Forecasting Credit Spread Movements Using Machine Learning

Abstract: In this paper we are forecasting the magnitude and the direction of the Credit Spread, which is the difference of AAA Corporate Bond Yield and 10 – Year Treasury Constant Maturity Yield. We predicted the magnitude and direction of the credit spread using simple models such as Simple Moving Average (SMA) and a variant, the Exponential Moving Average (EMA). The SMA and EMA models hold a strong assumption that the prediction is the average of the past. This property makes them ineffective in forecasting an arbitrary number of days into the future. Next, we used the ARIMA model, but it requires stationary data and it does not capture non-linear relations between features. We used a Hidden Markov Model to find different regime shifts and produce stationary data. However, we were still not capturing non-linear relations between features. Finally, we implemented different Long Short-Term Memory (LSTM) models which resolved all the drawbacks of the previous models and performed the best at forecasting the credit spread.

Keywords: Credit spread, Machine Learning, Neural Networks, Recurrent Neural Networks, Long Short-Term Memory (LSTM), Forecasting, Vector Autoregressive Integrated Moving Average (ARIMA), Hidden Markov Model (HMM), Simple Moving Average (SMA), Exponential Moving Average (EMA), Technical Indicators.

CONTENTS

1. INTRODUCTION

Credit spread or Yield spread represents the difference in the yield of a risky security versus that of a risk-free security with same or different maturity. In layman terms, credit spread represents the confidence in the economy: a higher credit spread means uncertainty in the economy whereas a narrower one represents confidence in the economy. In this paper we are forecasting the direction and magnitude of a credit spread; Moody's Seasoned AAA Corporate Bonds' (maturities 20 years and above.) yield relative to yield of 10-Year Treasury Constant Maturity (Figure 1).

There are a variety of ways to forecast the credit spread: a rule-based method (SMA and EMA model), Statistical Learning (Linear Regression and Vector Autoregressive Integrated Moving

Average), and Machine Learning (a special kind of Recurrent Neural Network - LSTM model and its invariants).

We start with the simpler models (easy to understand) and discuss the problems associated with these models. We then move towards a machine learning based approach and show why it is better for forecasting time series data.

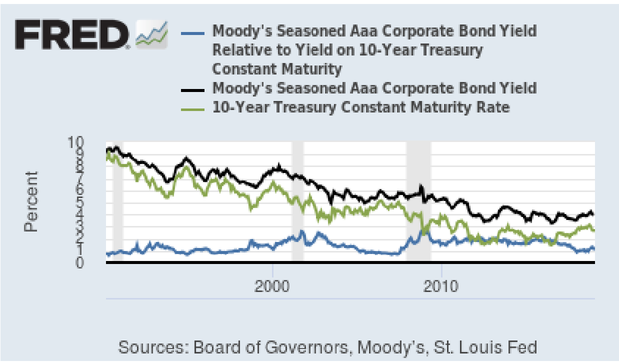

Figure 1: Credit Spread, 10-year Constant Maturity Rate, Moody's Seasoned AAA Corporate Bond Yield [1]

2. MODEL FRAMEWORK

Define Variables

Leg 1 = Moody's AAA Corp Bond Yield

Leg 2 = 10-year constant Maturity Rate

Credit Spread = Leg 1 – Leg 2

To define a model, we first need to understand factors which can affect our credit spread time series.

a.Factors Affecting Credit Spread Time Series

1) Fundamentals

Credit spread can be affected by the dynamics of supply and demand.

Demand	**Supply**	**Bond Price**
Increase	Decrease	Increase
Decrease	Increase	Decrease
Increase	Increase	Unpredictable
Decrease	Decrease	Unpredictable

Table 1: Supply and Demand of leg 1 and leg 2

We require an indicator which can explain the effects of this fundamental factor of supply and demand in our data. We hold the assumption that the Effective Federal Fund Rate is this indicator. The Federal Open Market Committee (FOMC) meets eight times throughout the year to determine the federal funds target rate. The FOMC considers a variety of economic factors, such as: employment, consumer spending and income, business investments, etc. The movement of the effective federal fund rate has a strong correlation with the supply and demand of leg 2 in our model framework. This, then, indirectly affects leg 1.

There are other fundamental factors which can affect the credit spread. However, we assume that specific risk can be eliminated by using portfolio diversification. We are only interested in forecasting systematic risk present in our credit spread series.

2) Sentiments

We define sentiment as the confidence in the economy. In order to quantify sentiment, we are using the ted spread:

Ted Spread = 3 Month LIBOR based on US dollars – 3 Month Treasury Bill

LIBOR - rate at which one bank charges to lend to other banks

We consider the 3 Month Treasury Bill to be the safest asset in the market and ted spread to be the risk premium. Therefore, if ted spread increases, there is a higher risk premium. A higher risk premium implies less confidence in the economy.

3) Exogeneous Variables

The credit spread time series also depends upon other asset class returns and volatility. Figure 2 shows that it is difficult to determine whether the credit spread is directly proportional or indirectly proportional with respect to these variables. For example, in stocks and bonds sometimes we see positive linear correlation and other times negative linear correlation.

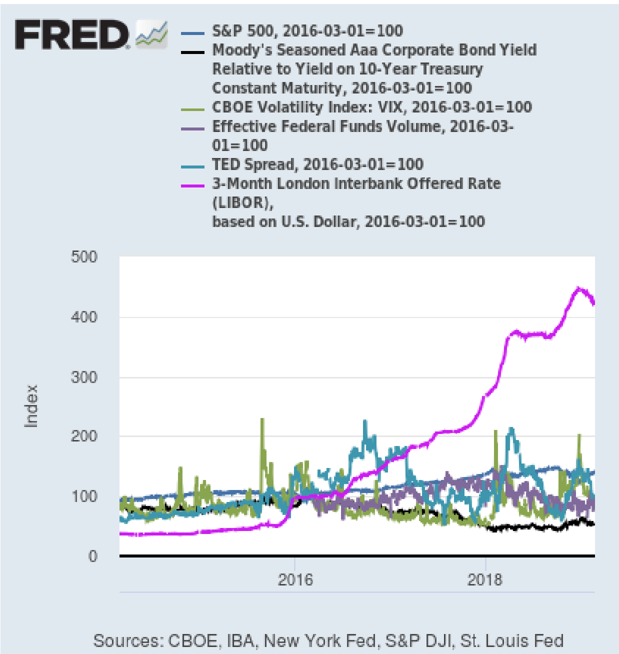

Figure 2: Exogeneous Variables [2]

4) Software technical errors

There may be bugs in trading software which may generate and execute a lot of buy and sell orders which can affect the credit spread series. Humans are not perfect, and this is unpredictable.

b. Assumptions for Defining Model

1) The past can give us some clues about direction and magnitude of our credit spread series. We develop models to learn those patterns and apply it to the present to predict the future. However, this is not going to be perfect because our models cannot and do not include future policies and events which is crucial in perfect forecasting.

2) Our data consists of the daily closing price of the credit spread. Therefore, if some algorithms observed anomalies during closing time and traded a lot of quantities of the legs of our credit spread, this pattern may arise in our models. If this kind of event happened, then our assumption is that the market will revert back to its state just before this event. Then, due to technical errors in the software, if price increases during closing time then we are not able to predict this phenomenon.

3) We are assuming that investors and traders are rational. This means that if they perceive that credit risk is present in the economy, then their trading activity will widen the credit spread and not narrow the spread down.

c. Data Engineering

We are using the daily data at the end of the trading day. The logic that we have used for missing data entries is to take the average of the 3 rolling past and future data points. If one or more data point(s) is missing, then we consider the average of the remaining data points. Data was extracted from St. Louis Fed data repository (https://fred.stlouisfed.org). We used R and Python programming language for implementing all our models. The libraries used for each model have been mentioned in the respective model discussion.

We calculated MSE, SMAPE, MAE and MDA for each of our models, and the same was used for evaluating our models.

MSE - Mean Squared Error measures the average of the square of the errors[10][14].

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

SMAPE - Symmetric Mean Absolute Percentage Error[11] is an accuracy measure based on percentage of errors.

$$\mathrm{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

At is Actual Value and Ft is Predicted Value

MAE - Mean Absolute Error[12] is a measure of difference between two continuous variables.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}.$$

MDA - Mean Directional Accuracy[13] is a measure of prediction accuracy of a forecasting. It compares the forecast direction (downward or upward) to the actual realized direction.

$$\frac{1}{N} \sum_t \mathbf{1}_{sign(A_t - A_{t-1}) == sign(F_t - A_{t-1})}$$

3. Naïve Models - SME and EMA

a. Logic

The simple moving average predicts the next day to be the average of n past days, where n is the window size. The window then moves to make a prediction for the next day. This is known as the rolling mean. Exponential moving average is a slight variant of SMA which preserves values that occur earlier in the window.

b. Method

The SMA and EMA models assume that the prediction for the credit spread of tomorrow is a simple function of the previous days.

To implement the SMA model, we looped through our training data and calculated the prediction for the next day to be the average of the previous n days. This resulted in a list of predictions which we compared with the true values to calculate MSE, MAE, and MAPE.

To implement the EMA model, we used the same approach as implementing the SMA model, however, there is a decay factor which weighted earlier values greater. This model only takes a small fraction of the most recent values enabling the preservation of older values. There is no window for the EMA model as the running mean is calculated as a function of the previous running mean and the decay factor.

c. Results

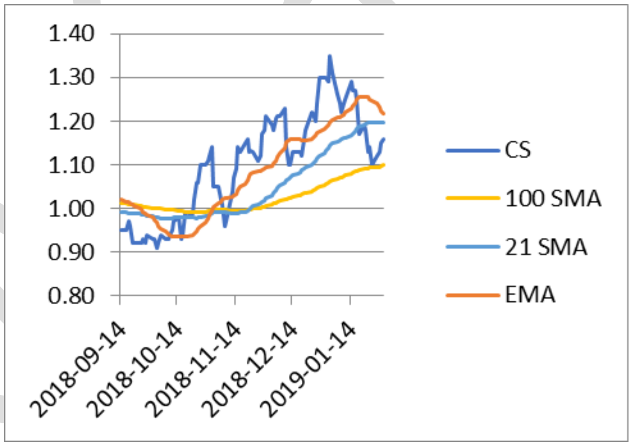

Figure 3: SMA & EMA models

	MSE	MAE	MAPE	MDA
EMA	0.00004	0.00635	2.10513	15.24
SMA (window=21)	0.00090	0.0271	9.08018	24.51
SMA (window=100)	0.00585	0.07855	27.44802	38.02

Table 2: Predicting one day ahead

There is a trade-off between magnitude and direction accuracy. The larger the window size, the lesser the accuracy of magnitude prediction but the better the direction prediction and vice versa.

d. Pros and Cons

These simple models are very accurate at predicting one day ahead but cannot predict further. The simple formula used by these models do not take any economic factors and shocks into account.

e. Technical Indicators

To improve the predictive power of our models, we used technical indicators like Bollinger bands, Moving Average Convergence and Divergence (MACD) and Relative Strength Index (RSI) but they have more or less the same problems as SMA and EMA models.

Graphs for RSI, Bollinger Bands, MACD:

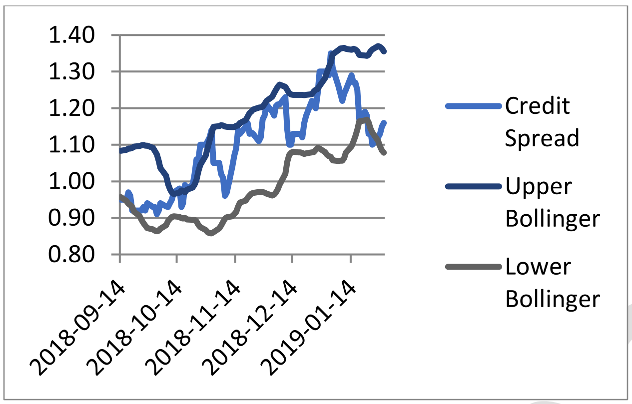

Figure 4: Bollinger bands

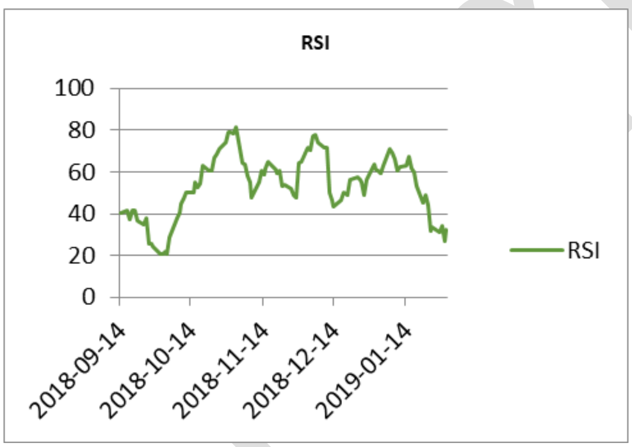

Figure 5: RSI

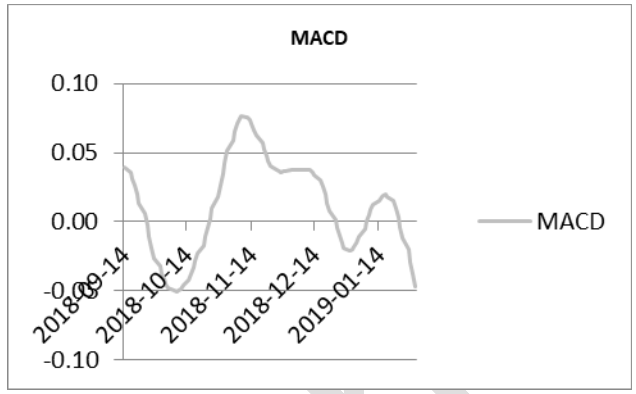

Figure 6: MACD

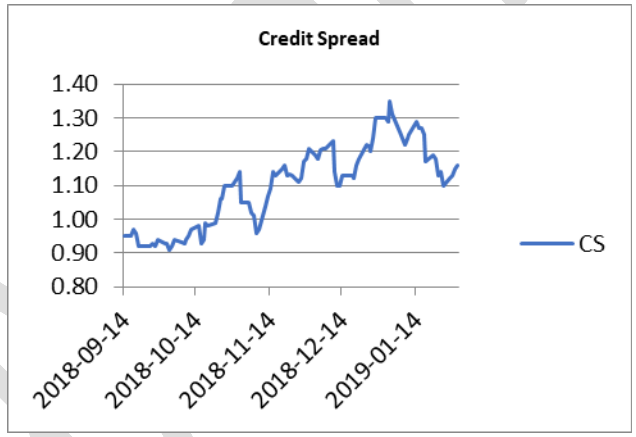

Figure 7: Credit Spread

4. ARIMA Model

a. Logic

An Autoregressive Integrated Moving Average (ARIMA) Model is very popular in time series analysis [4]. Box and Jenkins (1970) popularized this model.

To capture momentum and mean-reversion, the model considers its own past behaviour as inputs. To capture "shock" information in a time series, a moving average is used. Historically, the ARIMA[5] model performs best under the following assumptions:

b. Assumptions

1) The error process is homoscedastic (constant) over time

2) Time Series is weak stationary
3) Time Series have short memory

c. Results

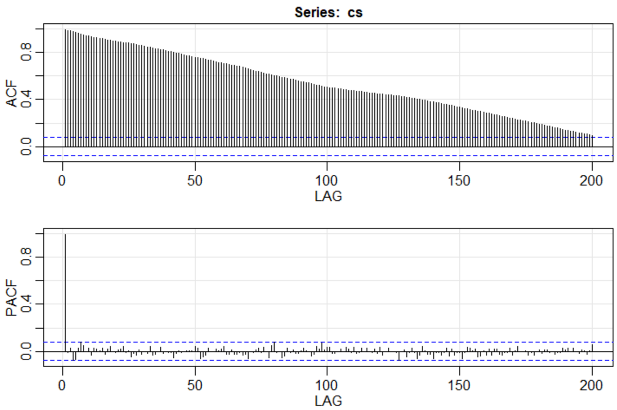

Figure 8: ACF (Auto Correlation Function) and PACF (Partial Auto Correlation Function) of Credit Series

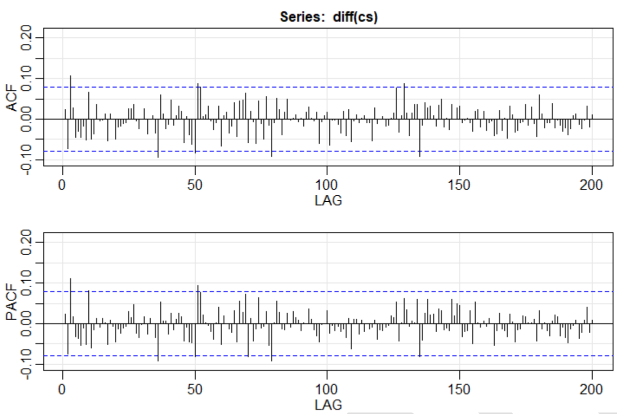

Figure 9: ACF and PACF of the interday difference of the series

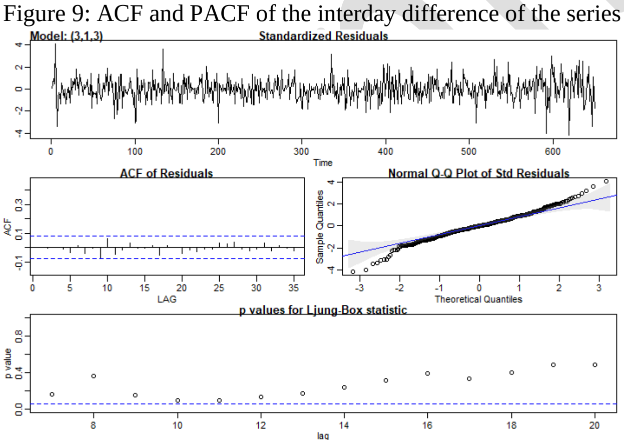

Figure 10: Residual Diagnostic plots for Model (3,1,3)

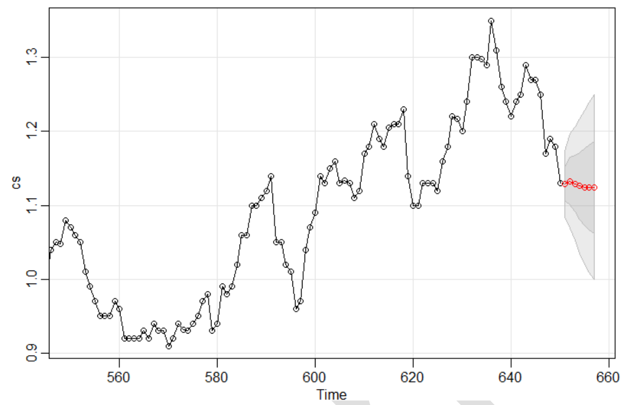

Figure 11: ARIMA Model Forecast

Figure 8 shows that the original credit spread series maintains long memory as ACF decays slowly. This means correlation is still present even at a large lag. Our series is not stationary. To make our series stationary, we have taken the interday difference. When you perform this operation you lose information, but Figure 9 shows our resulting data is stationary. Now, we are able to fit the model with an autoregression of p = 3, an interday differencing of 1, and a moving average of q = 3. Our residual diagnostic plots in Figure 10 show that the resulting model fits the assumptions mentioned above. Finally, we use this ARIMA model to forecast the credit spread shown in Figure 11. The model is good for predicting short term but not great for long range forecasting. In the long term, the model just reverts to the mean; in other words, just predicts a straight line.

d. Shortcomings

1) The ARIMA Model captures the linear relationship between features and does not predict well when relationship is non-linear.
2) Strong assumptions mentioned above
3) Not good for long range forecasting

To solve these problems we can first, decompose credit spread into linear and non-linear components. Then we can apply ARIMA model for linear component and then use neural network for capturing non-linear component. However, it is not good to divide the time series into two components because we cannot ensure its accuracy. To solve this kind of problem we can use Long Short-Term Memory Model.

5. Hidden Markov Model for Regime Identification

We used a Hidden Markov Model (HMM) to find different regime shifts in our data. Therefore, the resulting data is stationary in the different regimes.

HMM parameters are trained by using expectation–maximization (EM) algorithm which is a gradient-based optimization method. The algorithm will generally get stuck in local optima, so we fit the model with various initializations and selected the highest scored model.

We have predicted the optimal sequence of internal hidden state using Viterbi Algorithm.

The following parameters were used in our Gaussian Mixture Emissions Hidden Markov Model[3] (GMM HMM) to find 3 different regimes/states in our credit spread time series:

Number of states in the model.	3
Number of states in the GMM.	2
Initial state occupation prior distribution.	1.0
Matrix of prior transition probabilities between states	1.0
Maximum number of iterations to perform.	10
Convergence threshold. EM will stop if the gain in log-likelihood is below this value.	0.01
Each state uses a prior diagonal covariance matrix	0.01

Table 3: GMM HMM Parameters

Using GMM HMM we find the following regimes:

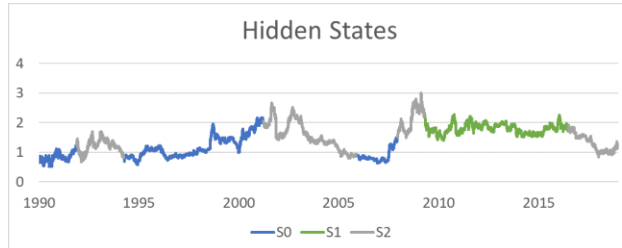

Figure 12: Resulting Regimes

	S0	S1	S2
# Days	2845	1877	2838
Mean	1.07%	1.79%	1.49%
Standard Deviation	0.35%	0.17%	0.46%

Table 4: Data for Regimes

Figure 12 shows how our 3 regimes split up our credit series. Table 3 shows the properties that each regime carries. Regime S2 shows a more volatile time period then the other 2 regimes and Regime S1 is the calmest.

6. Neural Network Models

a. Logic

Long Short-Term Memory (LSTM) Models[6][7][8] are very powerful at making predictions from time-series data for an arbitrary number of days ahead. They are a special kind of Recurrent Neural Network (RNN). Standard RNNs run into a problem with time series data, if the relevant information and the point it is needed is very far apart. This is called the long-term dependency problem. LSTM models take advantage of RNNs chain-like nature that is intimately related to time series, while they do not have the long-term dependency problem. LSTM models can remember the relevant information for long periods of time. They can do this by maintaining a cell state.

Information is added and removed to the cell state by “gates” of the LSTM cell. Gates regulate which information is kept or removed by using sigmoid neural net layers and pointwise multiplication. The first step the LSTM cell does is decide which information is going to be thrown away from the cell state. The next step is to decide which information is going to be added to the cell state. The third step is to finally update the cell state. Finally, the last step is outputting a filtered version of the updated cell state.

b. Method

We implement 4 different models each with a point-by-point approach and a multiple sequence approach. The first model used 1 feature: the credit spread time series and did not take into account the different regimes found from HMM. The second model also used only 1 feature: the credit spread time series but was trained specifically for the S2 regime. The third model used 6 input features: Credit spread, LIBOR, TED rate, Fed Fund rate, VIX, and S&P 500 and did not take into account the different regimes. Lastly, our fourth model used the same 6 input features but was trained specifically for the S2 regime. For each model, we trained an LSTM with a point-by-point approach and a multiple sequence approach. A point-by-point approach makes a prediction for the next day by using only the current day information. A multiple sequence approach makes a prediction for the next day by using a moving average over a certain number of previous days. In essence, the multiple sequence approach attempts to predict future momentum trends whereas the point-by-point approach attempts to just predict the next day.

Notice, our models fit into two categories: number of features and whether or not the model was trained on the S2 regime specifically (or across the whole data set). We want to answer these questions: Is it better to use just the credit spread series as input or should we also use other time series such as the Fed Fund rate, S & P 500, etc. as mentioned above? Will the LSTM model perform better if we train it on a specific regime or does the model already take this into account?

There are certain parameters that each of our LSTM model uses. To fit the best model, we found that using 3 neural net layers with 128 neurons, 20 epochs, and a dropout rate of 0.25 is best. The number of neural nets and neurons is an implementation detail of neural networks. The number of epochs is how many iterations the model goes through training. The dropout rate is a parameter that helps prevent the model from overfitting on the training data. It is a fraction of training data that is not included in the training through each epoch. Another way we prevent overfitting the model, is setting a low epoch count. If you train the model too many times, it becomes more and more overfitted to the training data.

Note, our point-by-point models predict one day ahead and our multiple sequence models make predictions every 21 days for the next 21 days[9].

c. Forecasting Results

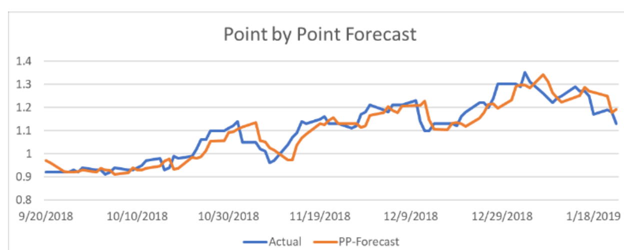

Figure 13: Univariate Model 1 with point-by-point approach

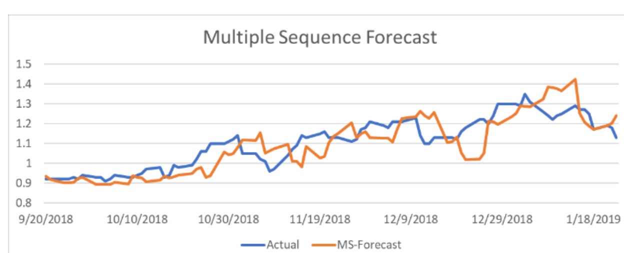

Figure 14: Univariate Model 1 with multiple sequence approach

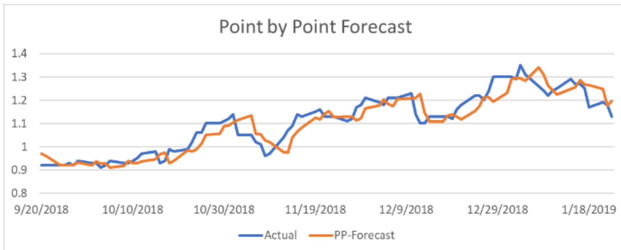

Figure 15: Univariate Model 2 with point-by-point approach

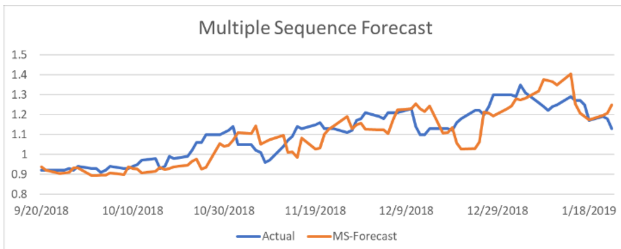

Figure 16: Univariate Model 2 with multiple sequence approach

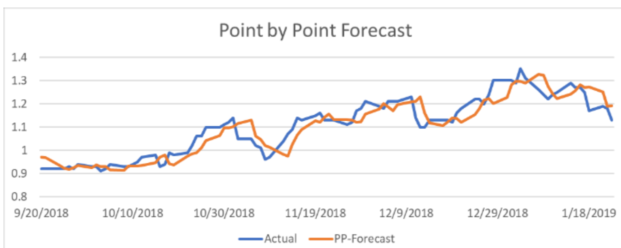

Figure 17: Multivariate Model 3 with point-by-point approach

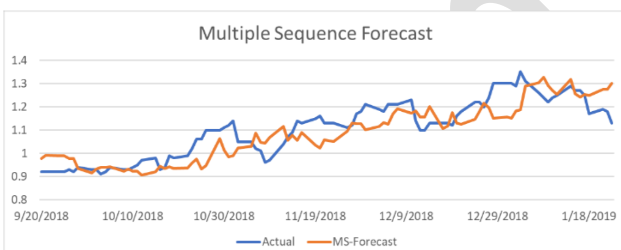

Figure 18: Multivariate Model 3 with multiple sequence approach

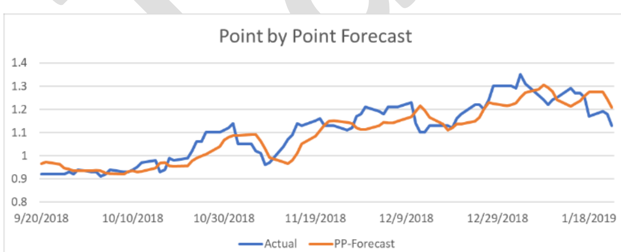

Figure 19: Multivariate Model 4 with point-by-point approach

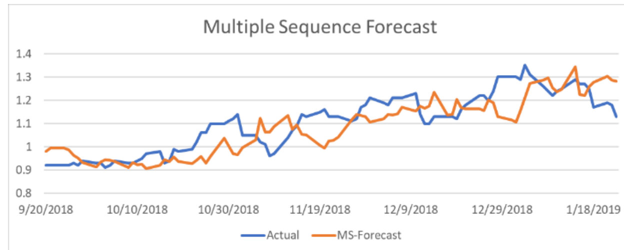

Figure 20: Multivariate Model 4 with multiple sequence approach

	Model 1		Model 2	
	LSTM		LSTM-S2	
	PP	MS	PP	MS
MSE	0.001184327	0.003858	0.002007	0.005825
MAE	0.026317856	0.046539	0.036076	0.06002
MAPE	1.8261%	3.2272%	3.2508%	5.3336%
SMAPE	1.8255%	3.2230%	3.2560%	5.3883%
Accuracy	59.2760%	60.0000%	55.5556%	57.7778%

	Model 3		Model 4	
	MVLSTM		MVLSTM-S2	
	PP	MS	PP	MS
MSE	0.0012	0.0039	0.0031	0.0065
MAE	0.0267	0.0477	0.0467	0.0641
MAPE	1.8550%	3.2853%	4.1540%	5.7116%
SMAPE	1.8528%	3.2733%	4.1902%	5.8133%
Accuracy	60.5430%	60.3620%	58.8889%	51.1111%

Table 5: LSTM Model Comparison

1) By analyzing Table 5 and the resulting graphs (Figures 13-20), we can see that point-by-point, is best in forecasting for one step ahead whereas the multiple sequence approach predicts better for long range forecasting.

2) All 4 models, with both point-by-point and multiple sequence approaches have similar predictive power in terms of both direction and magnitude.

3) We experimented with training LSTM models for the specific regime: S2. Our results show that the prediction power of the LSTM models which were trained specifically for that regime and the models that were trained across the whole time series, are the same. This implies that LSTM

models address the problem of hidden regime shifts in the credit spread time-series data.

4) We conclude that Model 1, where we use only one input feature (the credit spread time-series) and did not train it specifically for a single regime, performs the best.

d. Comparing ALL Models

Issues	Naïve Models	ARIMA Models	LSTM Models
Recognize Regime Shifts	No	No	Yes
Is a Black Box (not transparent)	No	No	Yes, neural net layers are not transparent
Subject to Overfitting to Training Data	Possible	Possible	Possible, but we have helped reduce chances of overfitting by using drop-out rate and less # of epochs
Capture Relationships between other features	No	Yes	Yes
Short Range Predictive Power	Good	Good	Good
Long Range Predictive Power	Bad	Okay	Good
Requires Stationary Data	No	Yes	No
Captures Non-Linear Relationship between features	No	No	Yes

Table 6: Model Comparison

7. CONCLUSION

In this paper, we have tackled forecasting credit spreads. We started out with naïve models that do not consider anything other than the average of the credit spread values from previous days. We then considered technical indicators to predict the credit spread, but they face the same problems as naïve models. Then we moved to ARIMA models, which predicts well, but to use that model, we have to assume that our data is stationary. In the financial field this is a tough assumption to justify. Also, ARIMA models cannot capture non-linear patterns. To make the data stationary and find different regime shifts we used Hidden Markov models. Then we used machine learning based recurrent neural networks to solve the problem of fitting non-linear patterns and the other drawbacks listed. The LSTM models have good predictive power, but it is tough for anyone to understand how it came up with its predictions; it is a black-box. At the end of the day, the data tells the story. Our models are a function of history and as such, should be used with caution. Some models are better at forecasting than others, but all our models do not consider future events and future policies which is crucial to the forecasting of credit spreads.

REFERENCES

[1] Board of Governors of the Federal Reserve System (US), 10-Year Treasury Constant Maturity Rate [DGS10], retrieved from FRED; https://fred.stlouisfed.org/series/DGS10. Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity [AAA10Y], retrieved from FRED; https://fred.stlouisfed.org/series/AAA10Y; https://fred.stlouisfed.org/series/AAA.

[2] S&P Dow Jones Indices LLC, S&P 500 [SP500], retrieved from FRED, https://fred.stlouisfed.org/series/SP500. ICE Benchmark Administration Limited (IBA), 3-Month London Interbank Offered Rate (LIBOR), based on U.S. Dollar [USD3MTD156N], retrieved from FRED, https://fred.stlouisfed.org/series/USD3MTD156N.

Federal Reserve Bank of St. Louis, TED Spread [TEDRATE], retrieved from FRED; https://fred.stlouisfed.org/series/TEDRATE. Chicago Board Options Exchange, CBOE Volatility Index: VIX [VIXCLS], retrieved from FRED; https://fred.stlouisfed.org/series/VIXCLS. Board of Governors of the Federal Reserve System (US), Effective Federal Funds Rate [DFF], retrieved from FRED; https://fred.stlouisfed.org/series/DFF. Federal Reserve Bank of St. Louis, Moody's Seasoned Aaa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity [AAA10Y], retrieved from FRED; https://fred.stlouisfed.org/series/AAA10Y.

[3] Hamilton, J.D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle, Econometrica, 57, 357–384.

Source code for identifying HMM regime

https://hmmlearn.readthedocs.io/en/stable/index.html

[4] Time Series Analysis and Its Applications with R Examples, Authors: Shumway, Robert H., Stoffer, David S.

[5] Box, G.E.P. and G.M. Jenkins (1970). Time Series Analysis, Forecasting, and Control. Oakland, CA: Holden-Day

R is used for ARIMA model. Package Used: 'astsa'

LSTM: was proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber.

[6] 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.

In 1999, Felix Gers and his advisor Jürgen Schmidhuber and Fred Cummins introduced the forget gate (also called "keep gate") into LSTM architecture, enabling the LSTM to reset its own state.

[7]Felix Gers; Jürgen Schmidhuber; Fred Cummins (1999). "Learning to Forget: Continual Prediction with LSTM". Proc. ICANN'99, IEE, London: 850–855

[8] https://en.wikipedia.org/wiki/Long_short-term_memory

[9] https://github.com/jaungiers/LSTM-Neural-Network-for-Time-Series-Prediction

[10]https://en.wikipedia.org/wiki/Mean_squared_error

[11]https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error

[12]https://en.wikipedia.org/wiki/Mean_absolute_error

[13]https://en.wikipedia.org/wiki/Mean_Directional_Accuracy

[14] https://medium.freecodecamp.org/machine-learning-mean-squared-error-regression-line-c7dde9a26b93