Stock Prediction with LSTM/SVR

Time-Series Financial Analysis with TensorFlow

Derrek Gass  
Engineering & Computer Science  
CSU-Sacramento  
 Sacramento, California USA  
 derrekgass@ecs.csus.edu

Alexander Lee  
Engineering & Computer Science  
CSU-Sacramento  
 Sacramento, California USA  
 alexwlee@ecs.csus.edu

ABSTRACT

There has been much development and interest in the application of machine learning to solve problems of pattern recognition in the analysis of signals such speech and images, and in recent years, has extended past this domain into many other fields. One such application is the understanding of patterns in financial time-series data. Based upon the past closing values for a stock, we would like to predict the value of the stock in the future with some degree of certainty so that traders may be informed about the state of their investment and respond with the proper course of action. Current methods rely on traditional financial indicators such as ARIMA, linear regression, moving averages, and other methods. For the purposes of risk assessment, anomaly detection, and optimization of asset portfolios, the implementation of machine learning proposes solutions to aide in this complex quantitative decision-making process. In this research project, we propose two different predictive models based upon previous research and developed with machine-learning tools; a support vector regression implementation, and a long short-term memory model. Both models performed with a RMSE of <0.05, and provide a baseline prediction of the next day value of an asset, indicating that further research may prove beneficial for solving the problem at hand.

CCS CONCEPTS

• Computing Methodologies • Applied Computing

KEYWORDS

Machine Learning, Artificial Intelligence, Finance, Computers in other domains

1 Introduction

Predicting the future would always be of value to researchers in every field. For economic analysis, even just a few percent improvement in prediction quality would allow for huge performance gains over time. Stocks are highly volatile, and patterns within difficult to recognize. Recently the field of machine learning has proven its contribution in understanding this particular problem. At the moment, applications of this technology can provide insight into the near future value of an asset, and help traders make informed decisions.

Our approach includes the deployment of two machine learning solutions that predict the next day value based upon a sliding time window. We propose an SVG model, and a LSTM model implemented in TensorFlow with Python that performs better than traditional models alone. The rest of this paper is divided into discussing the problem formation, algorithm design and implementation, experimental evaluation, and related research topics.

2 Problem Formation

We would like to take a particular stock history and predict the next day value of that asset. Then, we produce performance metrics which asses how close this value was to the actual value, and update the model by moving the time window. This network can be retrained for any asset, providing the data wrangling has been handled appropriately.

3 System/Algorithm Design

The input into our model is a sliding time window represented by an pandas 1D X array of length 36, representing the previous 36 days adjusted closing price. The adjusted price is normalized and the network trained against a y value representing the next day’s adjusted closing price. This output from our trained network is a floating-point Y value, which is inverse scaled to produce the current price in US dollars. We would like to optimize our functions to minimize the standard error as they train, producing a quality predictive indication tool.

3.1 System Architecture

We adjusted the history of values to be arrays representing segments of time. This allows us to deploy our predictions on a sliding time scale, and update the neural network weights in the ways specific to each algorithm. Both SVR and LSTM are sequential models, yet SVR is implemented with scikit-learn, and LSTM with TensorFlow. Both models are well formulated to handle regression problems, yet only one of the models is capable of producing predictions further into the future than a single day. Let’s explore each algorithm in detail.

3.2 Support Vector Regression



Figure 1: SVR Network Architecture

*3.2.1 Support Vector Regression*

Support vector regression is a way to normalize data and make predictions based upon a hyperplane that exists in parallel to an imaginary band that projects from the regression line with a width of +/- 1 standard deviation.[1]



Figure 2: SVR Graphical Representation



Figure 1: SVR Kernel Functions

This algorithm is great in theory, as it attempts to minimize error within some preset tolerance, yet when the future happening is taken into account, the hyperplane shifts, and the current projection is updated. We can’t know how future updates will be applied until the present becomes the past, so in truth, this algorithm will only be useful for near future projections such as a 1 day or 7 day lookahead. The farther out in time, the more the uncertainty compounds and the RMSE explodes.

3.3 Long Short-Term Memory



Figure 3: LSTM Top Level Architecture

*3.3.1 Long Short-Term Memory*

LSTM’s are artificial neural networks and more specifically a type of deep learning model. Because it is composed of layers with bias functions with forward and back-propagation, it is useful for time-series predictions in which long-term dependencies and short-term conditions are present in the data.[3] This means it can be potentially used to predict further into the future than the SVR model. As in figure 4, the LSTM architecture is composed of 4 main components, an input gate (bias) controlling how much a network weight updates, a “forget gate” controlling how much a neuron will forget, the current input into the function, and the output.

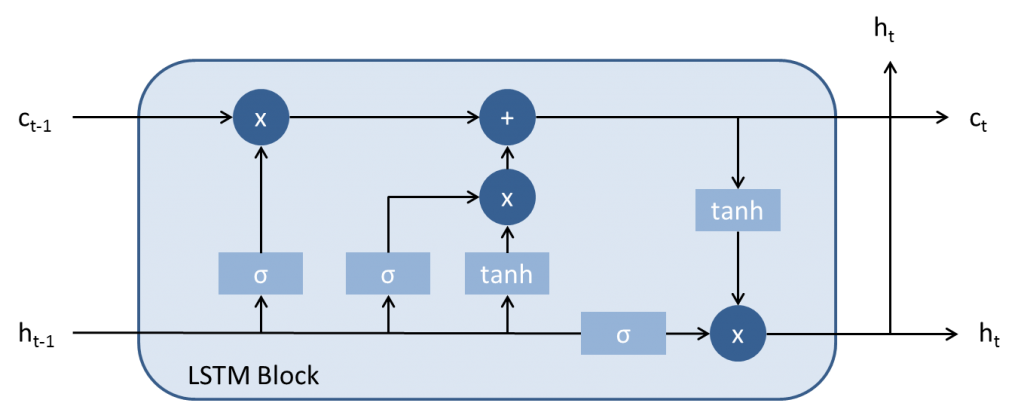


Figure 4: LSTM Layer

The first LSTM block takes the initial state and first time-sequence data, and outputs the first price value. The next block takes the previous network plus with output and calculates its predicted output value, and updates the biases affecting its forward and backward neighbors.

4 Experimental Evaluation

We deployed our architectures to predict the next-day price of the S&P 500 index. We will import the data, manipulate it into the proper normalized form for our network. We will minimize our error in training and compare the RMSE of each model afterward to determine if such analysis would prove beneficial to traders. We would like to at least improve upon a simple moving average as an estimator (see figure 5).

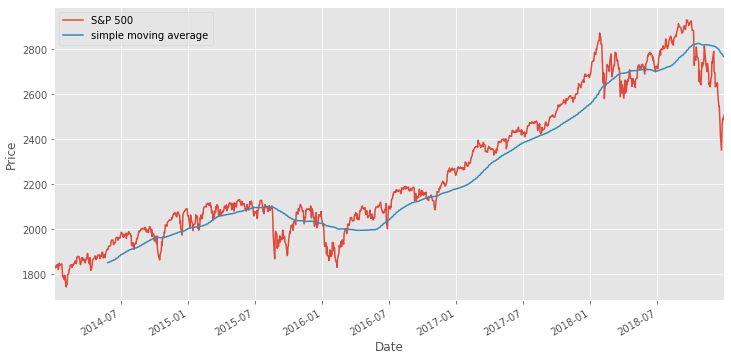


Figure 5: Simple Moving Average

4.1 Methodology

We imported our S&P data from the Yahoo Finance API into Pandas and wrangled the data selecting for adjusted closing price, and formatting the data for a time window of 36 days. This can be interpreted as the time-steps for our algorithm (see figure 6).

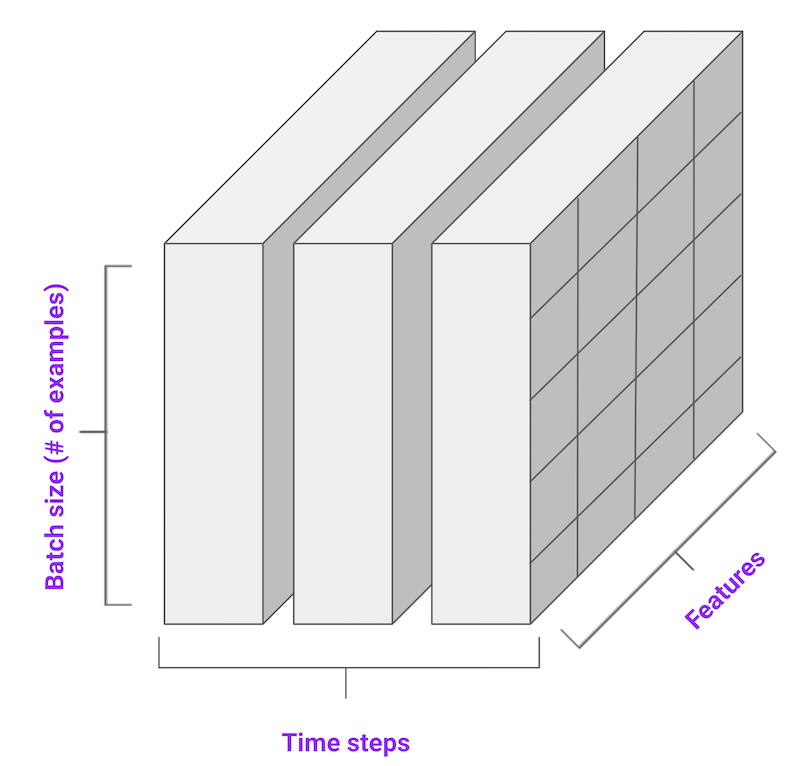
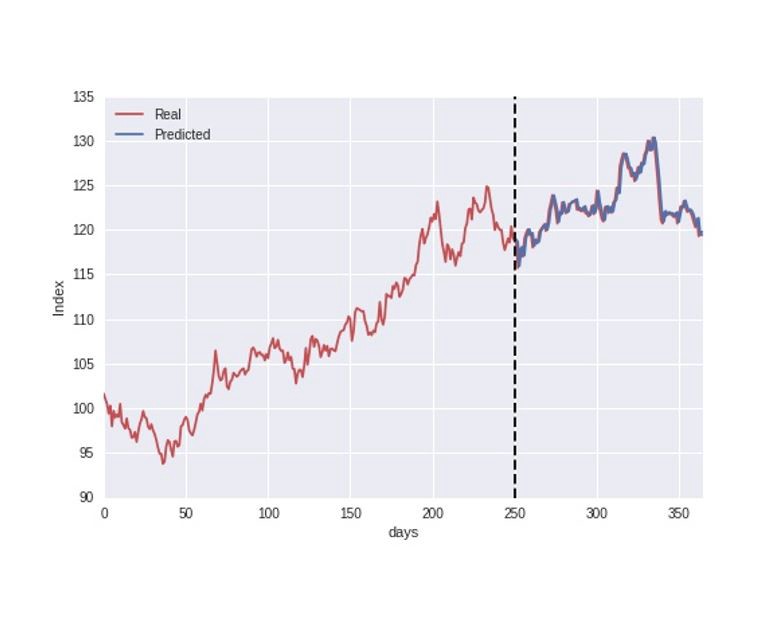
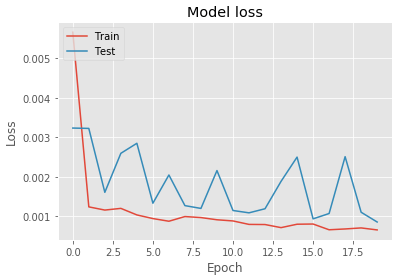


Figure 6: Encoding of Time-Series Data

We collected the history of adjusted price for the S&P 500 from the beginning of 2014 to the end of 2018. This period will serve as our training set. It contains 1222 adj. closing prices serving as our input feature. We split this training data into 70% train, 30% validation. Therefore, we trained on 855 samples of 36 time steps, and validated on 367 samples. We then curled the Yahoo Finance API once again, selecting the adj. closing values for 2019. This serves as our test data. This can be understood visually by referring to figure 7.

 Figure 7: Train/Test Split

We trained both the SVR implementation and LSTM models on this data, and optimized using Adam, with dropout layers in between each LSTM layer set to a coefficient of 0.2.



**Figure 8: Training Loss Function**

We trained the models for 20 epochs, and chose the best one of each. Our results significantly outperform the simple moving average model, with an RMSE <0.05 on the prediction of the next day value.

4.2 Results

Our results prove promising. We were able to minimize error to below 0.05% on both models. This means that if we had a $100 investment, we would at most lose $5 on any given day. This might not make you rich overnight, but it could work well in tandem with other algorithms and traditional financial indicators.



Below is a comparison of the two models’ predictions of the next day price versus the actual price.

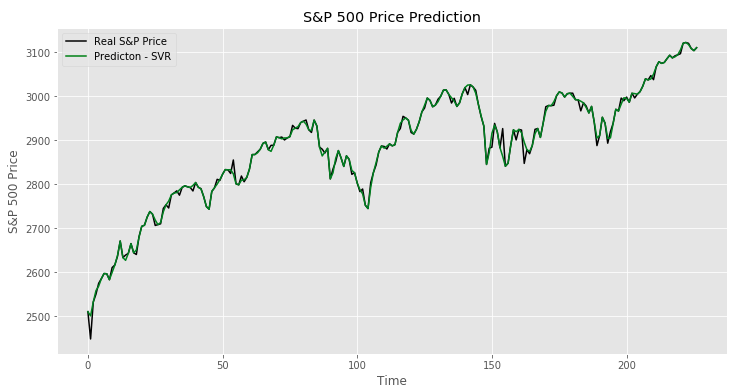


Figure 9: SVR (Prediction vs Actual)

It appears that the SVR is tightly coupled, yet it is actually updating over time. The issue is once the black curve in figure 9 (representing the actual price) disappears, there is no basis to formulate a prediction on. In other situations, the SVR model broke down and the RMSE exploded, so this will only work for a next-day prediction situation or when the model is less noisy. However, it is a quality next-day predictor, at 0.007 RMSE.

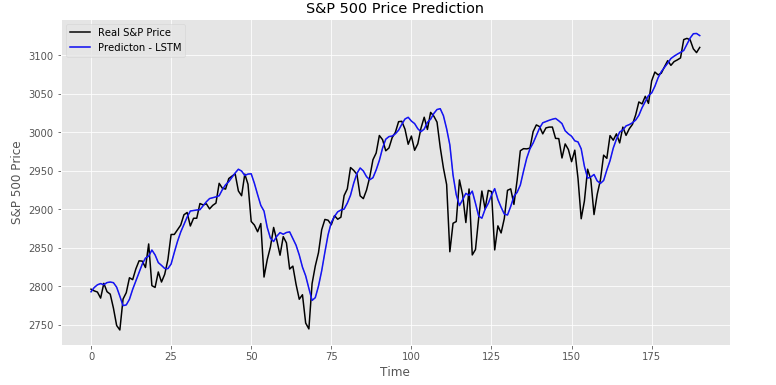


Figure 10: LSTM (Prediction vs Actual)

The LSTM model also performed quite well. The one main issue with LSTM is that it takes time for the network to update and adapt to new data. It is quite good for longer-term predictions, yet there is some RMS type lag with the prediction. It had the lower bounded performance, at 0.02 RMSE, however, in exploration, we have found that the LSTM performs better than the SVR model once the time-scale of prediction increases.

5 Related Work

Other work found applies mostly to time-series predictions. Some previous work [1] showed how using an SVR with a kernel approach could be applied to time series data, yet this application was too generic for our task at hand. The information was only theoretical, but proved a good basis for research. Other research involved implementing LSTM’s with time series data [2] and [3], but not specifically in the field of financial quantitative analysis.

6 Conclusion

You can try with the might of man and machine, but you cannot predict the future. Determining the next day price is an easy task, since the conditions of that day depend most heavily on the day before. However, it is apparent that these advanced solutions to problems of regression can be of beneficial aid to traders, as well as anyone making qualitative decisions involving risk.

7 Work Division

We split the tasks equally and coded the project together. Much of the implementation details that took considerable time were understanding how to implement a time-series structure with an LSTM and SVR. The other chunk of work was the document and presentations. These elements took the most time.

8 Learning Experience

Time Series information is difficult to work with. The stock market, as well as the future, become more uncertain the farther ahead you look. Machine learning may help provide insight, yet the real trading should be handled by a human monitoring sets of algorithms. Perhaps it is a good thing we cannot predict the future.

ACKNOWLEDGMENTS

Special thanks to Dr. Chen. Without your guidance through the machine learning class, this endeavor would remain elusively out of reach to undergraduate students.

REFERENCES

[1] Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. Neural computation, 12(10), 2385-2404.

[2] Gers, Felix A.; Schraudolph, Nicol N.; Schmidhuber, Jürgen (Aug 2002). "Learning precise timing with LSTM recurrent networks" (PDF). Journal of Machine Learning Research. 3: 115–143.

[3] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276.

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00