Predicting Stock Returns Using Deep Learning Models

Amartya Prasad
Department of Mathematics, Courant Institute of Mathematical Sciences
New York University
New York, NY 10012
ap5891@nyu.edu

Steffen Roehershieim
Leonard N. Stern School of Business
New York University
New York, NY 10012
sr4376@nyu.edu

Tianmu Zhao
School of Professional Studies
New York University
New York, NY 10003
tz1264@nyu.edu

Abstract

In this paper, we build a number of Deep Learning Models, such as CNN, LSTM and CNN-LSTM to predict future stock returns using a data set with over 300 features and 3000 stocks for a decade. We learn that modelling stock price movements is rather difficult and more complex models may not necessarily perform better than simpler models like Logistic Regression or Random Forests.

1 Introduction

As shown by several papers, such as [1], machine learning can add value in predicting the cross-section of stock returns. However, it has not clearly been shown that deep learning has significant better performances than classical machine learning models like Random Forest or Gradient Boosting. One of the reasons is that deep learning only really plays out its strength on a large enough dataset with millions of data points, e.g. the US Stock market.
In general, deep learning algorithms have shown outperformance by handling complex correlation structures in computer vision and time series analysis in Natural Language Processing. In Computer Vision, Convolutional Neural Networks (CNN) have shown that they can achieve subhuman performance like YOLO or SSD by analyzing the Complex Correlation Structure of images. In Natural Language Processing, Recurrent Neural Networks like the Long-Short Term Memory (LSTM) Model have led to significant improvement in translation tasks. Both, the CNN and LSTM Model handle very critical parts of financial data extremely well. Financial datasets have the attributes of very complex correlation structures and nonlinear time series dynamics which would make them ideal for a CNN-LSTM Model. The aim of this paper is to combine CNN Structures with a LSTM Model on a large enough financial dataset to reasonably train a such a complex Model and show its potential for outperformance.

2 Problem Definition and Algorithm

2.1 Task

We are using 377 Features to predict the probability that one stock will outperform the Median of all 3000 Stocks. Based on the predicted probabilities we will construct Long-Short Portfolios which will be evaluated on the potential to generate profit.

2.2 Algorithm

We are using the following 6 Algorithms:

- Logistic Regression
- Random Forest
- XGBoost
- Multilayer Perceptron
- Long-Short-Term-Memory (LSTM)
- CNN - LSTM

We are training the sklearn implementation for Logistic Regression and Random Forest. For the Hyperparameter tuning of XGBoost we are using Hyperopt which is a bayesian parameter optimizer. For the LSTM and MLP we do Hyperparameter tuning which is discussed later. The more complex model which is designed by us is the CNN-LSTM (1)

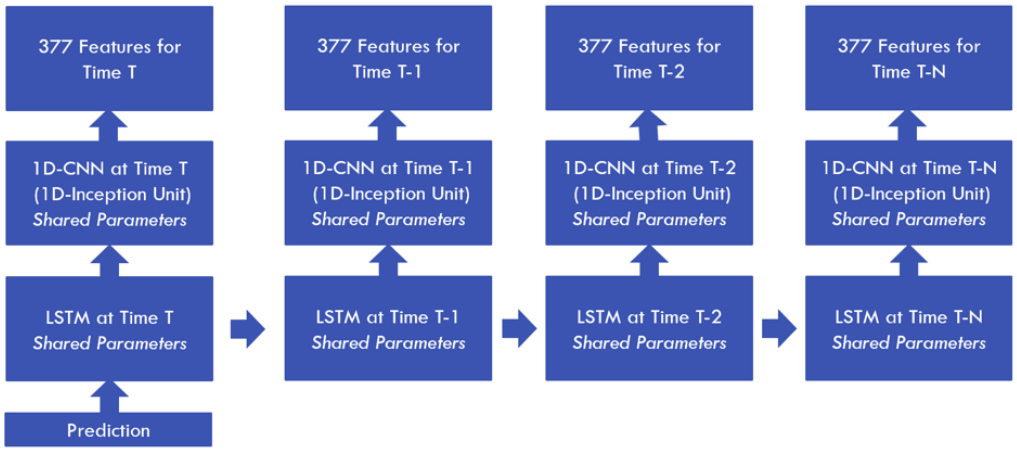


Figure 1: CNN-LSTM

We are using a CNN block which has shared parameters overtime. The outputs of the CNN are forwarded into the LSTM Block which also has shared parameters overtime. We are going back in time for 10 steps.

3 Experimental Evaluation

3.1 Data

The Data for the Stock Selection Model come from CompuStat. The dataset starts in 01-Jan-2004 and is used until 31-Dec-2014. The remainder of the data is for further Out-Of-Sample Analysis. The following filters are applied to the dataset:

- only Common Stock used

- The Stock needs to trade for at least 1 year
- The Company should have no missing or data errors during this period. This leads to an average of around 3000 Stocks in the investment sample. All stocks are adjusted for dividends and corporate events. Below are stocks available over time (2):

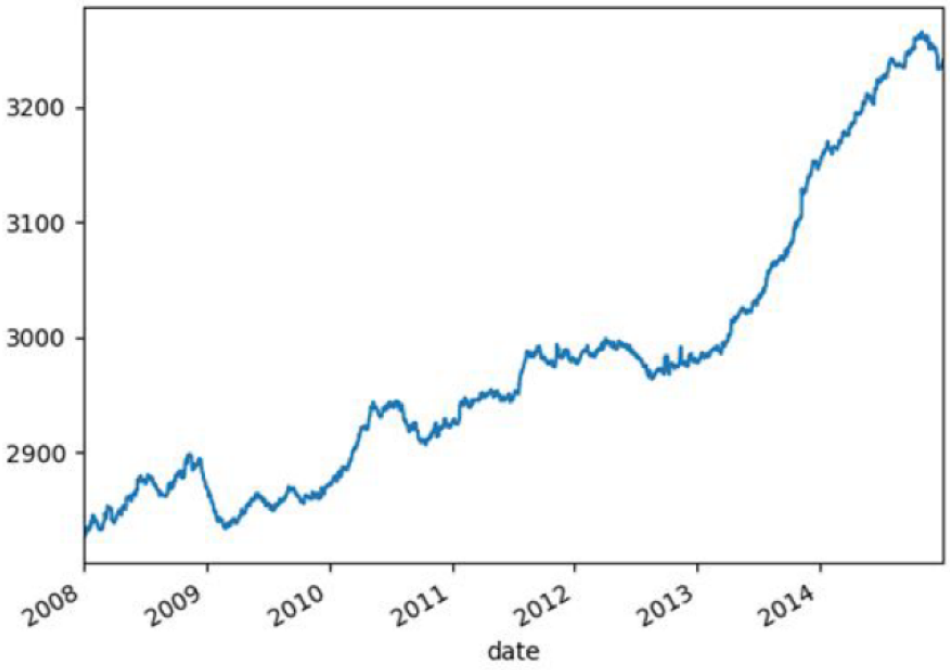


Figure 2: Stocks Available Over Time

Feature Creation The model uses the following Features on a High Level(3.1)

Target The Prediction Problem is a binary Classification. The Classification is based on outperformingor underperforming the weekly median of all stocks returns. Binary Classification from a Machine Learning perspective is a lot simpler than a regression. The Goal is to show that one can add value on the simpler question before moving to a harder regression problem. Roughly 8.4 million target data points are created. Number of Target Data Points = 11 years*252 days*3000 stocks *1 Target = 8.4mil

3.2 Methodology

We design our experiment in the following way. We want to show the step by step improvement for adding more complexity. We startout with a Logistic regression which has only linear Feature Extraction and no knowledge of time series. We move than two different routes, One class of Models which is good at Non-linear Feature Selection like Random Forest, XGBoost, MLP and the second class of model which handle time series extremely well like LSTM models. The Large class of Models is going to be a CNN-LSTM which can theoretically to do both really well. (refer to 3.2)

Training strategy: We trained the model on one year's data and use the trained model to predict the next year, starting from 2005 until 2014, with the first training year being 2005 and the last predicted year 2015

	Nr of Features
Return Feature	31
Price Feature	32
Time std Feature	30
Cross Ranked Return Feature	31
Calendar Feature	31
Cross Std Feature	31
Cross Time Std Ranked Feature	30
Technical Features	53
Technical Features Normalized	51
Technical Features Normalized Ranked	51

Figure 3: Features

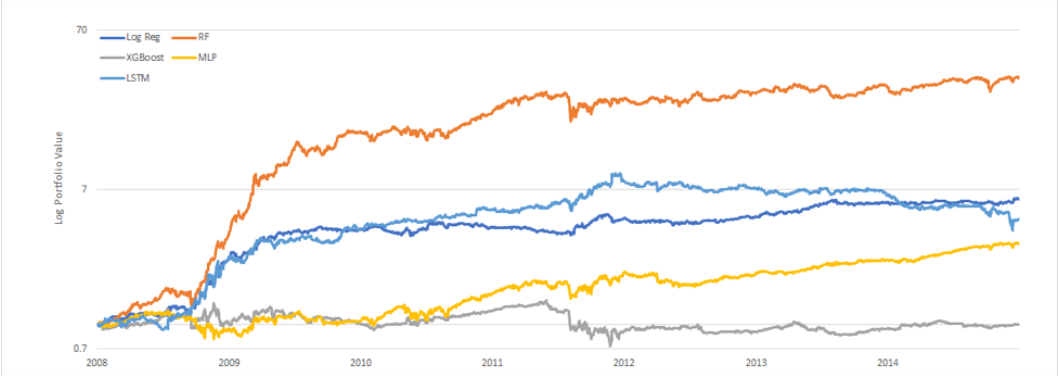
Binary classification: The target labels are either 0 or 1. 1 implies that the predicted 5-day return for that stock is equal to or above the median return.

Multiclass classification: As advised in the presentation, we applied the multiclass classification to our neuron network model in order to better "inform" the model about the problem. The new class label is generated by the following methods: the return of all stocks of each day would be ranked we would label the best 10% to be 2 and worst 10% to be 0, while labeling the medium part to be 1. So we have a 3-class classification problem. The 3-class is applied current only to MLP because of the time limit.

Model selection strategy: we first set up the baseline of all models. We used Logistic regression for binary classification. We then choose 2 groups of algorithms: cross-section algorithms and time series algorithms. The cross-section algorithms includes Random Forest, XGboost and MLP while the time series algorithms we used include LSTM. Then finally, we ensemble these two kinds of algorithms into cross sectional time series algorithm, which in our case, include CNN-LSTM.
Return analyse strategy: We evaluate the performance of the model by simulating the stock investment based on the prediction of models. The decision of stock selection will be made weekly based the prediction of returns of 5 days in the future. We set up 5 portfolios and assign each of them with 100,000 dollars. Each portfolio would make its decision only on specific day of the week. For example, the portfolio 1 will make decision on every Monday, etc. In every decision, we would choose 50 best stocks to long and 50 worst to short. We assume we would equally distribute the money into each of the stocks. And finally we will average those 5 portfolios to acquire return analyse metric of the model.

3.3 Results

Finance Performance :
We actually see that the BaseLine is performing really well from a finance perspective. Almost all



	Base Line	Cross Sectional Algorithm			Time Series Algorithm	Combined Cross Sectional With Time Series Algorithm
Domain Metrics	Logistic Regression	RF	XGBoost	MLP	LSTM	CNN-LSTM
Return	27.62%	55.86%	2.99%	20.02%	25.56%	18.50%
Volatility	18.38%	31.6%	24.1%	25.8%	27.7%	27.8%
Sharpe Ratio	1.50	1.77	0.12	0.77	0.92	0.66
Max DrawDown	-17.26%	-35.1%	-48.0%	-33.3%	-56.2%	-32.3%

Figure 4: Financial Performance Metrics

the more advanced models are underperforming and not working as well. The only Model which is outperforming the Baseline molde is actually Random Forest.

	Base Line	Cross Sectional Algorithm			Time Series Algorithm	Combined Cross Sectional With Time Series Algorithm
Machine Learning Metrics	Logistic Regression	RF	XGBoost	MLP	LSTM	CNN-LSTM
Accuracy	50.91%	51.78%	50.86%	50.42%	50.49%	50.99%
Precision	51.16%	52.03%	51.11%	50.68%	50.65%	51.26%
Recall	48.85%	52.41%	49.07%	46.70%	66.83%	51.75%
F1-Score	49.98%	52.22%	50.07%	48.61%	57.62%	51.50%
ROC	51.01%	52.46%	51.12%	50.67%	50.46%	51.44%

Figure 5: Machine Learning Metrics

Machine Learning Performance The most important Classification Metrics in this example is Accuracy. Logistic Regression has also here a very strong performance compared to the rest. The only Models which are beating the Baseline are Random Forest and CNN-LSTM. XGBoost comes actually very close. However, it is important to point out that this does not translate into actual finance performance.

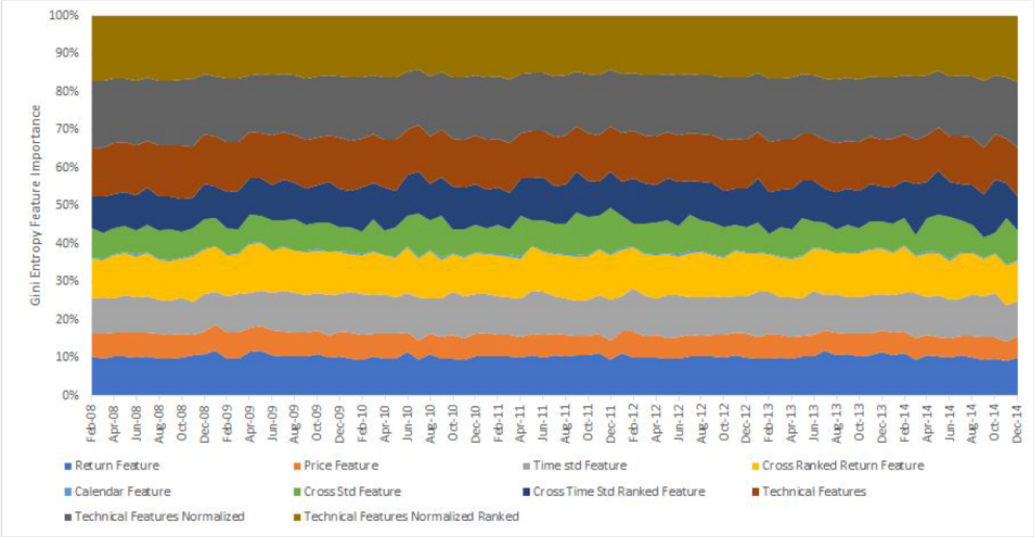


Figure 6: Feature Importance for the Validation Period of the Random Forest

Feature Importance (3.3) shows the Feature Importance for the Validation Period of the Random Forest.Technical Indicators are, as expected, the most important source of Information for the Model. However, The Return Features are in general also important but calendar effects do not matter. Another remarkable point is that the Feature importance is relatively stable over time.

(3.3) shows the Machine Learning Metrics overtime for Random Forest. It is important to point out that the model breaks down and stops work for several periods where the accuracy actually drops below 50%. So it is worse than a random forecast. Potential ways to addressing these problems would be to come up with a Hedging Model or add Bayesian Uncertainty Estimations. The low signal to noise ratio can also be further seen in the ROC curve in (3.3). This shows that one can add value but it is very small and hard.

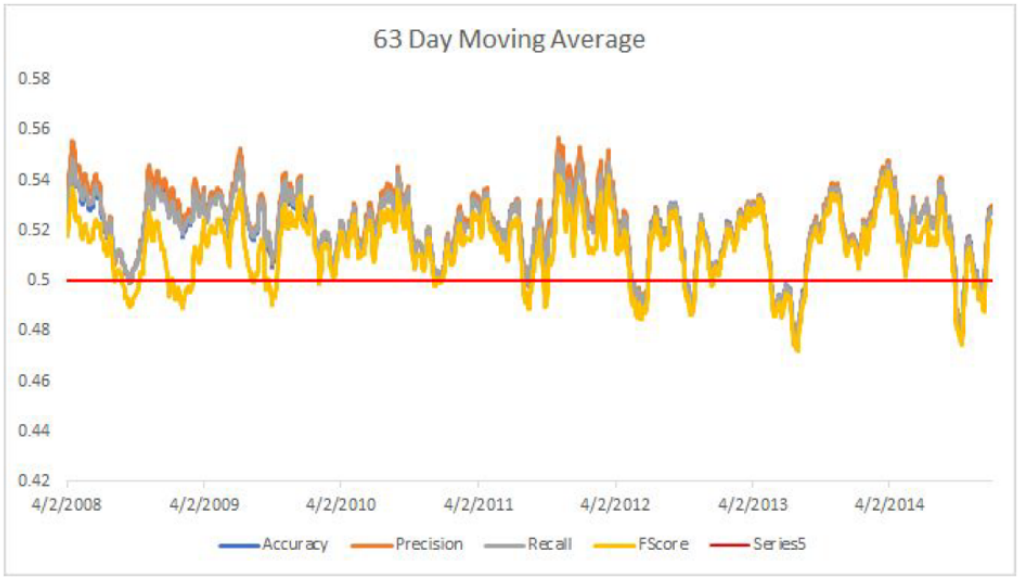
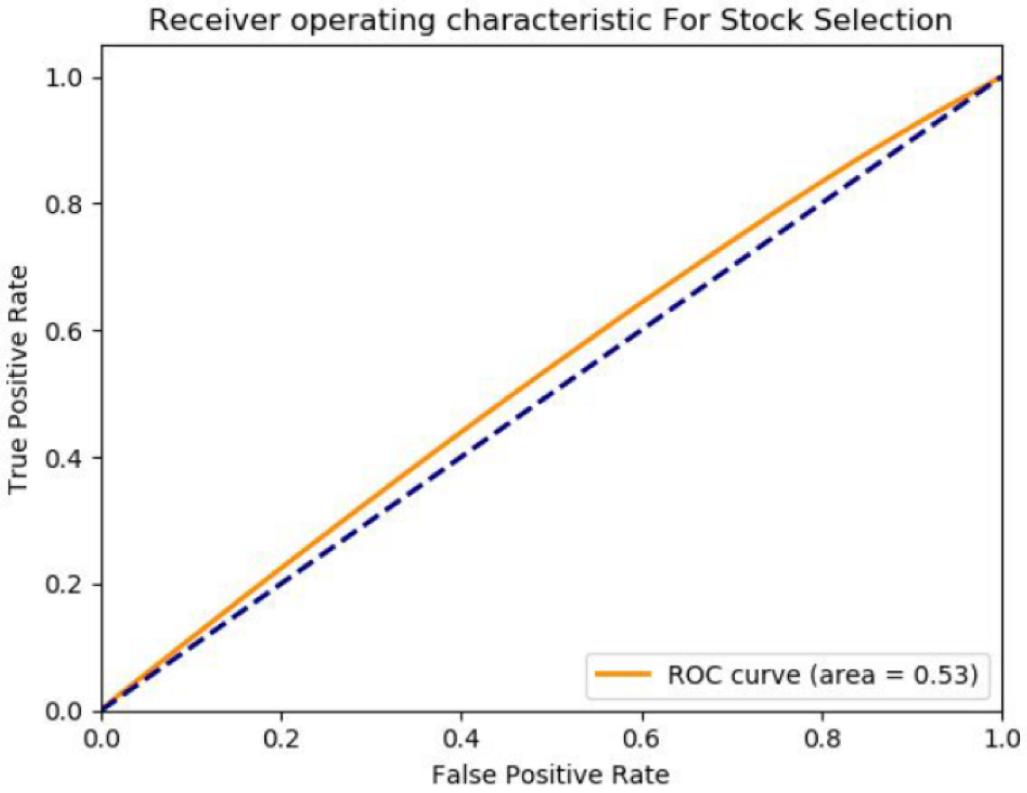


Figure 7: Machine Learning Metrics over time for Random Forest



3.4 Discussion

Our Hypothesis did not hold that a more complex model like CNN-LSTM can outperform a simple model like Logistic Regression on large Financial Dataset. However, we could show that once we are using better CrossSectional Algorithms like Random Forest there is the potential to outperform. The underperforms of the more complex models could be explained by not enough training and tuning. However, another reason could be that those Models need a better defined target than the Median. We actually create a new target which consist out of 3 labels. The fist label is the Top 10% of the stocks over the next 5 Days and second label is the 10% worst performing stocks of the next 5 Days. the last labels are all the remaining stocks. We train this for the Multilayer Perceptron Model and achieve significantly better results.

	Multilayer Perceptron	
Domain Metrics	Binary Classifcation	3 Classification
Return	20.02%	36.22%
Volatility	25.8%	17.6%
Sharpe Ratio	0.77	2.06
Max DrawDown	-33.3%	-19.3%

The Return, Sharpe Ratio and MaxDrawDown are improving significantly (3.4).

4 Conclusions

The Conclusion is that it is very difficult to train and build more complex models to outperform simple base Models like Logistic Regression on that Dataset. Further Work would include better training and Hyperparameter training, create better Targets or labels but also include Bayesian uncertainties around the Predictions to see when the Model maybe starts to break down.

5 Supplementary Information

Our code can be found on `https://github.com/kewiei/PTSA-project`

6 Acknowledgments

We would like to acknowledge Professor Cristina Savin from NYU's Center for Neural Science for discussions.

References

[1] T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018. doi: 10.1016/j.ejor.2017.11.05. URL `https://ideas.repec.org/a/eee/ejores/v270y2018i2p654-669.html`.

7 Student Contributions

All three members were involved with each part of the project.