH Zurich, 2015.

[4] L. Bachelier, "Théorie de la spéculation," in *Annales scientifiques de l'École normale supérieure*, vol. 17, 1900, pp. 21–86.

[5] E. F. Fama, "Random walks in stock market prices," *Financial analysts journal*, vol. 51, no. 1, pp. 75–80, 1995.

[6] "The sveriges riksbank prize in economic sciences in memory of alfred nobel 2013." [Online]. Available: https://www.nobelprize.org/prizes/economic-sciences/2013/press-release/

[7] D. M. Dettling, "Statistical analysis of financial data," January 2017.

[8] C.-H. Park and S. H. Irwin, "What do we know about the profitability of technical analysis?" *Journal of Economic Surveys*, vol. 21, no. 4, pp. 786–826, 2007.

[9] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7, 2001.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] "Federal reserve bank of dallas." [Online]. Available: https://www.dallasfed.org/

[12] F. Partony, "Stock picks from space," *The Atlantic*, 2017.