International Journal of xxxxxx
Vol. x, No. x, (20xx), pp. xx-xx

# Deep Learning based Forecasting of Credit Derivatives Indices

Rajneesh Kumar Singh

*WorldQuant University*
*singhrajneesh@gmail.com*

***Abstract***

*In this paper we apply Deep Learning architecture to predict the spread of credit default swap (CDS) on the North America Investment Grade index (CDX.NA.IG). The implemented Long-Short Term Memory (LSTM) model is compared with a baseline Support Vector Machine (SVM) model through the root mean squared error. We also test the hypothesis that enhancing the feature space of the LSTM model with a rolling time series data of Hurst Exponent improves the training data fit and test data forecast accuracy.*

***Keywords:*** *CDS Spreads, LSTM, Hurst Exponent, SVM, Hypothesis Testing*

## 1. Introduction

In his debut non-fiction book titled "The Greatest Trade Ever: How One Man Bet against the Markets and Made $20 Billion", American journalist Gregory Zuckerman narrates behind the scenes story of how John Paulson had pulled off the greatest trade in financial history. "By piling into complex "credit default swaps" against mortgages – in effect, insurance policies that would pay out if homeowners defaulted – his fund made an unthinkable $15bn (£9.8bn) in a year, $4bn of which he took home himself (Stewart, 2010)" and that dwarfed George Soros's billion-dollar currency trade in 1992. The 2015 movie "The Big Short" is based on the same Paulson trade.

Credit default swap (CDS) is a credit derivative instrument which is basically like an insurance on the default of the underlying reference entity. The reference entity could be a corporate or any sovereign. These are instruments which protects the buyer in case of default. In return the protection buyer pays a quarterly fee to the protection seller. Many such credit default swaps can be bundled together and form an index swap. For example CDX.NA.IG is one such portfolio of single-name credit default swaps on the 125 most liquid investment grade corporates in North America. European and other regional equivalents also trade actively. Further there are options and tranched products on these macro credit indices (Markit, 2021). For a more detailed understanding of the credit default swaps readers are referred to the Federal Reserve Board's discussion paper on the same (Bomfim, 2022).

Credit derivative products gained immense notoriety and have largely been blamed for the Great Financial Crisis of 2008-2009. These markets had largely been unregulated before that and is still opaque to many market participants. In an IMF working paper (Elliott, 2009) the authors highlight how credit derivative markets can increase systemic risk due to the inter-connectedness of large financial institutions and how policy makers need to be aware of the "blind-spots" in this market.

Recently the CDS of Russian Federation came into focus as it launched a full-fledged war against Ukraine in early 2022. We can in Figure 1.how the premium on the CDS sky-rocketed as the war progressed and there were fears that the Russian state would not honor the coupon payments on their outstanding bonds. Also notice the volatility around this CDS price.

ISSN: XXXX-XXXX IJXX

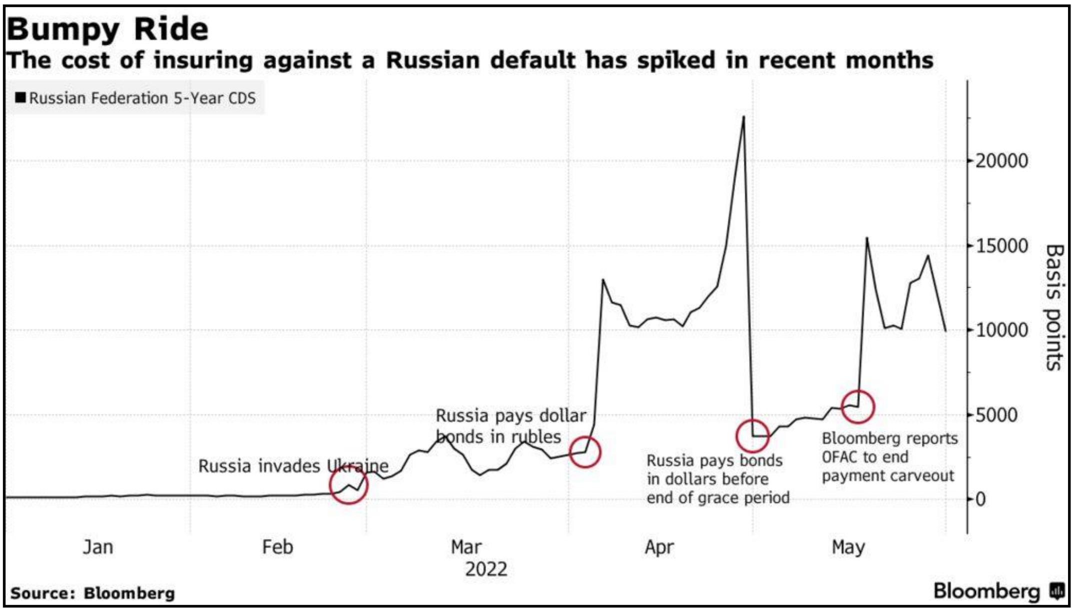



**Figure 1. Russia CDS during Ukraine War in 2022**

To implement a hedging and/or a trading strategy, the accurate forecasting of CDS spreads becomes important. This is useful not only for investors looking to gain from the price moves, it is important for risk managers looking to hedge counterpart risk and for policy makers framing a policy response to systemic risk after global markets.

We focus our forecasting task not at individual corporate or any country CDS level. Our focus is going to be credit indices which are like barometer of overall credit risk of whole economy just as the VIX index represents volatility measure for macro equity index, SPX. While other asset classes like equity, bond and foreign exchange get lot of attention with regards to new forecasting techniques using machine learning and deep learning, work on credit derivative asset class looks minimal in comparison. We look to bridge this gap with our study.

## 2. Related Work

While a lot of studies has been done on the determinants of CDS spreads, the literature is sparse on price forecasting. (Avino, 2014) did an analysis of two linear forecasting models, ordinary least squares and an AR(1) model as well as a Markov regime-switching approach and found some evidence of statistical predictability. His instrument of choice was the iTraxx CDS index and the data covered the 2008 financial crisis. iTraxx index is a pool of CDS on 125 investment grade corporates in Europe. They found the Markov switching model underperforming the linear models under study.

With the advent of machine learning models researchers have begun to apply them to the credit derivative markets. Support Vector Machines were applied by (Gündüz, 2011) to single-name CDS prices and were found to outperform the Merton model. This study is extended by (Zhang, 2018) where they predict the returns rather than the CDS prices and also adds a few features.

In an IMF working paper by (Hu, 2019) ensemble machine learning methods were applied on firm level accounting-based, market-based and macroeconomic variables to generate CDS spreads which can be helpful for arriving at CDS spreads for companies who don't have an active CDS market. The ensemble methods used were Bagging, Gradient Boosting and Random Forest.

Are CDS spreads predictable? And does efficient market hypothesis apply to the credit derivatives markets? With this dual goals (Vukovic, 2022) investigated the daily CDS spreads for 513 leading US companies over the time-period 2009-2020. The study splits the period as pre and post Covid-19 to check if there were changes in the market efficiency. The forecasting tools they used were Support Vector Machines, Group Method of Data Handling, Long Short-Term Memory and Markov switching auto regression. The results show the prediction results are not very different before and during Covid-19. But they did find that the market became less efficient during the pandemic. The GMDH and MSA models outperformed SVM and LSTM.

As we can see the latest machine learning and deep learning models for time series forecasting of credit default swaps spreads has been able to provide better results compared to traditional methods. This is consistent with the findings in other asset classes where researchers have been reporting superior results. This also means that efficient market hypothesis may not hold for CDS price movements and the moves may not be mere random walks. Fractal Market Hypothesis (FMH) could be an alternative for CDS prices especially during periods of market turbulence.

FMH studies have been conducted on various markets using a variety of methods to establish the fractal properties of the time series. Using the CDS spreads of Turkey, Russia, South Africa and Brazil, (Günay, 2016) examine the long-memory dependency in its volatility. Hurst Exponent Analysis was used by (Balkan, 2022) on the CDS spreads for 34 OECD countries between March 2003 and February 2020. Using rescaled range analysis with four different frequencies the researchers were able to show persistency in all CDS spreads and therefore upholding the FMH.

The pricing of CDS under Generalized Mixed Fractional Brownian motion has been considered by (He, 2014) and they provide a closed-form analytical expression for the CDS under risk-neutral assumption. The long memory in highly volatile time series of cryptocurrencies is examined using Hurst exponents of log returns in (Sheraz, 2022) as this nascent market has demonstrated sufficient divergence from normal distribution assumption.

## 3. Methodology

Our literature survey has provided evidence of long-term memory in CDS prices from two independent approaches that researchers have taken: fractal markets and using deep learning model like LSTM. This motivates us to combine the two approaches and see if better forecasting results can be achieved.

Broadly speaking we are curious to see if a data driven model like LSTM performs better than traditional model driven SVM and if Hurst Exponent Analysis on CDS spreads helps with forecasting accuracy. To our knowledge there are no study using Hurst Exponent and machine learning/deep learning at the same time. With this dual objectives in mind we would like to test a couple of hypotheses as below.

**Hypothesis I:** LSTM model performs better than traditional SVM in fitting and predicting CDS spreads.

**Hypothesis II:** The forecast accuracy improves when we incorporate a rolling Hurst Exponent series as an additional input feature.

This paper is organized as follows: in the next section, we review our dataset and present a statistical summary of the CDX.NA.IG CDS spreads; we describe our methods: LSTM and the baseline SVM model; next we will present our training and forecasting results with root mean squared error estimates and demonstrate the performance of each model; and lastly we will provide the summary and concluding remarks of our study.

### 3.1. Data

CDX.NA.IG index consists of 125 most liquidly traded CDSs in North America. We source the 5Y spread on this index for the last 10 years using Bloomberg terminal. Our data covers the period from January 2012 to December 2022 and 4018 observations in total. We will do a detailed study of the data patterns to check the suitability of using our proposed LSTM and SVM algorithms. Our exploratory data analysis will focus on identifying the time series properties of the data set. In particular statistical analysis will be done for identifying presence/absence of trends, cycles, seasonality and clustering in the data. We will test the time series of stationarity as well.

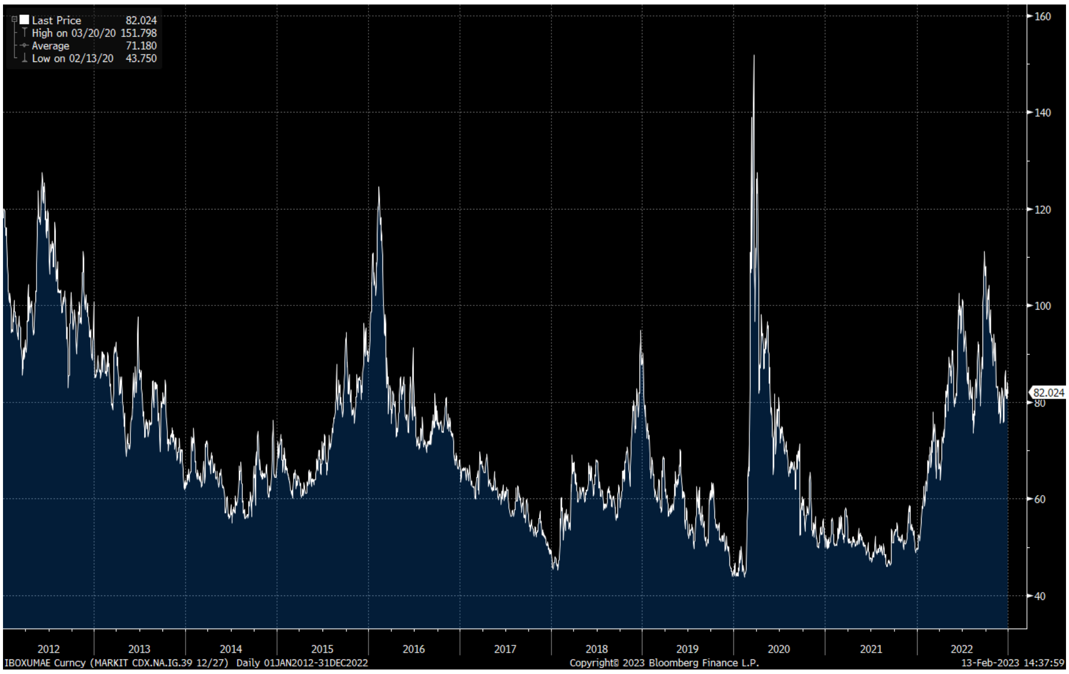



**Figure 2. CDX.NA.IG Historical Spread**

We find higher volatility in CDX.NA.IG returns compared to the S&P 500, while volatile market periods are notable in both, as shown in Figure 3. At the same time Figure 4 do tend to show that the returns deviate from normal distribution.

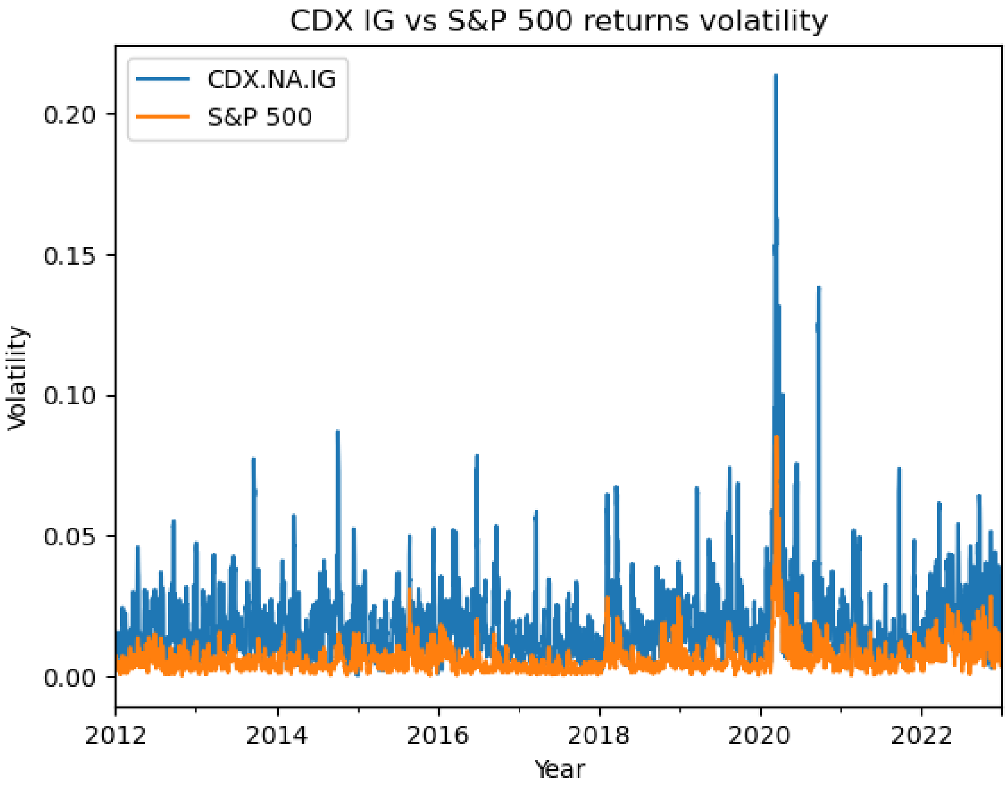



**Figure 3. CDX.NA.IG vs S&P Index Returns Volatility**

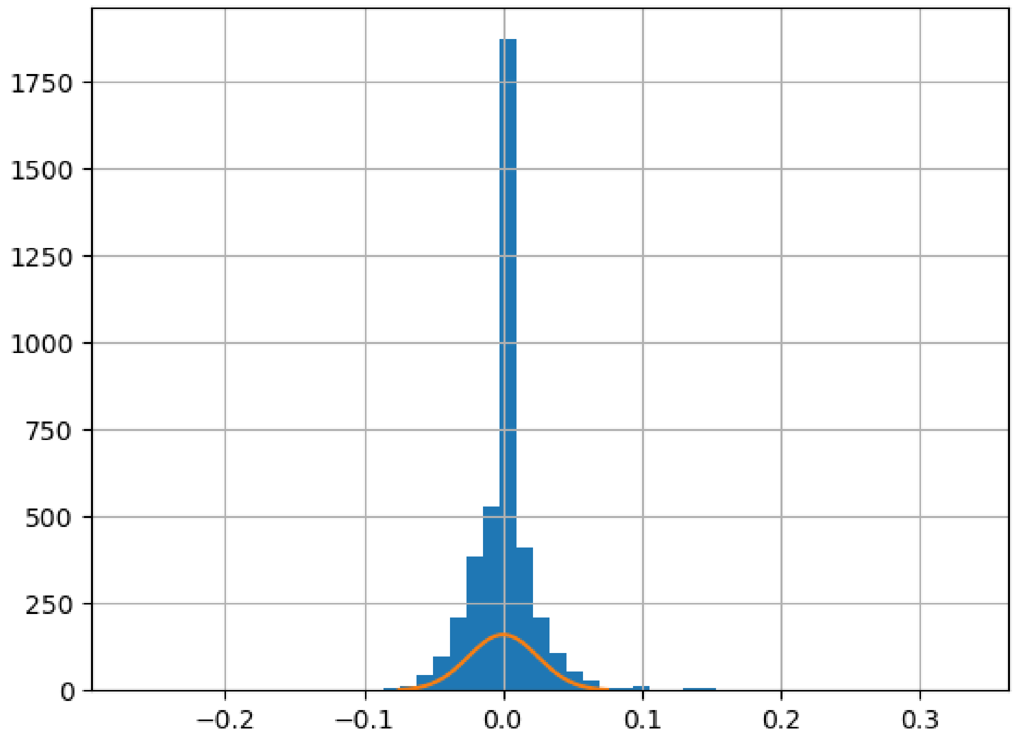



**Figure 4. CDX.NA.IG Returns Distribution**

```
count    4018.000000
mean       71.199974
std        17.428001
min        43.750000
25%        58.440000
50%        66.929000
75%        81.716750
max       151.798000
Name: IG_MID, dtype: float64
```

**Table 1. Summary statistics for the historical CDX.NA.IG spreads**

We perform an Augmented Dickey-Fuller (ADF) test for stationarity in spread and returns series. As seen in Figure 5 our p-value comes less than 0.05 which means we reject the null hypothesis that the data does not have a unit root and is therefore stationary.

### CDX.NA.IG: Stationarity Study

```
In [44]: from statsmodels.tsa.stattools import adfuller

         def check_stationarity(series):
             # Copied from https://machinelearningmastery.com/time-series-data-stationary-python/

             result = adfuller(series.values)

             print('ADF Statistic: %f' % result[0])
             print('p-value: %f' % result[1])
             print('Critical Values:')
             for key, value in result[4].items():
                 print('\t%s: %.3f' % (key, value))

             if (result[1] <= 0.05) & (result[4]['5%'] > result[0]):
                 print("\u001b[32mStationary\u001b[0m")
             else:
                 print("\x1b[31mNon-stationary\x1b[0m")

In [45]: check_stationarity(df['IG_MID'].pct_change().dropna())

         ADF Statistic: -17.579249
         p-value: 0.000000
         Critical Values:
                 1%: -3.432
                 5%: -2.862
                 10%: -2.567
         Stationary

In [46]: check_stationarity(df['IG_MID'])

         ADF Statistic: -3.471254
         p-value: 0.008757
         Critical Values:
                 1%: -3.432
                 5%: -2.862
                 10%: -2.567
         Stationary
```

**Figure 5. Code snippet for testing Stationarity of CDX.NA.IG spreds**

Since our ADF test shows that the returns of the time series is stationarity we proceed with the next step to check the autocorrelation. This is correlation with a lagged version of the time series. The Autocorrelation Function (ACF) plot in Figure 6 can help find the order of autocorrelation in the time series.

```
In [43]: from statsmodels.graphics.tsaplots import plot_acf
         from statsmodels.graphics.tsaplots import plot_pacf
         f, ax = plt.subplots(nrows=2, ncols=1, figsize=(10, 6))
         plot_acf(df['IG_MID'].pct_change().dropna(), lags=50, ax=ax[0])
         plot_pacf(df['IG_MID'].pct_change().dropna(),ax=ax[1], method='ols')

         plt.tight_layout()
         plt.show()
```

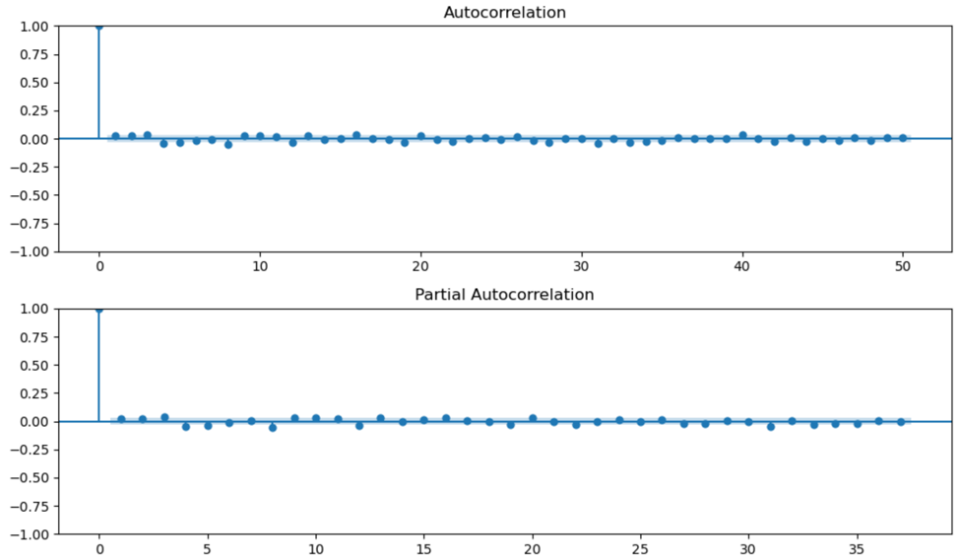



**Figure 6. Code Snippet for Testing Autocorrelation of CDX.NA.IG Returns**

We do not find strong evidence of autocorrelation of the CDX.NA.IG spread returns from the ACF and PACF plots in Figure 6 but at the same time a visual inspection of the spread changes time series do show clustering of returns. As noted from Figure 7 we can see large spread changes are clustered around few macroeconomic events of which the COVID-19 period stands out.

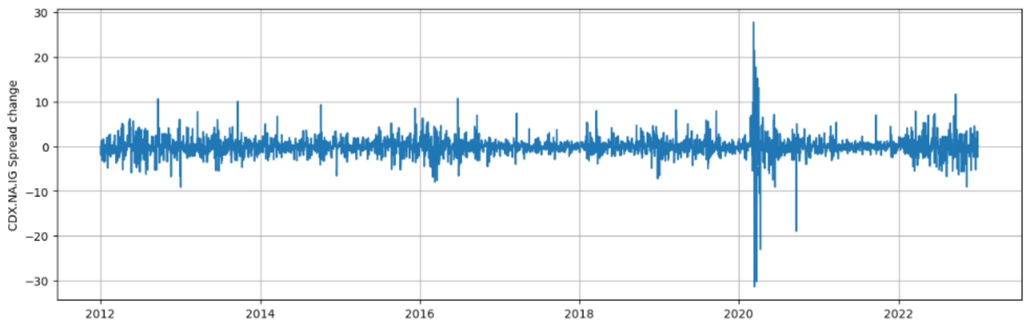



**Figure 7. Clustering of CDX.NA.IG Returns**

After doing a sub-period study for seasonality, we find little evidence of it as seen in the Figure 8.

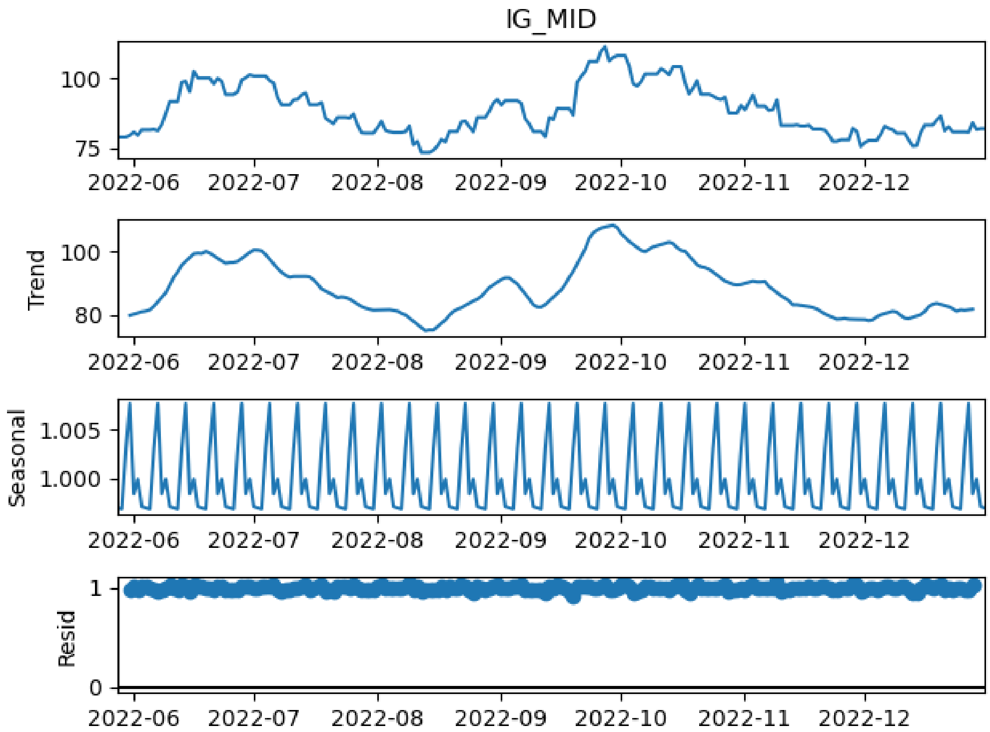



**Figure 8. Seasonality of CDX.NA.IG Returns**

### 3.2. Models and Training

We apply a deep learning based architecture, LSTM model, to the prediction of CDS spreads of the CDX.NA.IG index and investigate model performance versus a benchmark SVM model. We will use Mean Squared Error (MSE) as the primary performance metric. We use MSE as loss function because we strive to minimize the difference between predicted return and true return. The RMSE is formulated as

$$RMSE = \left(\frac{1}{T}\sum_{t}(\hat{y}_t - y_t)^2\right)^{\frac{1}{2}}$$

We will also focus attention on our hyper parameter optimization to arrive at the optimal values for the LSTM model parameters: number of epochs, learning rate, decay rate, batch size and look-back window. Adam is our choice of optimizer in the LSTM algorithm.

In order to test our two hypotheses we plan to conduct two set of prediction tasks:

1. LSTM and SVM forecasting with just the CDX.NA.IG time series data input.
2. LSTM and SVM forecasting with a time series of rolling Hurst Exponent values as an additional input feature.

The pseudo code for calculation of Hurst Exponent time series is presented in the appendix. A rolling time series of the Hurt Exponent will be appended to the CDX.NA.IG time series and will serve as additional input feature.

**3.2.1. Support Vector Machines:** SVM is a class of supervised machine learning and is often used for classification tasks but has found application in regression analysis. The basic idea behind SVM is to find a hyperplane that maximizes the distance between classes.

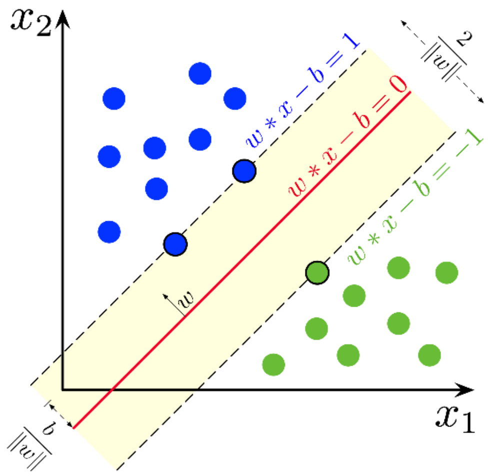



**Figure 9. Maximal Margin Hyperplane for SVM**

The solid red line is an example for a maximal margin hyperplane. The two blue and one green point on the dashed margin lines are the support vectors and the classification result depends only on these three points. The position of other data points do not affect the classification results. The support vector machine is an extension of the above classifier scheme where kernels are used to enlarge the feature space.

*Data transformation and hyper-parameters tuning:* For support vector regression we cannot use the dates as a feature. Therefore we transform the dates as integers and use it as our feature for the regression problem at hand. We split our data as training and test set and using the training se we perform a grid-search cross validation technique to arrive at the optimal values of our hyper-parameters namely *C=1000, eplison=0.01 and gamma=0.1* . Figure 10 shows the impact of *gamma* value to the overall fit of the model. As we can see the model is very sensitive to the *gamma* parameter. Small values of *gamma* means the model is not able to capture the complexity of the shape of the data.

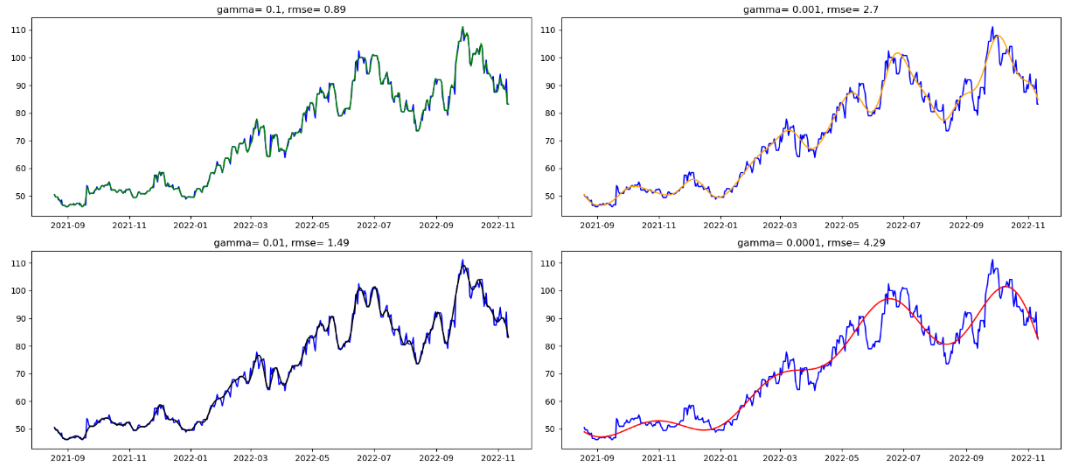



**Figure 10. Gamma sensitivity in SVM fit**

**3.2.2. Long Short-Term Memory:** LSTM is a type of Recurrent Neural Network where dependency on the past data can be modelled. Unlike the usual feedforward neural network, LSTM models have a feedback connection which can learn time dependent relationship in the data and are therefore more suitable for time-series data modelling. Another advantage of these class of models is its ability to overcome the vanishing gradient problems seen in traditional neural networks. Currently LSTM dominates forecasting problems in most of the asset classes and we intend to study its utility to credit derivatives through this study.

Figure 11 details a single unit of a LSTM architecture. In the context of modelling LSTM for time-series data each timestamp data is a single LSTM unit which takes the input at time *t* and process the data and passes on the output to the next LSTM unit. The forget gate controls what information doesn't get passed to the next unit and the input gate stores the information deemed to be important.

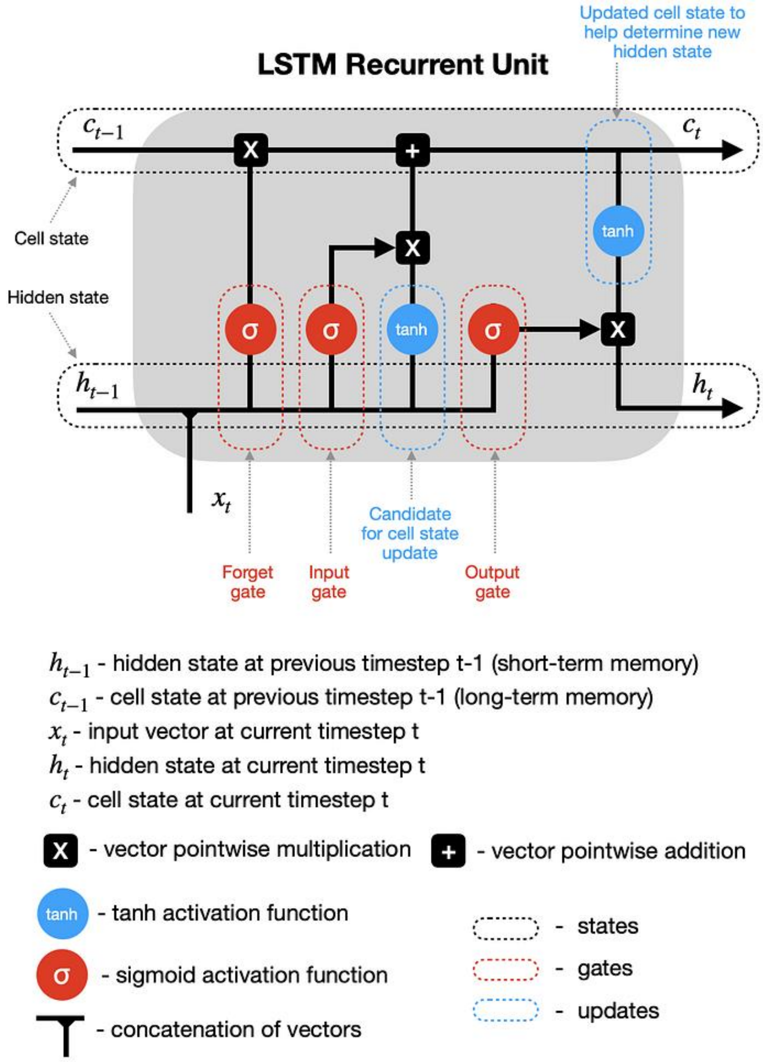



**Figure 11. A single LSTM recurrent unit** (Dobilas, 2022)

*Data transformation and hyper-parameters tuning:* We first normalize our CDX.NA.IG time series data and split it into a training and testing set. To feed our data to the LSTM model we transform our normalized data so that for each time *t* we provide an array of data for a chosen look back period. For example if we take a look back period of 5 days we build our feature data set with CDS spreads for the last 5 days and the target spread as the spread for the next day. In a way the LSTM model uses the CDS data for the past look back periods and fits the model to next day spread. This optimal look back period is arrived at through our hyper-parameter tuning process. We try a look back periods of [1, 4, 10, 25, 50] days and choose the look back period as 1 which has the least RMSE as seen in Figure 12. For this we fixed our epochs as 250 and batch_size as 64, which itself is arrived through a similar RMSE based hyper-parameter tuning process whose results are depicted in Figure 13 and 14 respectively. The optimization for epochs was run on [10, 50, 100, 250, 500, 750, 1000] and for the batch_size on [8, 32, 64, 96, 128, 256]. A dropout rate of 0.3 was fixed for the LSTM model setup. The popular *adam* optimizer and *mean_squared_error* loss metric was used.

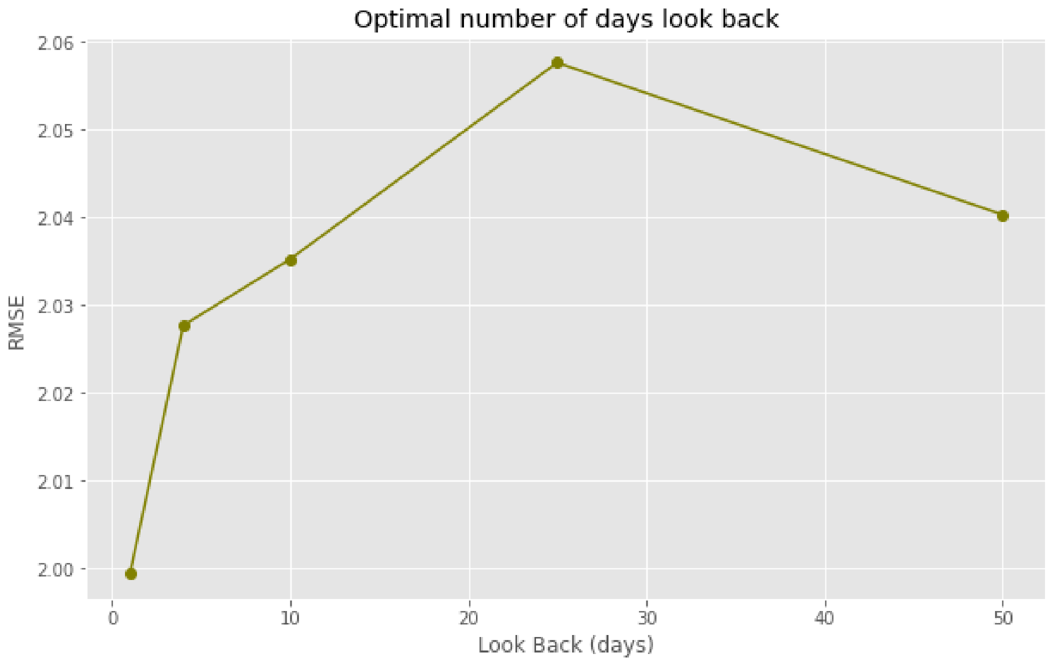



**Figure 12. Optimal look back period for LSTM model**

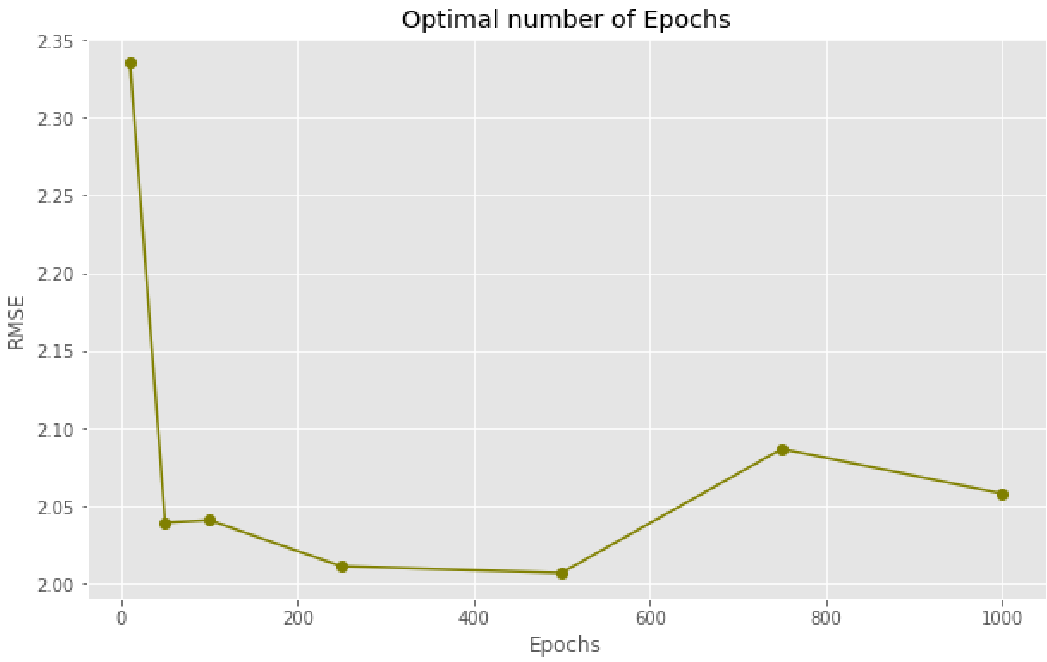



**Figure 13. Optimal number of epochs for LSTM model**

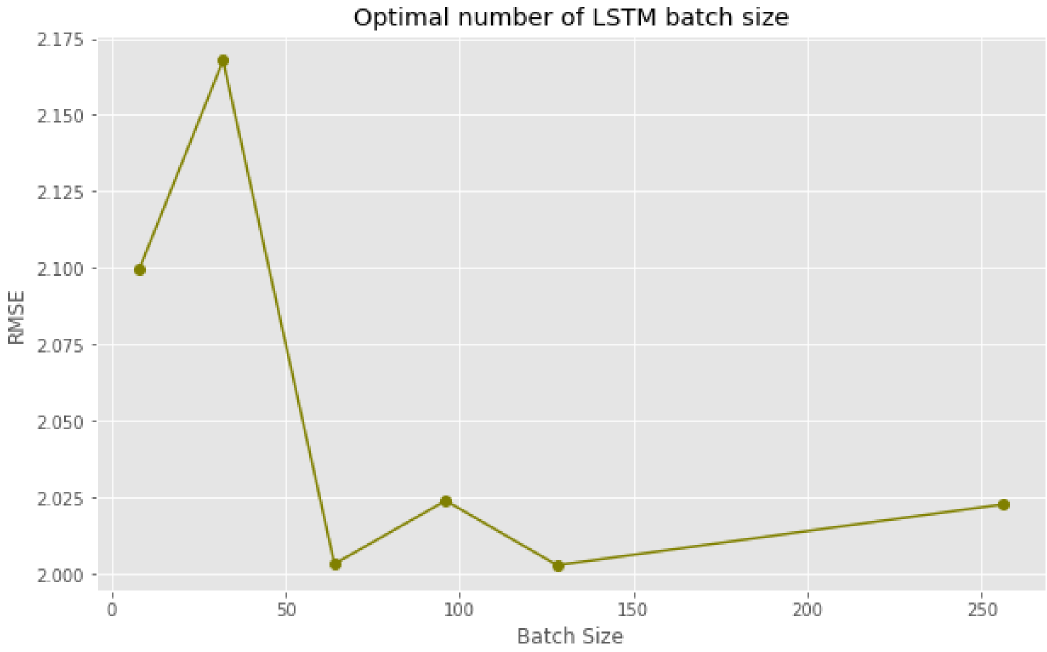



**Figure 14. Optimal batch size for LSTM model**

## 4. Results and Discussion

At the end of the hyper-parameter tuning process we arrive at the final parameters of our LSTM model with look back period as 1, epochs as 250 and batch size as 64. Figure 15 show a very good fit over the training and testing data and also a good tradeoff between bias and variance.

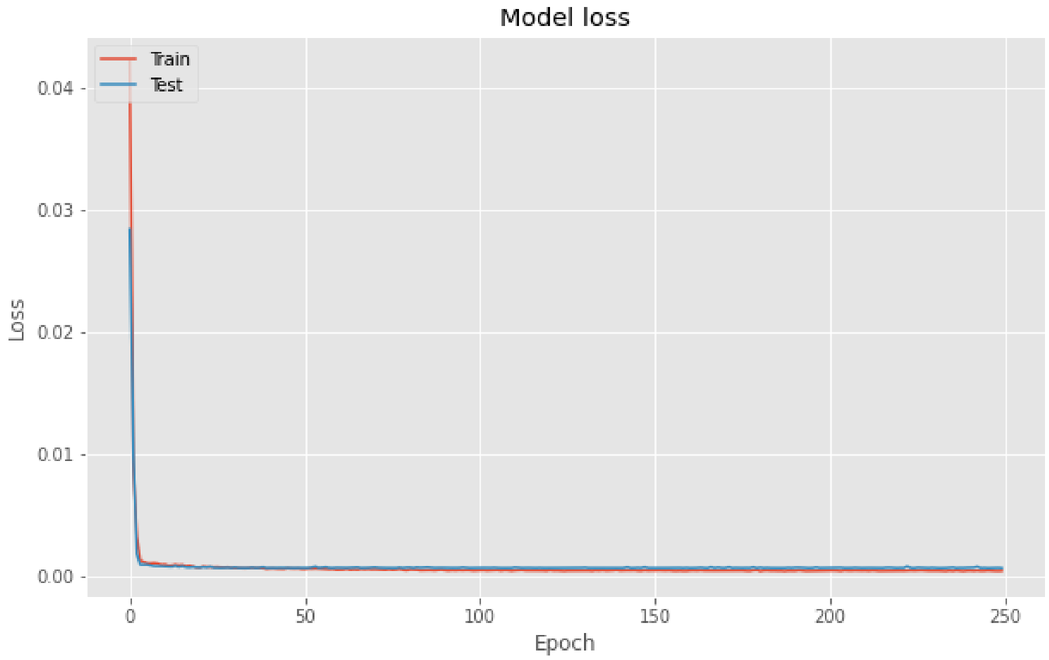



**Figure 15. Training vs test data model loss of the LSTM model**

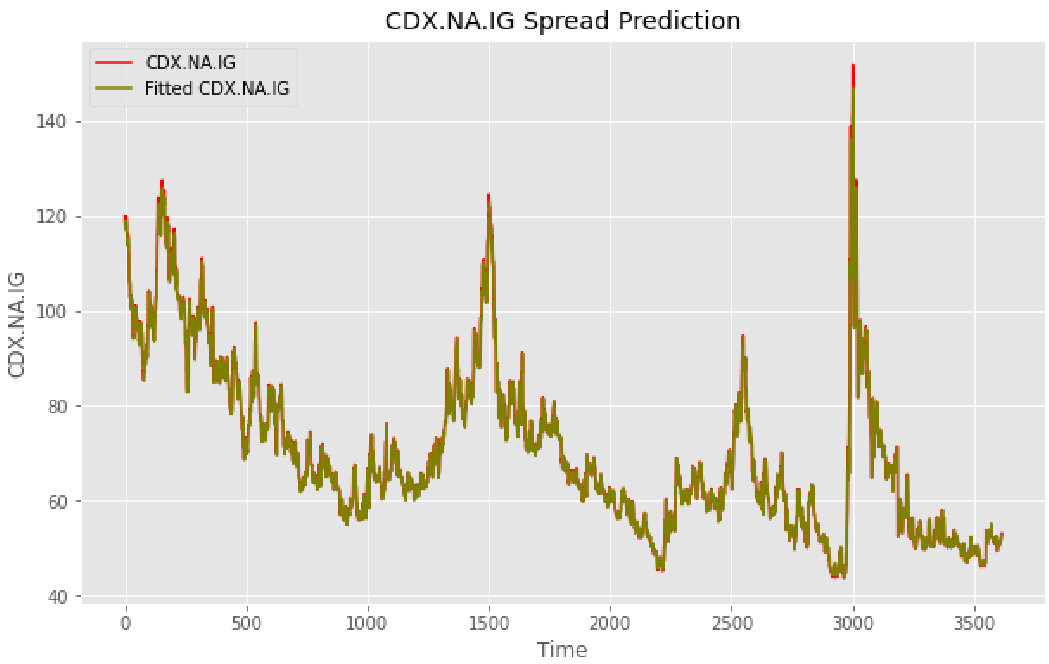



**Figure 16. Training data prices vs fitted prices of the LSTM model**

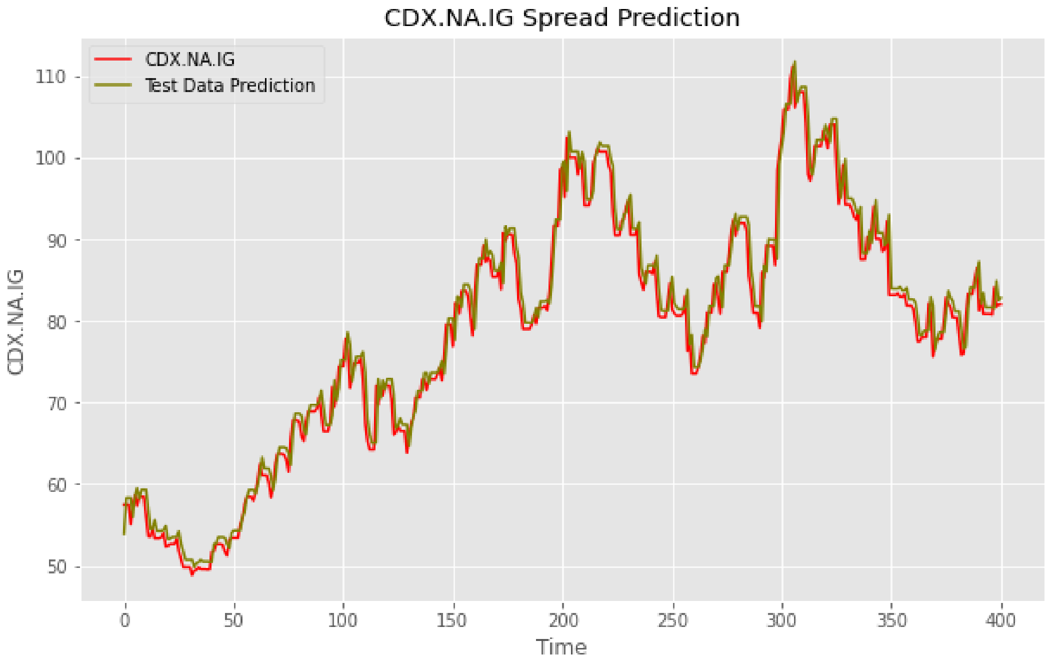



**Figure 17. Test data prices vs predicted prices of the LSTM model**

A visual inspection of Figure 16 and 17 for the fit vs actual prices for the training data forecast vs actual prices for the testing data does show that the LSTM model is able to achieve a good fit and forecast for the CDX.NA.IG spreads for the period under study.

Below we reproduce our two hypothesis that we testing in this study.

**Hypothesis I:** LSTM model performs better than traditional SVM in fitting and predicting CDS spreads.

**Hypothesis II:** The forecast accuracy improves when we incorporate a rolling Hurst Exponent series as an additional input feature.

Our first hypothesis that the LSTM model does a better job at forecasting the CDX.NA.IG spreads seems to be true as evidenced by a relatively low RMSE value of 2.217 that it achieves for the testing period. Similar metric for the baseline SVM model stands at 16.973. We also find that the SVM model seems to over fit the training data and ends up with very variance in the testing data. The temporal dependencies in the data is not being captured by the SVM model.

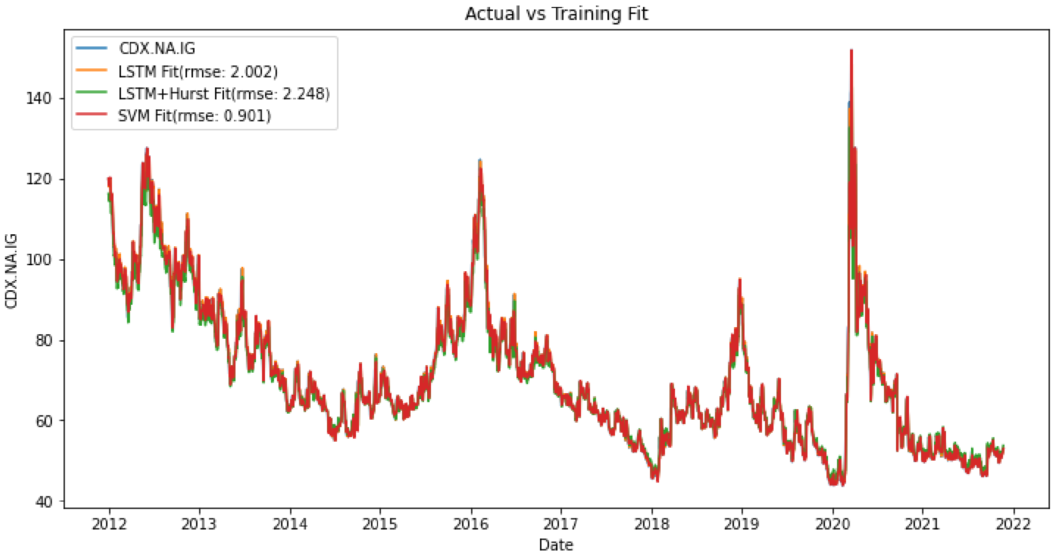



**Figure 18. Training data prices vs predicted prices for the three model**

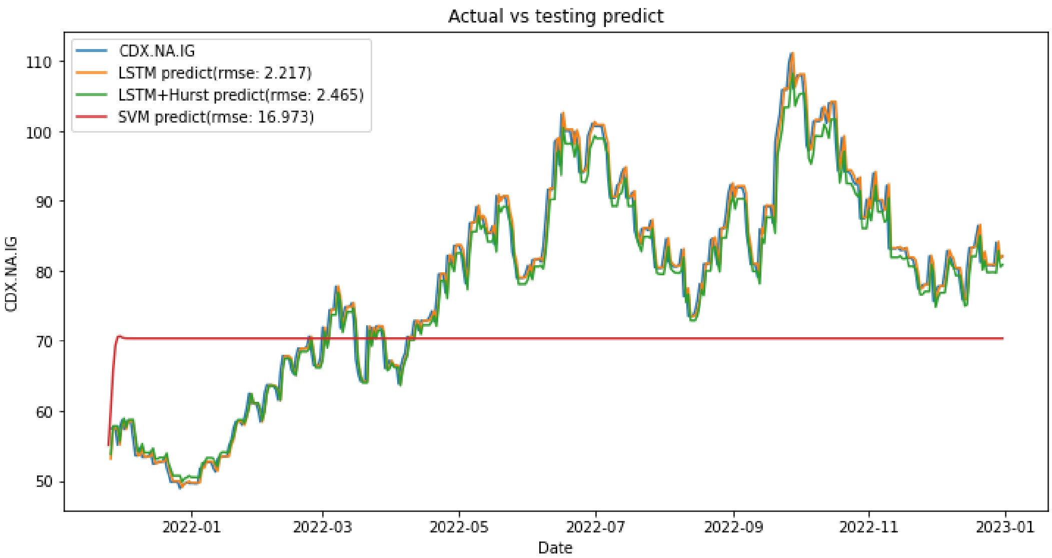



**Figure 19. Test data prices vs predicted prices for the three model**

The result of our second hypothesis comes out to be false. We tested if the presence of the rolling time series of Hurst Exponent would be able to achieve better fit in training data and/or better forecasting accuracy over the testing data. The RMSE for our LSTM + Hurst model comes out slightly worse vs the LSTM model. The RMSE compares as 2.002 vs 2.248 for the training data fit and 2.217 vs 2.465 for the testing data forecast for our LSTM and LSTM + Hurst models respectively. Figure 20 for the CDX.NA.IG vs rolling Hurst

Exponent does seem to show high correlation and would seem to improve the LSTM model performance. Our results do not find evidence to support the hypothesis.

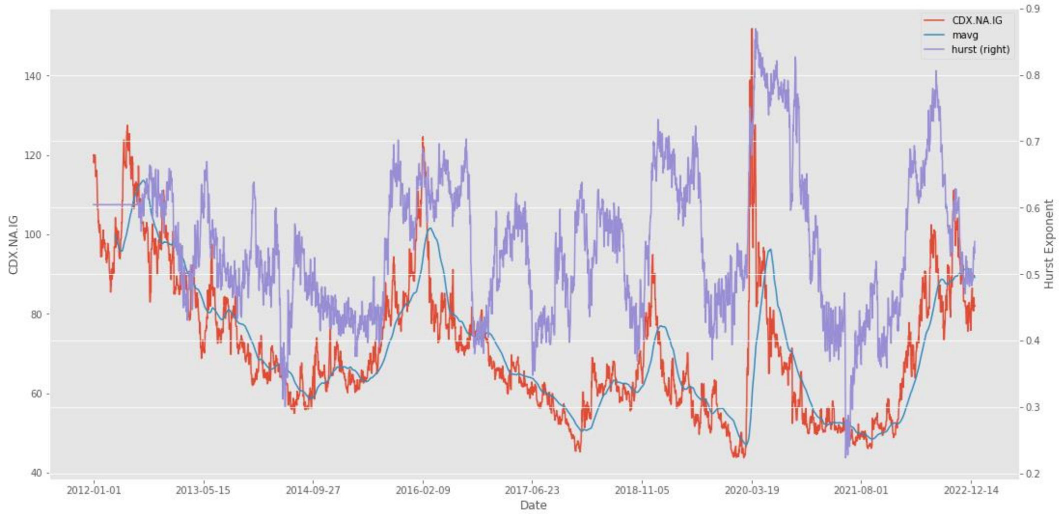



**Figure 20. Hurst Exponent vs CDX.NA.IG spreads**

## 5. Conclusion

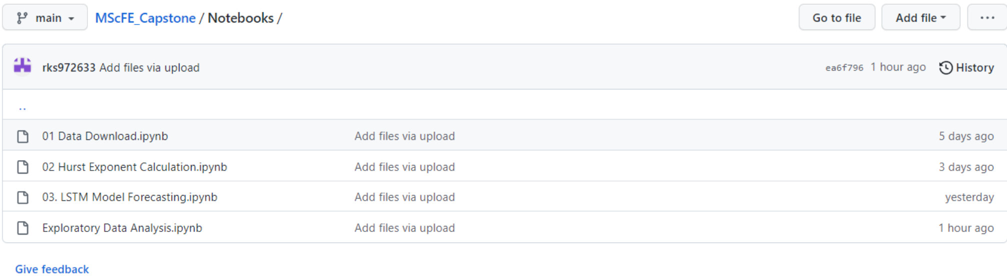
Neural network based deep learning architecture has found immense application in forecasting time series of financial data. Compared to other asset classes work in the area of credit derivatives is limited. Credit derivatives as an asset class has been in the limelight post the Great Financial Crisis of 2008-09 where it was blamed for causing outsized losses in banking books and the eventual bail-out by Central Banks across the globe. In our study we shown that as in other asset class forecasting, LSTM model does a better forecasting compared with a baseline SVM model. However we could prove our other hypothesis about enhancing the LSTM model prediction with an augmented feature space of a rolling time-series of Hurst Exponent.

## Appendix

### A. GitHub Repository

An online repository for the current project is publicly available at the below GitHub address. The plan is to provide all the files and working codes to the online repository. Currently four python notebooks are posted to the repository together with input data for replication of reported results.

https://github.com/rks972633/MScFE_Capstone




## B. Pseudocode

Below algorithm for calculation of Hurst Exponent is motivated from (Sheraz, 2022).

**Algorithm 1**. Hurst Exponent Computations

1. Suppose the given time series has length $N$. In the first step, convert the original time series to a log return series;
2. Divide the entire data series into "$A$", several contiguous sub-periods $I_a$ , with $a = 1, 2, \ldots, A$. Each element in $I_a$ is labeled as $N_{k,a}$, where $k = 1, 2, \ldots, n$. For each $I_a$, compute the average $e_a$;
3. Create a series of accumulated departures $D_{k,a}$ from the mean value for each sub-period by defining $D_{k,a} = \sum_{i=1}^{k} (N_{i,a} - e_a)$;
4. Define the range as maximum minus the minimum value of accumulated departures within each sub-period by $R_{I_a} = \max(D_{k,a}) - \min(D_{k,a})$;
5. Calculate the sample standard deviation $S_{I_a}$ of each sub-period $I_a$;
6. Normalize each range by dividing by the sample standard deviation. Therefore, the rescaled range for each $I_a$ is equal to $R_{I_a}/S_{I_a}$;
7. Compute the average of $\left(\frac{R_{I_a}}{S_{I_a}}\right)_n = (1/A)\sum_{a=1}^{A}\left(\frac{R_{I_a}}{S_{I_a}}\right)$;
8. Following the [48] Hurst exponent can be obtained using Equation (10). Estimate the Hurst exponent by running OLS regression, taking logarithm values of the series.

## Acknowledgments

I would like to thank Dr. Tiberiu Stoica at WorldQuant University for the guidance provided throughout the conduct of this study.

## References

[1] D. Avino and O. Nneji, "Are CDS spreads predictable? An analysis of linear and non-linear forecasting models", International Review of Financial Analysis, vol. 34, **(2014)**, pp. 262-274.

[2] E. Balkan and U. Uyar, "The Fractal Structure of CDS Spreads: Evidence from the OECD Countries", Romanian Journal of Economic Forecasting, vol. 25.1, **(2022)**, pp. 106.

[3] Bomfim, A. N. **(2022)**. Credit Default Swaps. Finance and Economics Discussion Series 2022-23. Retrieved from https://doi.org/10.17016/FEDS.2022.023.

[4] Elliott, M. J. **(2009)**. Credit derivatives: systemic risks and policy options? International Monetary Fund.

[5] Günay, S. a. **(2016)**. Long-memory in volatilities of cds spreads: evidences from the emerging markets. ournal for Economic Forecasting 1, 122-137.

[6] Gündüz, Y. a.-H. **(2011)**. Predicting credit default swap prices with financial and pure data-driven approaches. Quantitative Finance 11.12, 1709-1727.

[7] Gupta, R. G. **(2020)**. Retrieved from https://streetfins.com/analyzing-bill-ackmans-2-6-billion-cds-trade/.

[8] He, X. a. **(2014)**. The pricing of credit default swaps under a generalized mixed fractional Brownian motion. Physica A: Statistical Mechanics and its Applications 404, 26-33.

[9] Hu, N. J.-C. **(2019)**. Completing the Market: Generating Shadow CDS Spreads by Machine Learning. International Monetary Fund.

[10] Markit, I. **(2021)**. Retrieved from https://cdn.ihsmarkit.com/www/pdf/1221/CDS-Indices-Primer---2021.pdf.

[11] Sheraz, M. S. **(2022)**. Volatility Dynamics of Non-Linear Volatile Time Series and Analysis of Information Flow: Evidence from Cryptocurrency Data. Entropy 24.10, 1410.

[12] Stewart, H. **(2010, March 07)**. Retrieved from https://www.theguardian.com/books/2010/mar/07/the-greatest-trade-ever-by-gregory-zuckerman-review-heather-stewart.

[13] Vukovic, D. B. **(2022)**. Are CDS spreads predictable during the Covid-19 pandemic? Forecasting based on SVM, GMDH, LSTM and Markov switching autoregression. Expert systems with applications.

[14] Zhang, T. Y. **(2018)**. Predicting Credit Default Swap (CDS) Returns with Machine Learning. Available at SSRN 3152143.

## Authors

**Rajneesh Kumar Singh,** is a current student at WorldQuant University pursuing his Master of Science in Financial Engineering. He has been a financial markets professional for over 15 years and has mostly worked in the field of Credit Derivatives in various roles. Currently he works as a Credit Quantitative Research in a London based credit hedge fund.