Dylan Johnston | 1003852690
Masters of Engineering | Aerospace
University of Toronto | UTIAS

Predicting the Market

ECE1513
Introduction to Machine Learning
Wednesday April 10th

Contents

1 Introduction 1

2 Theory & Model 1

2.1 Financial Theory 1
2.2 Multivariate Gaussian Mixture Model 2
2.3 Support Vector Machines 2
2.4 Long Short-Term Memory Networks 3

3 Code Architecture 3

4 Results & Discussion 4

4.1 GMM 4
4.2 SVM 4
4.3 LSTM 5
4.4 Discussion 6

5 Future Work 7

6 Conclusion 7

List of Figures

1 Trends identified by the GMM unsupervised learning model colour coded on predicted data points. 4
2 SVM algorithm strategy return vs market return. 5
3 Accuracy of LSTM Model. 5
4 Loss of LSTM Model. 5
5 LSTM algorithm strategy return vs market return. 6

List of Tables

1 Averages and covariances of each trend identified by the GMM unsupervised learning model. 4

Abstract

The use of machine learning algorithms for the purpose of stock market price predictions was investigated. Aggregate, time series data was retrieved from Yahoo Finance, and several technical trading indicators were calculated and appended to the data set. A Gaussian Mixture Model was applied to the cumulative returns of the data set over time in order to identify different trend regimes that can be used as a feature for predictive algorithms. A Support Vector Classifier model designed to output a buy (long) or sell (short) signal was created, and the cumulative return given by these signals was compared to the cumulative return of the underlying security over the same period. Finally, a state-of-the-art Recurrent Neural Network model implementing Long Short-Term Memory neurons was implemented, utilizing time series financial information from the previous 50 trading days in order to predict the closing price of the security on the 51st day.

1 Introduction

Since its inception in 1956, the field of artificial intelligence has expanded into almost every sector in today's world. From speech and image recognition, to increasing agricultural crop yield, to medical diagnostics, to wide scale automation, AI has been used to optimize processes in new and innovative ways that are drastically changing the world around us.

In this paper, several machine learning algorithms are applied to financial data obtained from Yahoo Finance. Section 2 provides a brief introduction to the methods being used in this investigation, as well as some background information concerning the various financial metrics appended to the data set and used as features. Section 3 describes the architecture of the python script, while the results that were obtained using this script are provided in Section 4. A brief discussion of the results, and a comparison of the various machine learning methods implemented in this paper is included in Section 5. Finally, Section 6 provides some important next steps for improving this work.

2 Theory & Model

This section contains a brief amount of background information concerning the various financial metrics used as input features to the three machine learning algorithms used in this investigation, as well as some information concerning the machine learning algorithms themselves. Due to the brevity of this investigation (5 pages), the full analytic derivations of the various financial metrics and ML algorithms will be omitted, however, several resources will be included in the references section.

2.1 Financial Theory

The data sets that were used for this investigation were obtained from Yahoo Finance. These data sets provide the daily 'Open', 'High', 'Low', 'Close' (OHLC) and 'Adjusted Close' prices, as well as the daily volume traded, for the desired security. These metrics have some predictive capability on their own, however, since the purpose behind this investigation is to create a trading algorithm, several more advanced metrics were calculated and appended to the data set. These metrics are detailed below.

- **Periodic Returns** - From the initial features imported from Yahoo Finance, 'Close' prices were used to calculate the daily, weekly, and monthly changes in the price of the underlying security as a ratio of the current 'Close' price over the 'Close' price from the beginning of the period (1 day prior, 5 days prior, or 20 days prior respectively). The log of this ratio was then taken, since share prices have been determined to be log-normal.

- **Momentum** [17] - From the 'Close' prices, the momentum of the underlying security was also calculated. Momentum is an empirically observed tendency for rising asset prices to rise further, and falling asset prices to fall further, in real markets.

- **Bollinger Bands** [7] - One type of volatility based trading indicator called Bollinger bands use the arithmetic mean and standard deviation of a security's price in order to create a predicted movement channel within a security's chart. When the price moves away from the average, a short or long position can be taken and a profit can be generated as the underlying trends back towards it's average. The two parameters that can be controlled by the user for this metric are the number of periods used to determine the average and standard deviation, and the number of standard deviations used to determine the width of the channel. The standard periods = 20 and standard deviations = 2 were used for this investigation.

- **Moving Average Convergence Divergence (MACD)** [8] - The MACD indicator is a trend indicator used in technical trading. It utilizes two different exponential moving averages, and the difference between these two averages is known as the MACD, while the 9 day exponential moving average of the MACD itself is referred to as the MACD signal line. Additionally, a third indicator called the MACD crossover can be used, which is the difference between the MACD and the MACD signal line. The theory behind how the MACD and its derivatives are used is outside the scope of this paper, so suffice it to say these indicators can identify changes in trends, and thus opportunities to enter a bearish or bullish position.

- **Relative Strength Index (RSI)** [2] - The RSI indicator is a momentum oscillator which measures the velocity and magnitude of directional price movements. While several momentum oscillators exist, the RSI computes momentum as a ratio of higher closes to lower closes. The scale is from 0 to 1, where values close to 1 indicate the underlying has strong upward momentum, while values close to zero indicate stron downward momentum. Two RSIs were appended to the data set, one spanning 10 days (2 weeks), and the other spanning 1 month (20 days).

- **Chaikin Oscillator** [15] - The Chaikin oscillator is a volume oscillator that calculates the position of a security's closing price as a fraction of the daily price range of the security. This fraction is then multiplied by the daily volume in order to quantify the daily price range of a security. This volume oscillator provides technical traders with an indication that a price reversal is about to occur, and can indicate opportunities to enter into a profitable long or short position.

2.2 Multivariate Gaussian Mixture Model

Gaussian Mixture Models (GMMs) are a form of unsupervised learning that excel at identifying subpopulations within an overall population. GMMs can be used to group data points together in a data set based on how similar their features are in the sense of a Gaussian distribution. The following are components of a GMM:

- Number of samples inherent to the data set, 'N', and number of subpopulations within the population of samples, '$\mathcal{K}$'

- '$\mathcal{K}$' sets of parameters $\boldsymbol{\theta}$ associated with each component of the mixture. In the case of GMMs, these are the mean and covariance matrix for each Gaussian within the population $\theta_{k \in \mathcal{K}} = \{\mu_k, \Sigma_k\}$.

- A set of '$\mathcal{K}$' weights, 'ϕ_k' which must sum to 1, where each weight is the prior probability of a particular mixture component.

- A set of 'N' random latent variables, which identify the mixture component of a given observation

In this investigation, a GMM will be used to attempt to identify different trend regimes within the data set. The number of Gaussian components is varied, however it is assumed that all Gaussians have a spherical covariance shell. The Gaussian model is used in two ways for this investigation. First, it is trained on a portion of the data set, and the trend regime for the remaining data points is then predicted. This reduced data set is then used for the prediction stages. Second, the GMM is fit on the whole data set, and included as a feature for the whole data set to be trained on by the prediction stages. This is so that the prediction stages have a more extended data set to be trained on.

2.3 Support Vector Machines

Support Vector Machines (SVMs) are supervised learning models typically used for classification and regression analyses. The SVM algorithm involves mapping every point from the data set into a vector space such that the examples belonging to distinct categories are divided by a clear gap who's width has been maximized. New examples are subsequently mapped into the space and categorized based on which category they belong to according to the fitted separation gap.

SVMs are effective at non-linear classification due to a tactic known as the kernal trick. This strategy involves mapping the data set into higher dimensional feature space that allows the algorithm to fit a maximum-margin hyperplane, which can appear nonlinear in the original input space. One draw back from this trick is that working in higher dimensional feature space typically increases the generalization error of SVMs, depending on the number of training samples.

Several kernals are available to an SVM depending on the nature of the data set it is modeling. For the purpose of this investigation, the scope of kernal selection was restricted to radial basis functions. Several other hyperparameters in this model may be tuned, however, the scope of parameter tuning was outside the scale of this investigation.

2.4 Long Short-Term Memory Networks

One unique feature of financial data sets is that data samples within the set are typically organized chronologically. This provides an interesting opportunity: perhaps periodic or reoccurring trends can be discovered by analyzing data sequentially.

Recurrent Neural Networks (RNNs) are a type of artificial intelligence architectures that allow for nodes within the network to process sequences of inputs, by including past elements of the sequence as input to future elements. Fully connected RNNs suffer from a drawback known as the vanishing or exploding gradient problem, whereby during gradient descent optimization, the gradients used during back propagation approach zero or infinity respectively, causing the training time to increase drastically or the model to fail to ascertain any underlying trends.

One of the most useful AI architectures available today bypasses the vanishing gradient issue entirely. LSTM networks have feedback connections which make it a general purpose computer, allowing it to compute anything that a Turing machine can. LSTM cells are composed of 4 main components: a cell, an input gate, an output gate, and a forget get. These gates work in unison to allow a cell to retain past information for an arbitrary length of time, allowing the model to identify long term dependencies of later time series data on past time series data.

In this investigation, the structure of the LSTM model will be restricted to 2 LSTM layers with 100 nodes in each layer. The structure of the net, the number of epochs trained, the cost function used, drop out rates for each layer, the look back window for each sample, the prediction window for each prediction, the type of activation function used by each layer, and the input features are all hyperparameters that can be tuned within this model. Hyperparameter tuning will not be performed due to the scale of this investigation,

3 Code Architecture

In this section the general layout of the python script used to implement the above mentioned machine learning algorithms is described.

- **Data Retrieval and Augmentation** - Before the script is run, the desired security to be used for analysis can be entered within the function ’*main()*’, along with the desired time frame. It Additionally, hyper parameters of the various models can be found in the function ’*globalVariableDefine()*’. When the script is run, the function ’*dataImport()*’ retrieves the financial data from Yahoo Finance and saves the file as a ’.csv’. **It should be noted that occasionally ’*fix˙yahoo˙finance*’ has some sort of timeout error, and the kernal should be restarted so that the most current data set can be downloaded each time**. The data is then passed to the function ’*processData()*’, which creates a shifted version of the OHLC data, offset one day ahead so that the previous day’s OHLC are available for predicting the direction a security will move the subsequent day. Additionally, the function ’*augmentData()*’ calculates the various financial metrics outlined in Section 2.1 and appends them to the data set.

- **GMM Regime Prediction** - The augmented data set is fed to the function ’*regimeIdentify()*’. This function scales the columns of the data set, then creates a GMM used to identify different trend regimes. The purpose of this AI algorithm is to create a new input feature that is correlated to different directional trends.

- **SVM Model** - The unscaled data set with the newly appended regime feature is then passed to the function ’*SVMpred()*’. The data is scaled, redundant features are dropped from the data, and the classifier is fit using a portion of the data set. The SVM model is then tested on the remainder of the data set, and the cumulative return of the buy and sell signals generated by the SVM model are compared to the cumulative return of the underlying security in a buy and hold trade.

- **LSTM Model** - The unscaled data set is passed to the function ’*LSTMpred()*’. The data is passed to the function ’*processLSTM*’ where it is scaled and appended into train and test data sets. The LSTM model is then built buy the ’*buildModel()*’ function, and the model is fit to the training set. The model is then passed to the function ’*getScore()*’, where the model is tested on the remainder of the data set, and the cumulative returns of the buy and sell signals generated by the LSTM model are compared to the cumulative return of the underlying security in a buy and hold trade.

4 Results & Discussion

4.1 GMM

The GMM model attempting to find trends within the data set was optimized several times, and the number of trends that best aided the predictive capacity of both the SVM architecture and LSTM architecture was 4 trends. Figure 1 shows this optimization, and table 1 summarizes the mean and variance of each trend.

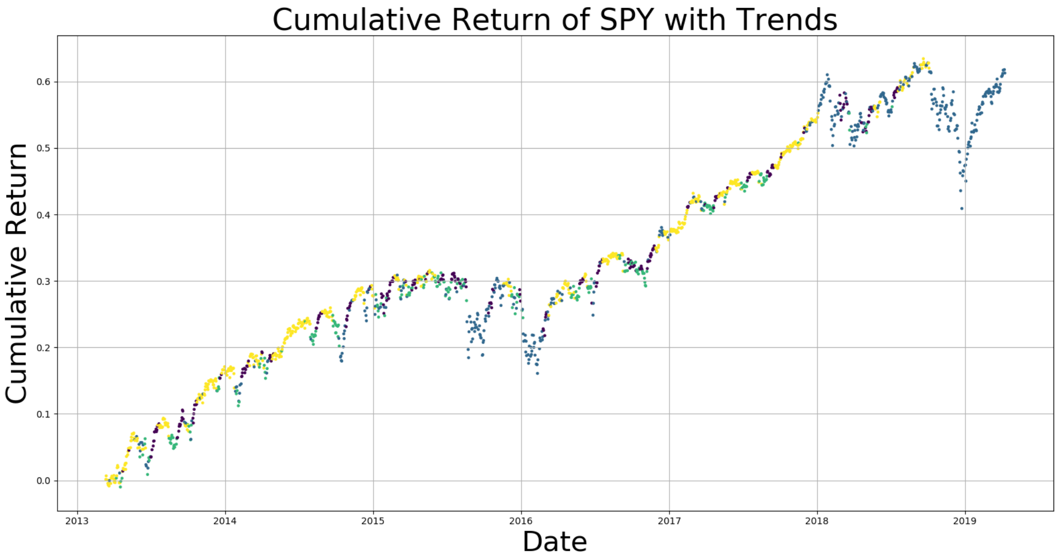

Figure 1: Trends identified by the GMM unsupervised learning model colour coded on predicted data points.

Trend	Mean	Covariance
Trend 1	0.248	0.139
Trend 2	0.210	0.973
Trend 3	0.236	0.102
Trend 4	0.253	0.068

Table 1: Averages and covariances of each trend identified by the GMM unsupervised learning model.

4.2 SVM

The SVM model was optimized on the data set containing the added trend feature from the GMM. The return of the SVM algorithm versus the market return is plotted below.

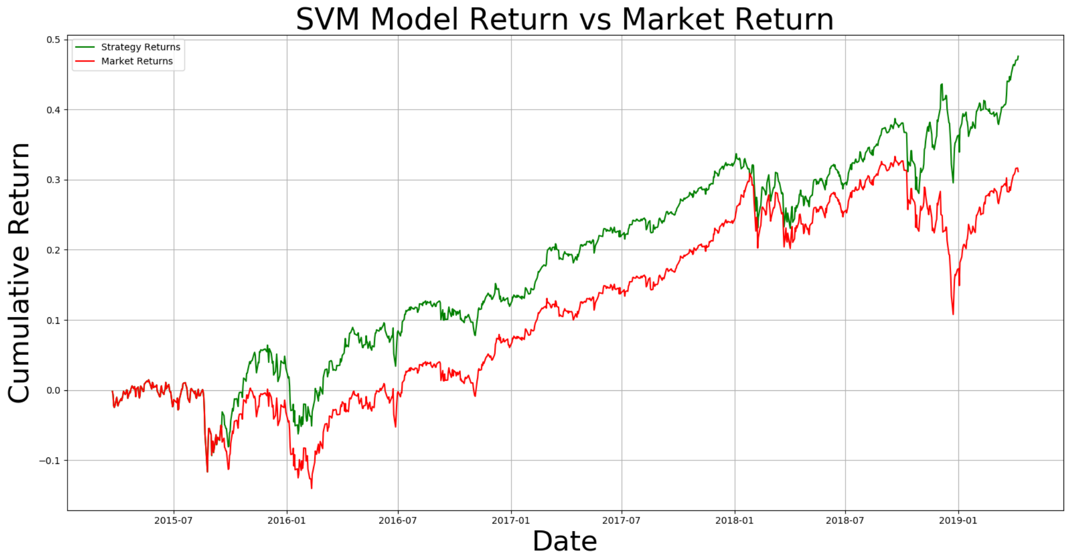

Figure 2: SVM algorithm strategy return vs market return.

The resulting accuracy of the SVM model was a mere 11.28%, however, it can be seen that this model still beats the return generated by the market.

4.3 LSTM

The LSTM model was trained over 500 epochs, using 2 layers with 50 nodes in each layer, with a dropout layer after each LSTM layer to help with generalization. The dropout rate selected was 0.2. This model uses a look back period of 50 days to predict the subsequent day. Figures 3 and 4 show the training and validation accuracy and loss respectively for this set of hyperparameters, while Figure 5 shows the return generated by the LSTM model versus the market return.

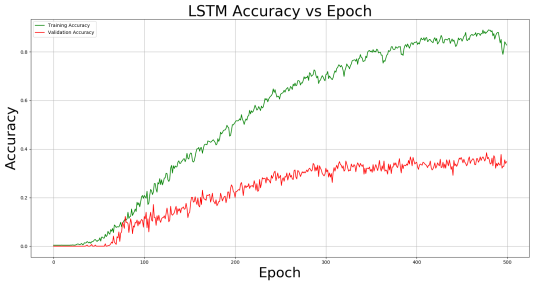

Figure 3: Accuracy of LSTM Model.

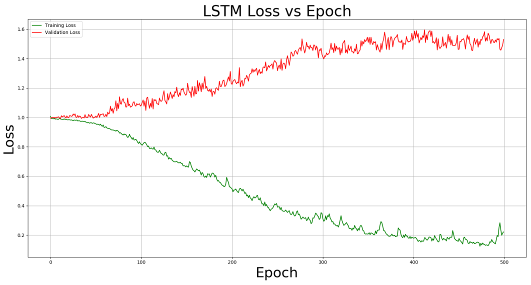

Figure 4: Loss of LSTM Model.

The LSTM model had a final accuracy of almost 70% on the training set, but only 34% on the validation set. It can be seen in Figure 5 that the model does indeed beat the market at the beginning, however, a better return than the market is much easier to achieve when the market is in a down trend, since in an uptrend, any mistakes sets you behind the market.

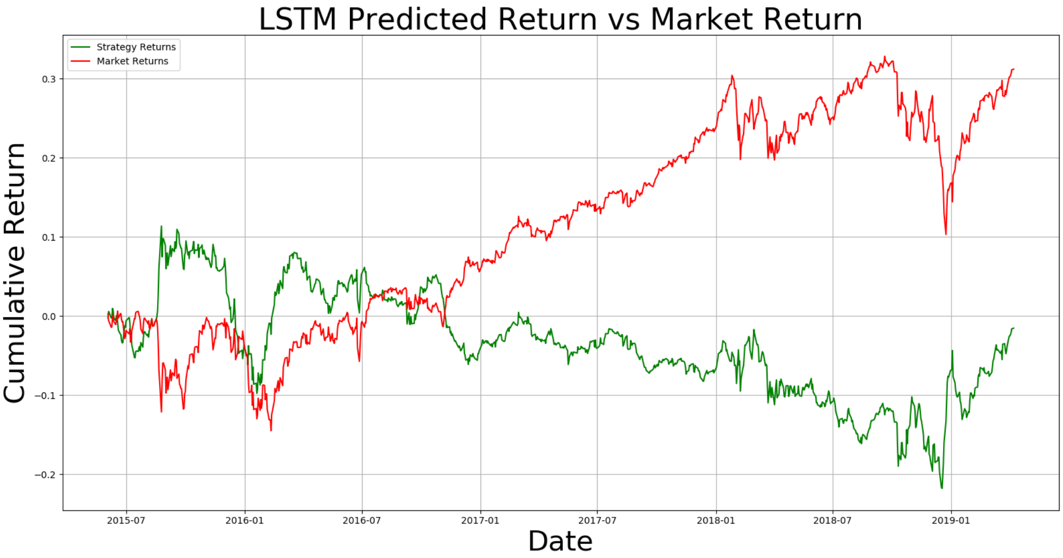

Figure 5: LSTM algorithm strategy return vs market return.

4.4 Discussion

In this investigation, machine learning was used in two separate capacities. First, unsupervised learning was used to attempt to discover trends within the data set. This application of machine learning is very beneficial for predictive purposes, since it effectively adds an additional feature to the data set. Theoretically, this type of optimization could be run with other valuable financial metrics. In this analysis, only technical trading indicators were used as input features, along with scaled prices and volume. This excludes many fundamental financial metrics which can be very useful indicators of the directional change in a securities price. For example, if the price to earnings ratio for a single company over a period of time is input into the data set, fluctuations in this metric could be correlated to changes in price as the market realizes a security is either under or over valued.

Further, if a number of different securities were input into a unsupervised learning model, the model may be able to identify which ones are over or under valued by comparing their various financial metrics to other companies within the same sector. Finally, based on the distributions generated by the GMM, data sets can be extended by creating data points that have features created by the normally distributed Gaussians. This allows for additional data to be generated, potentially increasing the predictive capabilities of the model.

The second capacity in which machine learning was used within this data set was in its attempt to predict subsequent changes in price of the underlying security by analyzing past data. Two different types of algorithm were used for this purpose: SVMs and an LSTM RNN.

These two different algorithms have different methods of predicting subsequent data points. SVMs model the data set by analyzing each data sample individually, mapping them into a potentially higher dimensional space and allowing for a hyperplane to differentiate between samples where the features indicate whether a long or short position should be taken. They do not consider data in a time series fashion, meaning that data points entered sequentially are not considered for their trends. This is a disadvantage in financial market prediction, since many technical trading indicators depend on recent changes in making predictions. MACD is a good example of this.

LSTMs on the other hand manage this task easily, retaining short or long term information within their cells, and being limited only by the look back window chosen by the data scientist to discover these underlying trends. Since financial markets depend heavily on these underlying trends, optimizing an LSTM network to find such trends allows for these networks to predict in a way that SVMs are unable to compete with. It should however be noted that while SVMs create a similar model every time, the optimized model and thus generated return of the LSTM network varies even if all hyperparameters are kept constant. Thus, LSTM models should be run a number of times, and the average results should be used as an indication.

It is clear that these two different ML architectures are each successful in their own way. LSTM RNNs are adept at analyzing time series data, and thus should be applied to technical trading indicators in order to discover opportunities to make money while a security is trending. As such, LSTMs could be used for buy and hold type predictions, whereby the underlying trends indicate that the security is likely to increase or decrease in value over a longer period. SVMs on the other hand are adept at separating data based on distinct features. This suggests an opportunity for SVMs to be implemented using volatility, volume and statistical information. SVMs can easily determine whether a security is undervalued or over valued compared to its current average price, and provide an architecture on which rapid day trading platforms can be built.

5 Future Work

The most important task that remains for this investigation is hyper parameter tuning. Due to the limited CPU and GPU available to the author, large scale parameter searches could not be performed within a reasonable amount of time. This includes varying which features are input into the models, as certain combinations of features may be better at predicting price movements than others. The hyperparameter tuning of these models is paramount to creating a platform that can trade with real money.

Once ideal parameters have been identified, the two models should be bootstrapped to ensure that they are truly successful models, and that the successful return is not some random luck.

Further, this model was trained on a single security. Some trends are more present in securities than others. If this model is to be trained on a single security, the model should be optimized to discover the specific trends intrinsic to that security. If, on the other hand, it was trained on a variety of securities, underlying trends within the market could be identified more readily. Specifically, the GMM unsupervised model might be better and identifying more complex trends, and thereby indicate opportunities to make money off of a trend more accurately.

Another area that this investigation did not touch on is trading strategy, instead simply buying at the open and selling at the close. A more sophisticated investment strategy for both the LSTM network and the SVM classifier should be developed to optimize entries and exits from positions, and thereby increase return.

Finally, markets are not ideal or predictable no matter what any financial advisor tells you. Markets are regularly affected by economic changes, news, and other unforseeable events. A sentimient analysis architecture could be added in order to more readily predict some of these events, and allow for an even larger profit to be achieved.

6 Conclusion

Data was retrieved from Yahoo finance, several technical trading indicators were calculated, and a GMM attempted to discover trends within the data as an additional feature column. LSTMs and SVMs were used to attempt to predict stock price directional change. Both methods seem to have unique niches within which to beat the market return.

References

[1] 10153181162182282. *Illustrated Guide to LSTM's and GRU's: A step by step explanation.* Sept. 2018. URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

[2] James Chen. *Relative Strength Index - RSI.* Mar. 2019. URL: https://www.investopedia.com/terms/r/rsi.asp.

[3] Thomas Fischer and Christopher Krauss. "Deep learning with long short-term memory networks for financial market predictions". In: *European Journal of Operational Research* 270 (Dec. 2017). DOI: 10.1016/j.ejor.2017.11.054.

[4] Rohith Gandhi and Rohith Gandhi. *Support Vector Machine - Introduction to Machine Learning Algorithms.* June 2018. URL: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[5] *Gaussian Mixture Model.* URL: https://brilliant.org/wiki/gaussian-mixture-model/.

[6] Alex Graves. *A. Supervised Sequence Labelling with Recurrent Neural Networks. Stud. Comput. Intell.* Uoft, 2008.

[7] Adam Hayes. *Bollinger Band.* Mar. 2019. URL: https://www.investopedia.com/terms/b/bollingerbands.asp.

[8] Adam Hayes. *Moving Average Convergence Divergence (MACD).* Apr. 2019. URL: https://www.investopedia.com/terms/m/macd.asp.

[9] J. B. Heaton, N. G. Polson, and J. H. Witte. Deep Learning in Finance[J]. arXiv, 2016. arXiv: 1602.06561.

[10] Sepp Hochreiter and Jfffdfffdrgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[11] C. M. A Hsu. "hybrid procedure with feature selection for resolving stock/futures price forecasting problems". In: *Neural Computing Applicaitions* 22 (2013), pp. 651–671.

[12] C. Kai, Y. Zhou, and F. A Dai. "LSTM-based method for stock returns prediction". In: *Proceedings of the 2015 IEEE International Conference on Big Data (Big Data).* CA, USA, 29 October-1: Santa Clara, Nov. 2015.

[13] W. Long, Z. Lu, and L. Deep Cui. "learning-based feature engineering for stock price movement prediction". In: *Knowl.-Based Syst* 164 (2019), pp. 163–173.

[14] Krishna Kumar Mahto and Krishna Kumar Mahto. *Demystifying Maths of SVM.* Jan. 2019. URL: https://towardsdatascience.com/demystifying-maths-of-svm-13ccfe00091e.

[15] Greg McFarlane. *How to Use the Chaikin Oscillator.* Apr. 2019. URL: https://www.investopedia.com/articles/active-trading/031914/understanding-chaikin-oscillator.asp.

[16] S. Minami. "Predicting Equity Price with Corporate Action Events Using LSTM-RNN". In: *Journal of Mathematics and Finance* 8 (2018), pp. 58–63.

[17] Investopedia Staff. *Momentum Indicates Stock Price Strength.* Mar. 2019. URL: https://www.investopedia.com/articles/technical/081501.asp.

[18] R. Q. Sun. "Research on Price Trend Prediction Model of US Stock Index Based on LSTM Neural Network". MA thesis. Capital University of Economics and Business, 2015.

[19] *Support Vector Machinesfffdfffd.* URL: https://scikit-learn.org/stable/modules/svm.html.

[20] Www.statsoft.com. URL: http://www.statsoft.com/textbook/support-vector-machines.