

1 Overview

The goal is to be able to predict the future direction (and magnitude) of change in the difference between yields of risky and risk-free securities, known as the credit spread. Our focus here will be on predicting the spread of investment-grade US corporate bonds over US treasury bonds, using past information about the credit spread along with assorted exogenous data to construct models for how the credit spread changes over time. We then extrapolate from these models to make claims about the credit spread's future movement. The two main learning frameworks we used were Support Vector Machines and Long Short-Term Memory Neural Networks. We also developed an ARIMA model of the data to use as a benchmark for our learning models.

2 Description of the Data

2.1 Credit Spread Data

All of the predictive models train on the difference between corporate bond and treasury bond yields for various different maturities. The bond yield data was all obtained from the St. Louis Federal Reserve's [FRED Economic Database](#). The corporate bond data was taken from the ICE Bank of America Merrill Lynch Corporate Effective Yield for [1 to 3 year](#), [3 to 5 year](#), [5 to 7 year](#), [7 to 10 year](#), and [10 to 15 year](#) maturities. The treasury bond data was taken from the [1 year](#), [2 year](#), [3 year](#), [5 year](#), [7 year](#), [10 year](#), and [30 year](#) Treasury Constant Maturity Rates, in addition to the [1 month](#), [3 month](#), and [6 month](#) Treasury Constant Maturity Rates.

Our first step towards computing the spread was normalizing the spread data to have mean equal to 0 and standard deviation equal to 1. We then arranged it into four groups based on bond maturity: 1 to 3 years, 3 to 5 years, 5 to 7 years, and 7 to 10 years. Since the corporate and treasury data for the given maturity groups have different granularity, the spread could not be computed as a pointwise difference. To compute the difference in the yields for each maturity group, we used numerical integration to find the area between the two yield curves. The result of these transformations is four sequences of time-series data of length 5509, one for each maturity group. The data begins on the 31st of December, 1996, ends on the 17th of January, 2019, and every time-step in the data is one business day. These time-series data were used to train the models used to predict the spread.

2.2 Exogenous Data

Some of our learning models (namely, the Neural Network) are capable of easily incorporating potentially irrelevant data into the model and autonomously weighing its influence so that the credit spread is optimally predicted. Therefore, we also make use of Dow Jones, S&P 500, NASDAQ, and VIX data from 1997 and inflation rate data from 2000 to the present.

3 Training and Testing the Learning Models

3.1 ARIMA

Our baseline model is the Autoregressive Integrated Moving Average. The ARIMA model was trained exclusively on the credit spread data, incorporating no exogenous information. The model is first fit to the first 4509 training elements of the credit spread time-series. Afterwards, the model

was iterated on the remaining 1000 validation points. The iteration consists of predicting one new value at a time, adding it to the training set, and refitting the model on the newly expanded training set. This is a fairly slow model to train and predict with, but it gives fairly good results as summarized in Table 1. The following figures plot the validation data points against the predicted ARIMA sequence during the iteration.

Table 1: ARIMA Root Mean Squared Error

Series	RMSE
1 to 3 year spread	0.013
3 to 5 year spread	0.014
5 to 7 year spread	0.014
7 to 10 year spread	0.015

Figure 1: ARIMA 1 to 3 year spread

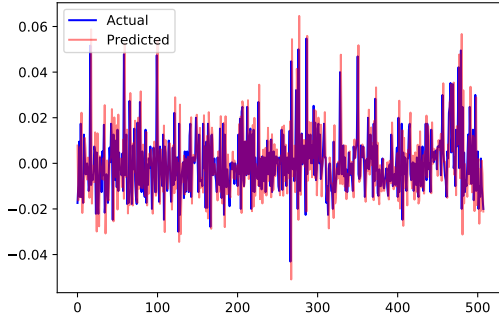


Figure 2: ARIMA 3 to 5 year spread

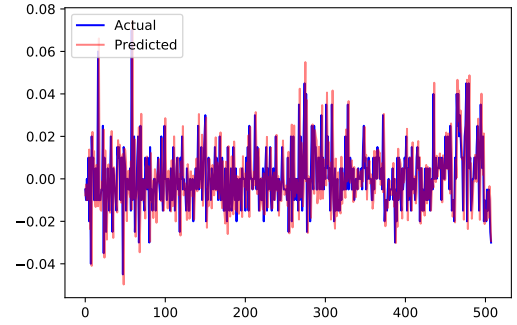


Figure 3: ARIMA 5 to 7 year spread

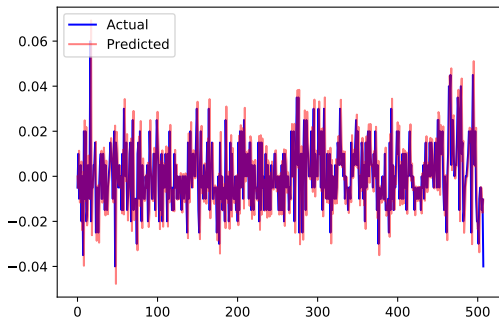
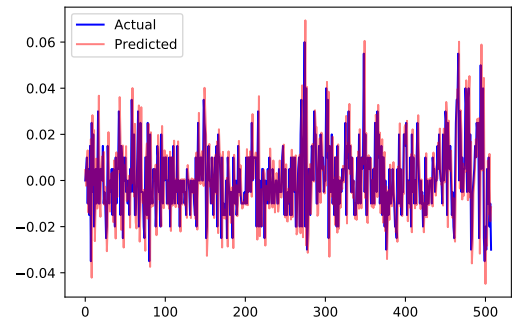


Figure 4: ARIMA 7 to 10 year spread



As we can see, the ARIMA achieves a very low Root Mean Squared Error, but it is very sensitive to high-intensity, low-impact fluctuations in the spread data. This is clear evidence of overfitting, and indicative of ARIMA's lack of suitability for predicting the credit spread.

3.2 Support Vector Machines

We implemented our Support Vector Machines (SVMs) using the SciKitLearn module `sklearn.svm.SVR`, which very conveniently implements a Support Vector Machine for regression instead of classification, which is more useful for our project. We were initially going to use a classification SVM

to classify future dates into one of either the “downward trend” or “upward trend” categories, but a regression captures not only this information but also information about the magnitude of a possible downward or upward trend.

In our SVM, we used the radial basis function (or **rbf**) kernel and trained the model on the first 5009 data points, using the remaining 500 data points for validation.

The data had to be transformed from time-series data to points in \mathbb{R}^n since the SVM is not capable of training directly on time-series data. To get around this, the model trained on the difference from one day to the next in the credit spread, labeled by the actual credit spread value for that day. The model also does not take into account just one day’s spread value to predict the next day’s; we implemented a 3-day look-back, so that the spread data for the previous 3 days was considered for predicting the next day’s spread. This resulted in having a tall/skinny rectangular, circulant matrix for our training data, instead of a long vector.

We also implemented 1-day and 5-day look-backs, but they did not perform as well as the 3-day look-back SVM model. The SVM’s performance on the validation data is summarized by the figures below.

Figure 5: SVM 1 to 3 year spread

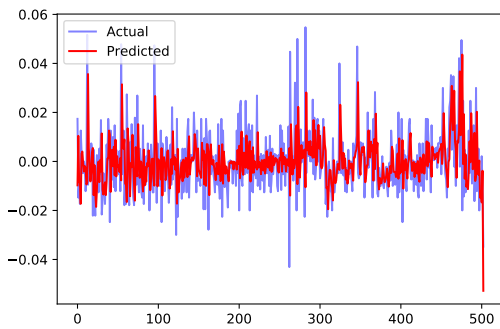


Figure 6: SVM 3 to 5 year spread

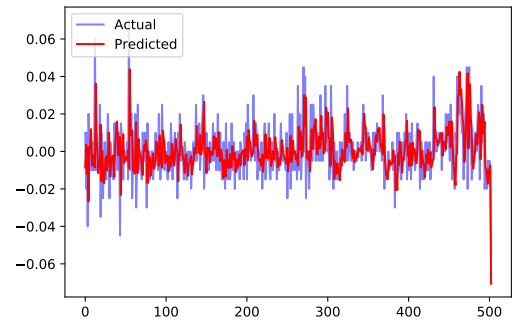


Figure 7: SVM 5 to 7 year spread

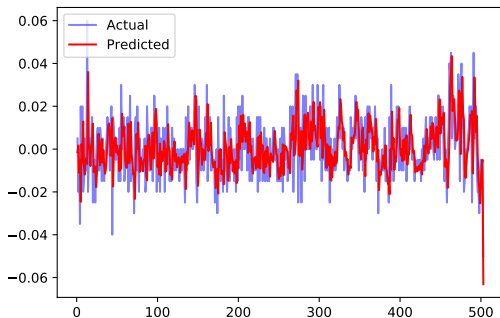
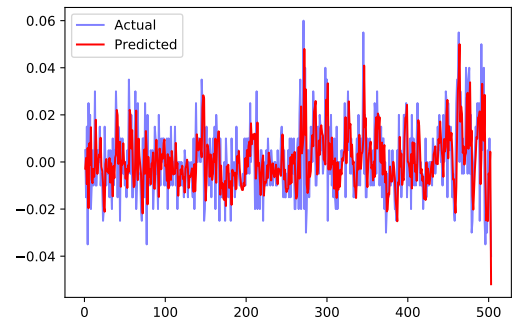


Figure 8: SVM 7 to 10 year spread



As we can see, the SVM model is predicting a high-magnitude downward trend at the end of the validation set for all of the maturity groups, so if we believe the SVM model, we would predict a strong downward shift in the credit spread, which would indicate confidence in the economy, among other things.

However, the caveat with this model is similar to that of the ARIMA: it is too sensitive to very large but fleeting fluctuations in the credit spread data. Therefore, we should not ascribe a high level of confidence to the prediction this model is making, since the very large downturn it predicts

is very likely to be a simple result of overfitting to noise in the data.

3.3 LSTM Neural Networks

Since the SVM seems to perform similarly to the ARIMA model, and both seem to fail to generalize past noise in the data, we attempted to use Neural Networks, which are famous for their ability to eschew irrelevant information and learn their own relevant features for the data. However, a traditional feed-forward neural network would have trouble learning on time-series data, in a similar manner to the SVM, since it is essentially just a function from \mathbb{R}^n to \mathbb{R}^m , and reducing the input dimension to $n = 1$ would be disastrous for the predictive capabilities of the network.

We initially had the idea of doing a look-back-based approach, as in the SVM, to arrange the data into points in e.g. \mathbb{R}^3 to train the network, but then we discovered that Long Short-Term Memory (LSTM) neural networks, a kind of recurrent neural network, are ideally-suited for making predictions on time-series data. As a result, we decided to use the **Regression LSTM Networks** implementation in **MatLab** for our neural network.

We used a variety of architectures for the neural network, as well as training the network on an assortment of data, to compare the relative effectiveness of these choices on the prediction quality. First, we used a simple architecture consisting of two sequential LSTM layers, each with 150 nodes, followed by a fully-connected layer with 150 nodes, which finally produces an output. This architecture was trained the 1 to 3 year maturity credit spread time-series in addition to various combinations of exogenous data from Dow Jones, S&P 500, NASDAQ, VIX, and the inflation rate. For the credit spread time-series, the first 5249 data points were used for training, and the remaining 260 were used for validation. The performance of these networks on the validation sets, along with graphs of the Root Mean Squared Errors, are presented in the figures below, annotated with the exogenous data used to train each model.

Figure 9: DJ, S&P 500, NASDAQ, VIX, inflation

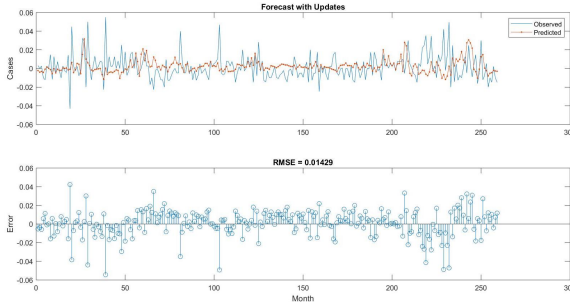


Figure 10: No exogenous data

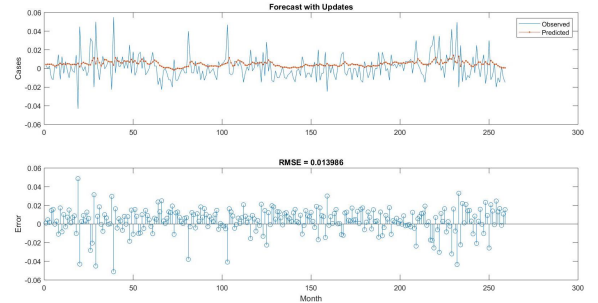


Figure 11: Dow Jones

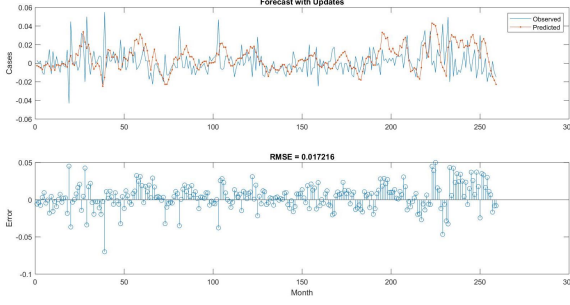


Figure 12: S&P 500

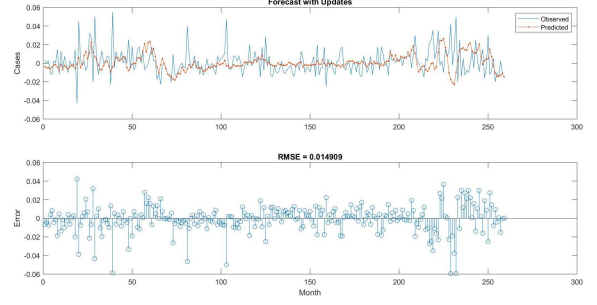


Figure 13: NASDAQ

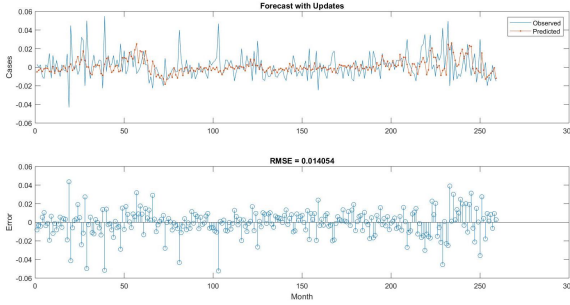
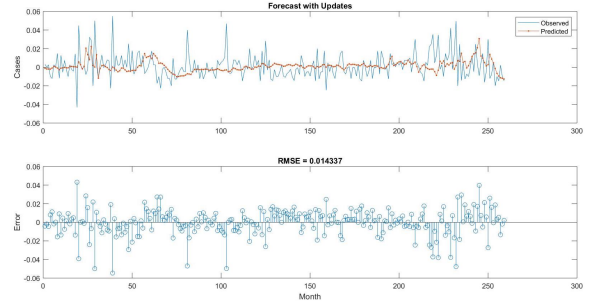


Figure 14: VIX



Next, we tried fixing the data set as the spread data plus S&P 500 and VIX and varying the neural network architecture. We tried one LSTM layer followed by two fully-connected layers with 100 nodes each, two LSTM layers followed by two fully-connected layers with 100 nodes each, two LSTM layers followed by two fully-connected layers with 50 nodes each, and two LSTM layers followed by two fully-connected layers with 200 nodes each. All of these architectures were trained on the same set of data: the credit spread time-series, along with exogenous information from the S&P 500 and VIX. The first 5259 points in the credit spread were used for training, while the remaining 250 were used for validation. The performance of the neural networks on the validation sets is presented in the figures below.

Figure 15: One LSTM layer, two fully-connected layers, 100 nodes each

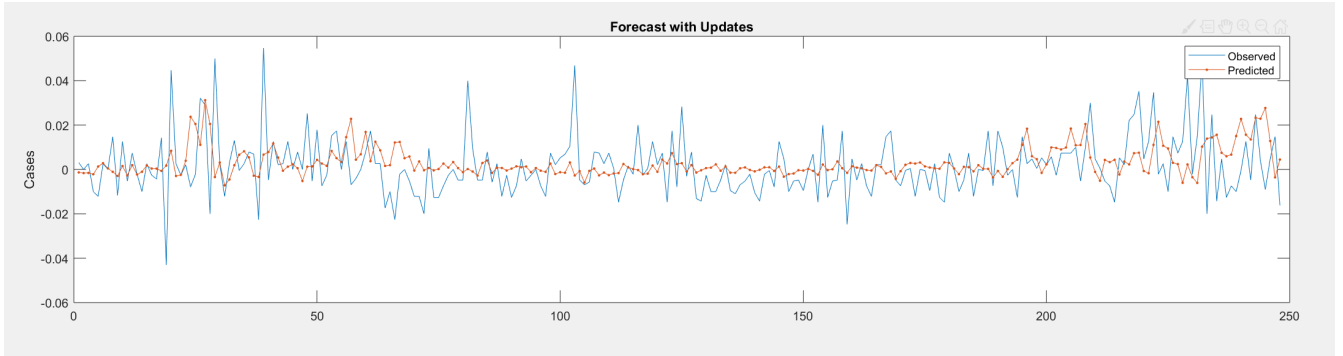


Figure 16: Two LSTM layers, two fully-connected layers, 100 nodes each

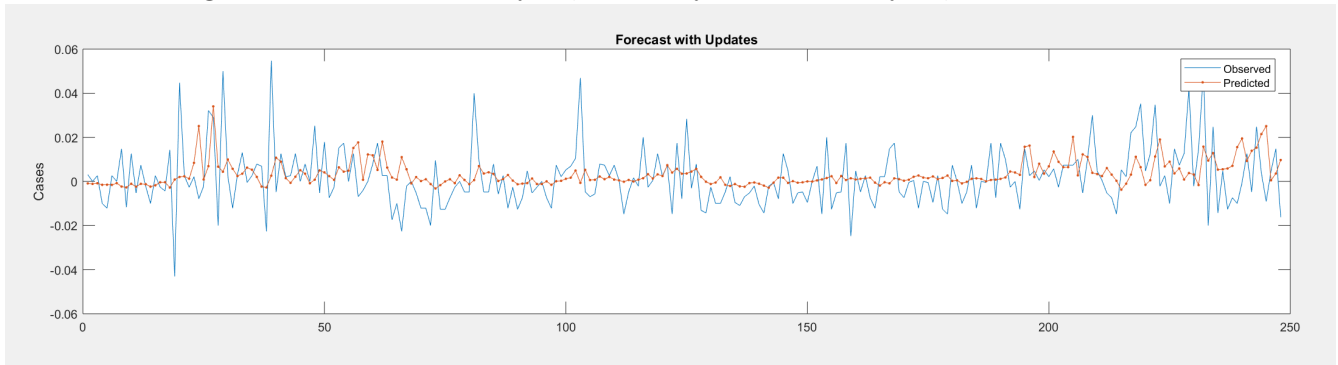


Figure 17: Two LSTM layers, two fully-connected layers, 50 nodes each

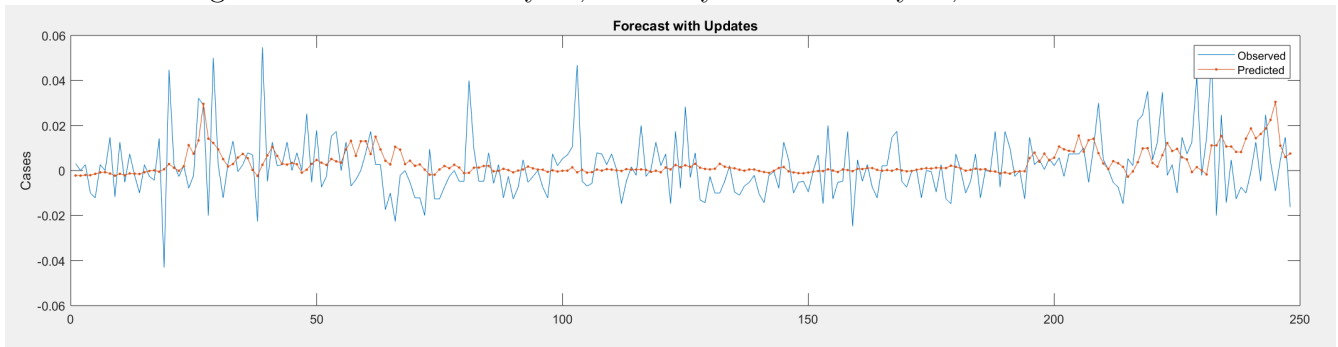
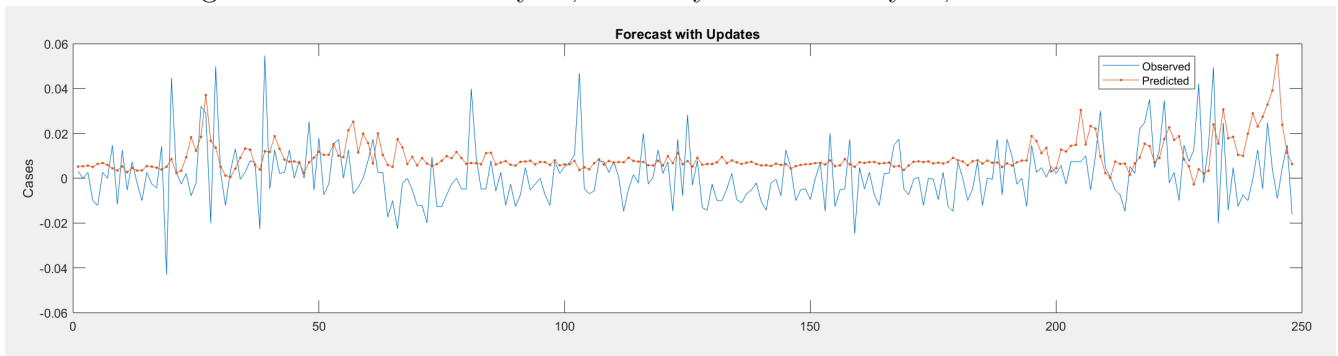


Figure 18: Two LSTM layers, two fully-connected layers, 200 nodes each



4 Summary of Results

5 Possible Future Directions for Research