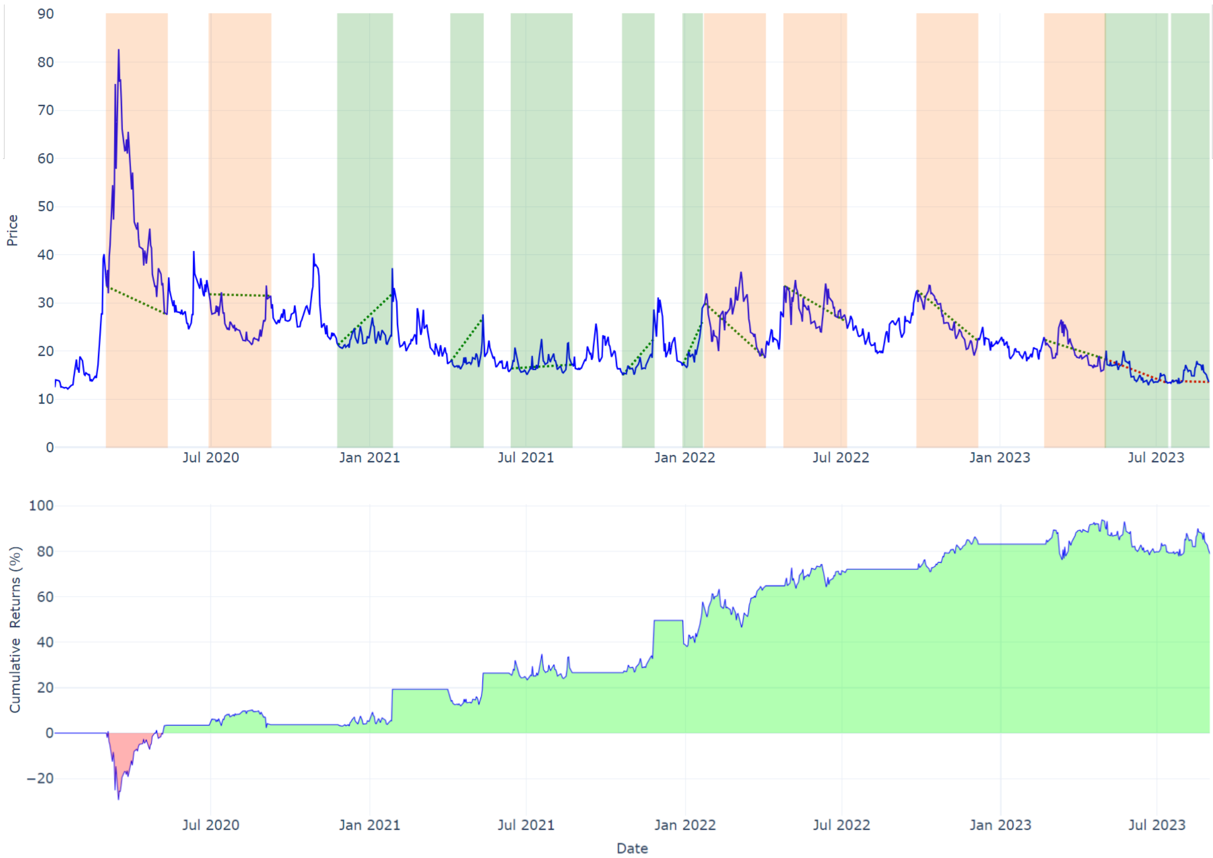
VIX VERTEX PREDICTIONS

ITU Research Project (7.5 ECTS)

Author/Student:
Christian Emil Haldan
cemh@itu.dk

Supervisor and Course Manager:
Luca Maria Aiello
luai@itu.dk



15-12-2023

Abstract

This study investigates the use of Long Short-Term Memory (LSTM) neural networks for predicting the Volatility Index (VIX), focusing on two main aspects: the comparative information gain of transformed technical indicators versus Open-High-Low-Close (OHLC) data. Additionally, we seek to find the efficacy of a custom loss function specifically designed to weigh predictions of high target values. As part of this research, we also develop a novel target variable, aimed at defining a floating points values to describe a local price ceiling or floor, relative to the current price. We evaluate model performance through loss scores and trading simulations, emphasizing cumulative and average returns. As all elements of the project work in symbiosis, we dubbed it ”VIX Vertex Predictions”. While the VIX is not a trade-able index, this research aims to show the predictive capability of the different input and model configurations. Throughout the report, we strongly emphasize that the trading simulator is merely a tool for simplifying evaluation. Findings reveal that technical indicators initially worsened the loss score's compared to models using OHLC data. While running trading simulations, the combination of technical indicators and the custom loss function outperformed other combinations on the test set, however, the mixed results in the of the combinations can only suggests a limited yet potential information gain from technical indicators when paired with the custom loss function. The study aims to contribute to the field of financial forecasting with the use of time series neural networks, while highlighting future research avenues for enhancing predictive accuracy for the volatility index.

The GitHub repository for this project can be found here [1]

1 Introduction

The financial market, has always attracted intense scrutiny and the desire to predict its movements. In the realm of market predictions, there has been a longstanding pursuit to forecast market trends, typically focusing on point-wise prediction methods I.E predicting specific future prices for given time frames. A comprehensive meta study of different research papers based on deep learning projects for financial forecasting (Zou, 2022) confirms the previous claim, as the methodology typically revolves predicting the future price using using price data. Lacking in this meta study, is research on forecasting the Volatility Index, or (VIX)...

The VIX is traditionally used to hedge long portfolio's in the events of market crashes, accepting that their option contract is likely to void, at the benefit of security in the event of a market crash. An article from the SteadyOptions Trading Blog (SteadyOptions, 2023) confirms this by stating; ”VIX is a great way to hedge your long portfolio. It is a well known fact that during severe market downturns, VIX spikes significantly, which can offset some of your portfolio losses. However, you cannot trade VIX directly. There are few ways to trade VIX”. That said, the VIX serves as a critical tool in financial decision-making for investors.

The transition from manual decision making to algorithmic approaches highlights a shift in market analysis paradigms. As shown in the meta analysis (Zou, 2022), models capable of self-learning and adapting offer a new frontier in financial forecasting, which in combination with domain knowledge could provide better insights for financial forecasting. Despite the advancements in algorithmic techniques, there is a notable gap in public knowledge for predicting macro trends of the volatility index, such as market peaks and valleys. Common practices often involve inputting Open-High-Low-Close (OHLC) data to forecast subsequent day prices, pointing towards a need for more nuanced approach for defining targets and features.

At the glance of an eye, the VIX appears to have a harmonic-like motion of the price movement figure 1, which was one of the key motivations for this research paper. This paper introduces approach in market trend prediction on the volatility index called ”Vertex prediction”. We introduce a target variable aimed at capturing the relative distance to a favorable market position, whether it being a peak or a valley. This approach shifts the focus from short-term price movements to understanding broader market trends. Long Short-Term Memory (LSTM) networks, a type of time series neural networks, have shown promising results in time series analysis (Paliari, 2021). We hypothesize that the LSTM can potentially replicate the insights once manually derived I.E technical indicators.

[1] Link to full GitHub repository; https://github.com/EmilHaldan/VIX_vertex_prediction

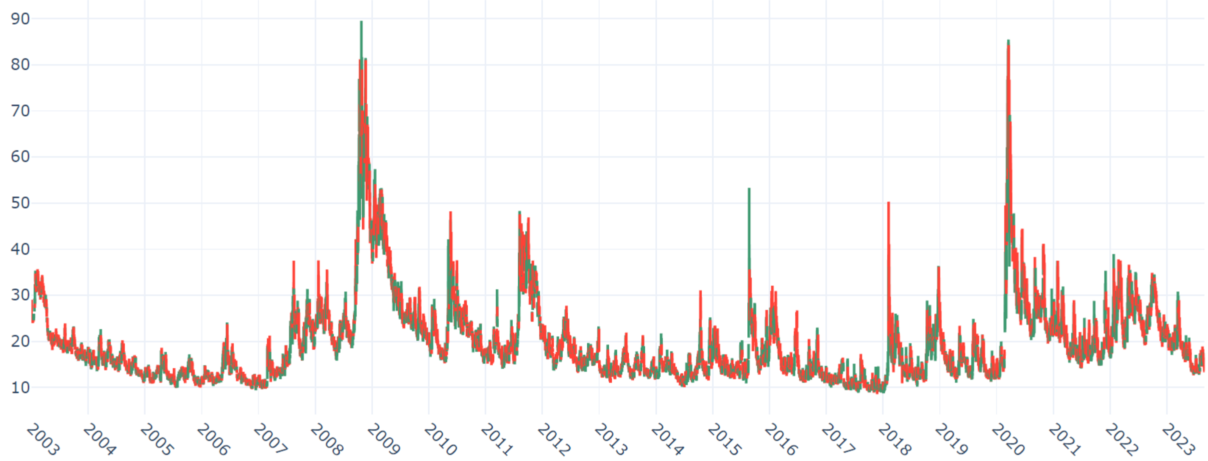


Figure 1: VIX histroical data 2003-2023

This paper will attempt to asses the predictive capabilities of deep neural networks on a niche subset of the financial market, the VIX. The research methodology involves training a LSTM-based neural network with a novel approach to defining the target. We aim to evaluate the effectiveness of technical indicators as features within these models. Additionally, the study will explore the impact in predictive capabilities of a custom loss function, specifically tailored for this task, and to assess its prediction capabilities. The models predictive capabilities will be evaluated using the loss score, as well as testing the model's performance in a simple trading simulation, where evaluation metrics will be based of cumulative returns, and avg. returns.

For this project, we would like to uncover the following research questions:

1. Do technical indicators have any information gain in comparison to only utilizing OHLC indicators, when used with times eries neural networks?

2. Will a custom loss function for the training process be more suitable to predict data points of interest, namely peaks and valley's of the volatility index?

2 Data

2.1 Volatility Index (VIX)

The Volatility Index (VIX), also known as the "fear index" or "fear gauge," is a popular measure of the stock market's expectation of volatility, particularly based on the S&P 500 index options. Managed by the Chicago Board Options Exchange (CBOE), the VIX is calculated and disseminated in real-time by the CBOE.

The Volatility Index (VIX) is calculated based on the prices of S&P 500 index options, which is a reflection of the market's expectation of 30-day forward-looking volatility. The options of the S&P 500 is used as a proxy for the market's forecast of future volatility. Mathematically, the VIX value represents the expected change in the S&P 500 index over the next 30 days. The exact calculation introduce a lot of concepts within finance, and a full understanding of the formula is out of scope for this project. A step by step guide of the calculation can be found on the CBOE website, but is not required in order to understand the findings of the research in this paper. (Chicago Board Options Exchange, 2023).

$$\sigma^2 = \frac{2}{T}\sum_i \frac{\Delta K_i}{K_i^2} e^{RT} Q(K_i) - \frac{1}{T}\left[\frac{F}{K_0} - 1\right]^2$$

$$\text{where} \qquad \mathrm{VIX}_t = \sigma * 100$$

2.2 The Data

To obtain the price data of the VIX, the yahoo finance API wrapper (yfinance)[2] for python was used. With high level code, we were able to obtain Dates, Open, High, Low, and Close values for each day. Yahoo finance provides unlimited historical data if the interval of the candles at of 1 Day, thus, as a university student writing this paper, this option that was chosen. The structure of the OHLC-data will be discussed in detail in section 2.3.

The original dataset spans from 2003-01-02 to 2023-08-31, and includes 2 financial crises, the first being at the start of 2008, and the next in 2020, as seen in figure 1.

Table 1: Summary Statistics of the entire dataset

	Open	High	Low	Close
count	5202	5202	5202	5202
mean	19.48	20.41	18.56	19.35
std	8.75	9.39	8.09	8.70
min	9.01	9.31	8.56	9.14
25%	13.64	14.18	13.12	13.54
50%	17.07	17.86	16.30	16.98
75%	22.56	23.60	21.48	22.40
max	82.69	89.53	72.76	82.69

2.3 OHLC data

In this study, we use Open-High-Low-Close (OHLC) daily candle data of the Volatility Index (VIX) from 2003 to 2023. By using OHLC-data, we can incorporate a summary of information into a 2 dimensional plot.

The x-axis denotes time, and the y-axis denotes the price of an asset or index. Each candle can be treated as 5 data points representing values of interest of an asset in the specific time frame, which in this case is a single day.

For a positive or green candle, the Open price is the bottom solid section of the candle, the close price is the top solid section of the candle, and the high and lows are represented as "wicks" of the candle. For negative or red candles, the open and close prices are swapped such that the bottom is

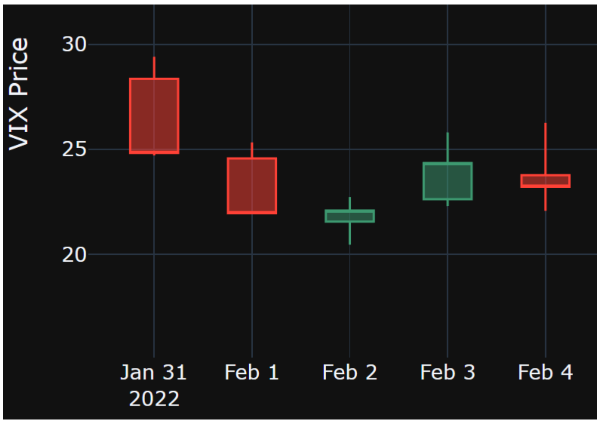


Figure 2: VIX sample price data

now the close price, and the top is now the open price. In theory, the close price of the previous candle should match the open price of the following candle, however there are often gaps in the 1 day candles, as events during the closing hours of the stock market have an effect on the market when it opens again. Furthermore, there are only 5 days of the week where the stoc kmarket is open, which means there is a gap of 2 days between Fridays and Mondays. All of the above is notable in figure 2, where an image of 1 week's worth of 1-Day candle data is visualized.

3 Methodology

The methodology of this research is a multifaceted approach designed to comprehensively analyze and predict the vertices of the Volatility Index (VIX) using a time series neural network. This section provides an overarching view of the various stages and techniques employed in our study, each contributing a unique and essential element to the overall research.

The methodology encompasses a range of steps, beginning with the introduction of technical indicators, which are used as the foundation for the features of the neural networks, and allegedly embeds a nuanced view of market trends (Bottom Street, 2023). We then discuss the partitioning of our data into training, validation, and testing sets, used to ensure robustness and reliability of the predictive models. The subsequent stages involve feature engineering, where we transform and standardize technical indicators to better suit the predictive process.

Central to our approach is the creation of a

[2] yfinance API: https://pypi.org/project/yfinance/

novel target variable, moving beyond conventional forecasting methods, intended to focus on extreme values of the VIX. The core of our methodology is the deployment of a Long Short-Term Memory (LSTM) model, tailored specifically to capture the nuanced dynamics of the VIX. Complementing this, we introduce a custom loss function, crafted with the intention to enhance the model's predictive accuracy during these extreme values.

Finally, the methodology's last phase ends with trading simulations, where we translate our model's predictions into practical trading strategies. This section not only serves as a test bench for the model's effectiveness but also demonstrates its potential application in real-world trading scenarios. Please note that the simulation is not comparable to real world scenarios, as the volume of the market is not considered. Additionally, it is assumed that the VIX is a tradable asset, however, in practice, trading the vix involves buying or selling futures or options of futures (Investopedia, Accessed 2023a, 2023) (SteadyOptions, 2023), which is slightly more complicated process than other assets such as stocks or commodities. Notably, both options trading and futures trading is out of scope for this project.

Each of the above mentioned stages are meticulously detailed in the following subsections, providing a comprehensive road map of the journey from conceptualization to a practical application and evaluation.

3.1 Data Cleaning process

The data cleaning process is a necessary step in any data science pipeline, ensuring the reliability of future analysis. It began with loading in the data and searching for unrealistic extreme values and undefined data points. None anomalies were detected during this process, thanks to the clean data provided by yahoo finance.

Visualizations and a brief exploratory data analysis was also performed in this phase. Visualizing the data helped identifying hypotheses for the underlying patterns, and the overall distribution of the data. Through plots and basic statistical analysis table 1, we gained essential insights into the dataset's characteristics. As the VIX is an index of volatility in the market, it's strays apart from typical stock market data, where a linear regression of the dataset will typically lead to a positive slope, in strong contrast to the VIX which is mean reverting figure 2.

This small step in the analysis ensured that the data used for the feature engineering and in our neural networks was of expected quality.

3.2 Technical Indicators

Technical indicators is one of the typical options to use as input data for financial market analysis (Bottom Street, 2023), offering insights into market trends, momentum, and potential turning points in asset prices. The indicators are based of historical market data, such as price and volume, these indicators serve as the foundation for various trading strategies and market analyses.

However, the use of technical indicators is not without controversy. While many traders and analysts rely on these tools for decision-making, others question their predictive power, arguing that market movements are inherently unpredictable and often influenced by irrational human behaviors and external factors beyond the scope of technical analysis. Although a questionable source, some guy names Raj Sukkersudha on Quora, mentions "a survery conducted by the CFA Institute, approximately 50% of professional investors (including portfolio managers and research analysts) reported using technical analysis to some extent in their investment decision-making process"(Quora User, 2023). On the contrary, Saxo Bank's inspiration disclaimer is: "The price information is historical and cannot be used as a reliable indicator for future returns or market prices" (Saxo Bank, 2023).

Regardless technical indicators are ideal for a simple source of vast features as they are great to compress historical information relative to the current price, and only rely on OHLC-data to create them. This section defines the diverse selection of technical indicators used within this project.

In the following subsections, O_t, H_t, L_t, and C_t, will denote the Open, High, Low, and Close values at time t.

3.2.1 AROON

The AROON is an indicator which states the amount time there has been since the previous highest or lowest price in a sequence of the past n candles, as stated on (Investopedia, 2023a)

$$\text{Aroon Up}_{n,t} = \frac{n - \text{Days since high}}{n} \cdot 100$$

$$\text{Aroon Down}_{n,t} = \frac{n - \text{Days since low}}{n} \cdot 100$$

If Aroon_Up = 100, the current price is the highest it has been looking n days back. Likewise, if Aroon_Down = 100, the current price is the lowest it has been looking n days back.

Figure 3: AROON_24 Up (Green) and Down (Red) 2022.

3.2.2 Bollinger Bands & Simple Moving Average

Bollinger Bands consist of a middle band being an n-period simple moving average (SMA), and the upper and lower bands at k times an n-period standard deviation of the close price added or subtracted to the SMA. (Investopedia, 2023b)

$$SMA_{n,t} = \sum_{t-n}^{t} \frac{C_t}{n}$$

$$\text{Upper Bollinger Band}_{n,t} = SMAn,t + (k \cdot \sigma_{n,t})$$

$$\text{Lower Bollinger Band}_{n,t} = SMAn,t - (k \cdot \sigma_{n,t})$$

$$\sigma_{n,t} = \sqrt{\frac{\sum_{i=t-n}^{t} (x_i - \bar{X}_{n,t})^2}{n}}$$

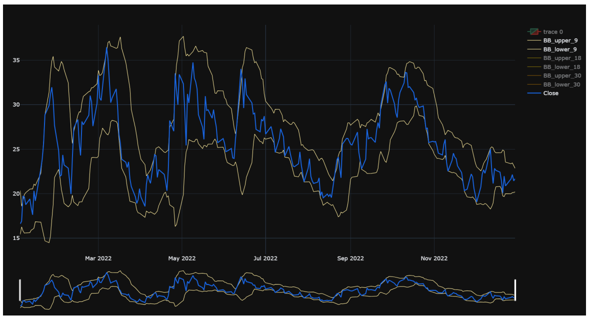


Figure 4: BB, K=2, N=9, sample 2022.

3.2.3 EMA

The Exponential Moving Average (EMA) is a variant of a moving average, which weighs the current price higher, and does infact not have any exponential components in the formula. The visualization in figure 5 illustrates EMA_n for different values of n. (Investopedia, 2023c)

$$\text{EMA}_{n,t} = (C_t \times \alpha) + (\text{EMA}_{n,t-1} \times (1 - \alpha)) \quad (1)$$

where:

- $\text{EMA}_{n,t-1}$ is the EMA value for the previous day (day $t-1$).
- α is the smoothing factor, typically given by:

$$\alpha = \frac{2}{n+1} \quad (2)$$

where n is the number of candles in the span of the EMA.

Figure 5: EMA 2022.

3.2.4 MACD

The Moving Average Convergence Divergence (MACD) is derived from two Exponential Moving Averages (EMAs):

The MACD (dark blue in figure 6) itself is the difference between EMA_{12} *and* EMA_{26}:

$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26} \tag{3}$$

The Signal line (orange figure 6), which is a 9-period EMA of the MACD, is given by:

$$\text{Signal} = (MACD_9 \times \alpha) + (\text{EMA}_{t-1} \times (1 - \alpha)) \tag{4}$$

The Histogram is the difference between the MACD and the Signal line, where legend has is that when these lines intersect, it's wise to place a call or a put trade. On the example of 2022's data it appears to be valid for long term investments.

In the code, the n is used as a scaling factor for the previous three mentioned values (9,12 and 26).

Figure 6: MACD 2022

3.2.5 ROC

The Rate of Change (ROC) calculates the percentage difference between the current price and the price n-candles ago. Presumably, as it's typical in finance to convert values to percentages, the formula is scaled by 100. (Investopedia, 2023a)

$$\text{ROC}_{n,t} = \frac{\text{C}_t - \text{C}_{t-n}}{\text{C}_{t-n}} \cdot 100$$

3.2.6 RSI

The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. The RSI oscillates between 0 and 100, where these maximum are seen if the ROC_1 is consecutively negative/positive throughout the entire period. ()

1. Calculate the gains and losses using close prices. A gain for any given time t is calculated when $C_t - C_{t-1} > 0$ Conversely, a loss occurs when $C_t - C_{t-1} < 0$.

   The Average Gain over n periods is then:

   $$\text{Average Gain}_{n,t} = \frac{1}{n} \sum_{t-n}^{t} \max(C_t - C_{t-1}, 0)$$

   $$\text{Average Loss}_{n,t} = \frac{1}{n} \sum_{t-n}^{t} \min(C_{t-1} - C_t, 0)$$

   where $\max(C_t - C_{t-1}, 0)$ ensures we only consider positive differences for gains.

2. Calculate the RSI for the first n steps:

   $$\text{RSI}_{n,t} = 100 - \frac{100}{1 + \frac{\text{Average Gain}_{n,t}}{\text{Average Loss}_{n,t}}}$$

3. Calculate the RSI for the next steps:

   $$\text{RSI}_{n,t} = 100 - \frac{100}{1 + \frac{\text{Avg Gain}_{n-1,t-1} + \text{Cur Gain}_t}{\text{Avg Loss}_{n-1,t-1} + \text{Cur Loss}_t}}$$

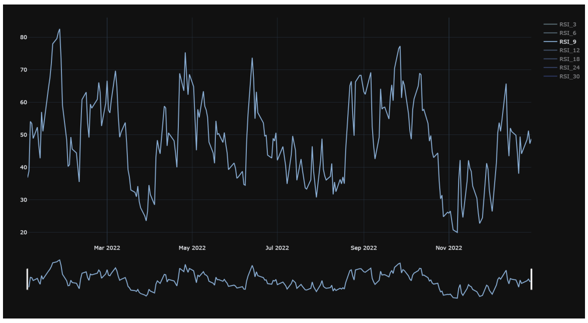


Figure 7: RSI_9 2022

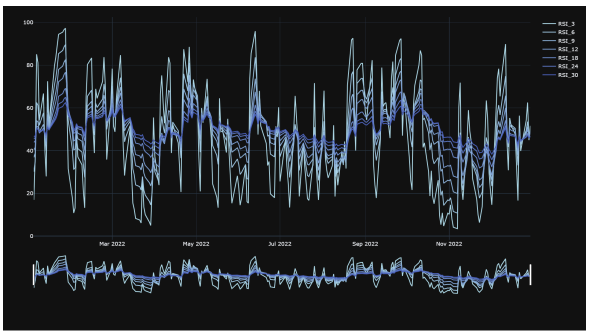


Figure 8: RSI_{all} 2022

3.2.7 Max_High and Min_Low

The Max_High and Min_Low are the only indicators which are not strictly technical indicators, but rather caches of maximums and minimums over a period of n rolling back. These features are used to derive the target, which will be discussed in detail in section 3.6.

$$\text{Max\_High}_{n,t} = \max_{t-n\leq k\leq t}(\text{High}_k)$$

$$\text{Min\_Low}_{n,t} = \min_{t-n\leq k\leq t}(\text{Low}_k)$$

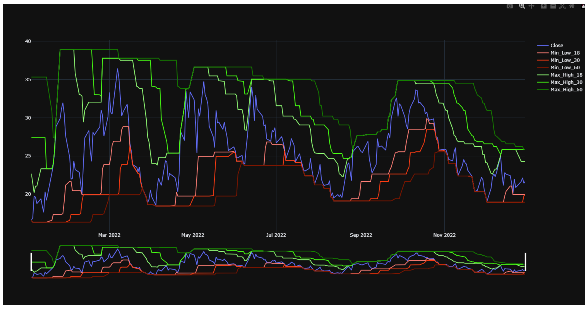


Figure 9: Max_High and Min_Low given some values of n 2022

3.2.8 High_Low_Diff, and High_Low_Mean

The High_Low_Diff is simply the difference of the two previous mentioned values at any given time t. Likewise, the The High_Low_Mean is the mean of the two previous mentioned values at any given time t.

$$\text{High\_Low\_Diff}_{n,t} = \text{Max\_High}_{n,t} - \text{Min\_Low}_{n,t}$$

$$\text{High\_Low\_Mean}_{n,t} = \frac{\text{Max\_High}_{n,t} + \text{Min\_Low}_{n,t}}{2}$$

3.3 Train, Validation, Test split

In line with standard machine learning practices, our dataset is partitioned into three distinct sets: training, validation, and testing. This division is critical for evaluating the performance and generalizability of our LSTM model in predicting the vertices of the VIX. As the datasat is used for timeseries analysis, and the LSTM includes windows of multiple rows table 3, the splits had to be done consecutively, in order to avoid information leakage.

The **Training set** encompasses data from January 2, 2003, to December 31, 2017. This amounts to 3776 days of 1 day candle data points. This period includes significant market events, most notably the 2008 financial crisis. Including such a distinct period in the training data exposes the model to extreme market conditions, hopefully enabling it to learn and adapt to future scenarios of extremely high VIX prices.

The **Validation set** spans from January 2, 2018, to December 31, 2019, which amounts to 503 days of 1 day candle data points, The absence of a market crash in the validation data is intentional, as it allows for the assessment of the model's performance in relatively stable or 'normal' market conditions. This set is pivotal for fine-tuning the model parameters and making necessary adjustments before the final evaluation.

Finally, the **Test set** covers the period from January 2, 2020, to August 31, 2023. This set includes the market dynamics affected by the 2020 market crash, triggered by the COVID-19 pandemic. The test period spans 923 days worth of data points. Testing the model on this data provides insights into its ability to handle recent and unprecedented market conditions, offering a robust evaluation of its predictive capabilities in the face of significant market disruptions.

The rationale for this specific partitioning of the data is to ensure that the model is trained and evaluated across a comprehensive range of market conditions, in contrast to follow the common practice of a 70/10/20 split for the train, validation, and test set.

3.4 Feature engineering

In the feature engineering phase, the primary objective was to transform the raw financial data into a format that is more suitable as features for time series neural networks. Firstly, it involved altering the features to be relative to the closing price, as it was hypothesized that the relative value of a technical indicator to the closing price is more interesting than the numerical value of the indicator itself. E.G Bollinger Bands in section 3.2.2 and the moving averages in section 3.2.3. This relative scaling attempts to imitate what analysts would base their decisions of when viewing charts of technical indicators.

The RSI and AROON have fixed ranges between 0 and 100, and the MACD has a soft minimum/maximum. Because of this they not undergo any transformation prior to their standardization in section 3.5. The list of tranformations can be seen in table 2.

Each technical indicator, has a parameter n, which denotes the window range of previous candle data points included in their calculation. In an attempt to include sufficient information, each indicator has a range of values for n. This leads to a total of 136 features being used (including the Open, High, Low and Close values) as input for the time series models. The list of values for n can be seen in table 2.

Table 2: Technical Indicators and their transformations

Technical Indicator	Transformation	Values for n
OHLC	None	N/A
BB	$(x_t - c_t)/c_t$	{3,6,9,12,18,24,30}
EMA	$(x_t - c_t)/c_t$	{3,6,9,12,18,24,30 ,36,42,48,54,60}
Max_High	$(x_t - c_t)/c_t$	{3,6,9,12,18,24,30} ,36,42,48,54,60}
Min_Low	$(x_t - c_t)/c_t$	{3,6,9,12,18,24,30} ,36,42,48,54,60}
High_Low_Mean	$(x_t - c_t)/c_t$	{3,6,9,12,18,24,30} ,36,42,48,54,60}
High_Low_Diff	$(x_t - c_t)/c_t$	{3,6,9,12,18,24,30} ,36,42,48,54,60}
AROON	None	{3,6,9,12,18,24,30}
RSI	None	{3,6,9,12,18,24,30}
MACD	None	{3,6,9,12,18,24,30}

3.5 Standardization

Standardization of the data to have a mean of 0, and a standard deviation of 1 was implemented. This normalization process is vital for facilitating better convergence during the models training process. To prevent information leakage, the mean and standard deviation used for the normalization were calculated based on the subset of data available before a given timestamp t. Specifically:

$$\bar{X}_t = \frac{1}{t}\sum_{i=1}^{t} x_i$$

$$\sigma_t = \sqrt{\frac{\sum_{i=1}^{t}(x_i - \bar{X}_t)^2}{t}}$$

For the validation and test sets, we employ the last known mean and standard deviation from the training set. This approach ensures consistency across datasets while avoiding the introduction of future information into the training process.

3.6 Target Creation

Objective and Rationale: A critical aspect of the research is the formulation of a target variable for the task of the predictive model. This section elaborates on the rationale and methodology behind the target creation process.

Traditionally, financial market predictions often revolve around forecasting the returns for the next trading period, such as the next day or the next candle in a candle chart. However, such an approach, while common, has its limitations in capturing broader market movements, especially when it comes to predicting significant turning points like local highs and lows.

The following sections will dive into the creation of the two targets used for this project. In this design, the first target is the ”Low Target”, which ranges from 0 to 1, where 1 indicates that the current price is very close to a local price floor, and that it would be a good time to buy a call option (Investopedia, Accessed 2023a), or ”buy”. The second target is the ”High Target” which also ranged from 0 to 1, where 1 indicates that the current price is close to a local price ceiling, hence, it would be a good time to buy a put option, or ”sell”. The targets were created such that trading decisions at target values greater that 0.99, would lead to the best possible returns.

One of the key objective’s in this research is to shift the focus from short-term price movements to predicting these crucial market points - the local highs and lows. This approach attempts to enable predictions to be converted directly to simple decision-making.

3.6.1 Defining High and Low targets

To algorithmically identify local highs and lows, we employed a window of size w. Using the window’s midpoint at time t, the algorithm looks back to the previous candles from $t - \frac{w}{2}$ and future candles up to $t + \frac{w}{2}$, to determine the local low ($\text{Min\_Low}_{n,t,w}$) and local high ($\text{Max\_High}_{n,t,w}$) points.

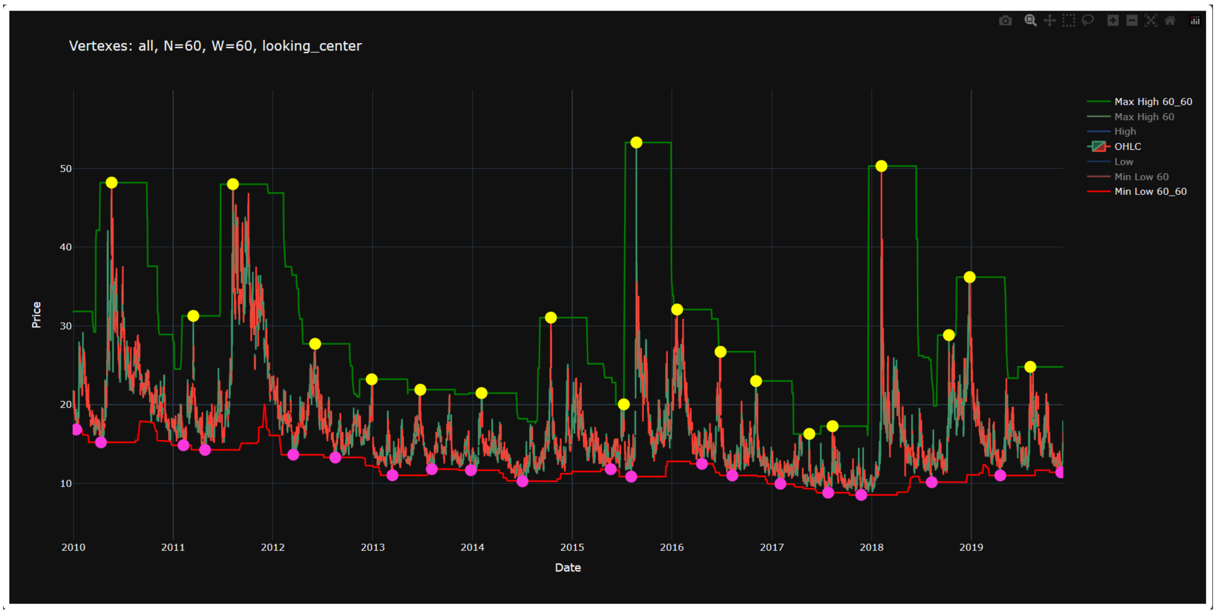


Figure 10: Target creation step 1.

$$\text{Max\_High}_{n,t,w} = \max_{i=t-n}^{t+n} (\text{Max\_High}_{n,t})$$

$$\text{Min\_High}_{n,t,w} = \min_{i=t-n}^{t+n} (\text{Min\_High}_{n,t})$$

$\text{Max\_High}_{n,t}$ and $\text{Min\_High}_{n,t}$ is defined in detail in section 3.2.7.

The ($\text{Min\_Low}_{n,t,w}$) represents the local floor, while the ($\text{Max\_High}_{n,t,w}$) represents the local ceiling. The ceiling and floor mentioned before can been seen in figure 10.

To define the points of interest, the date and value of a vertex (both High and Low), are stored when the High or Low of a candle is equal to the ceiling or floor. These points are visualized in figure 10 by the yellow and purple dots. After creating some samples of different values for n and w, some visual inspection deemed that $n = 60$ and $w = 60$ appeared to capture the local peaks and valley's to a degree that was acceptable. Theses previous mentioned ceilings and floors, establish the basis for creating the High and Low targets for the model.

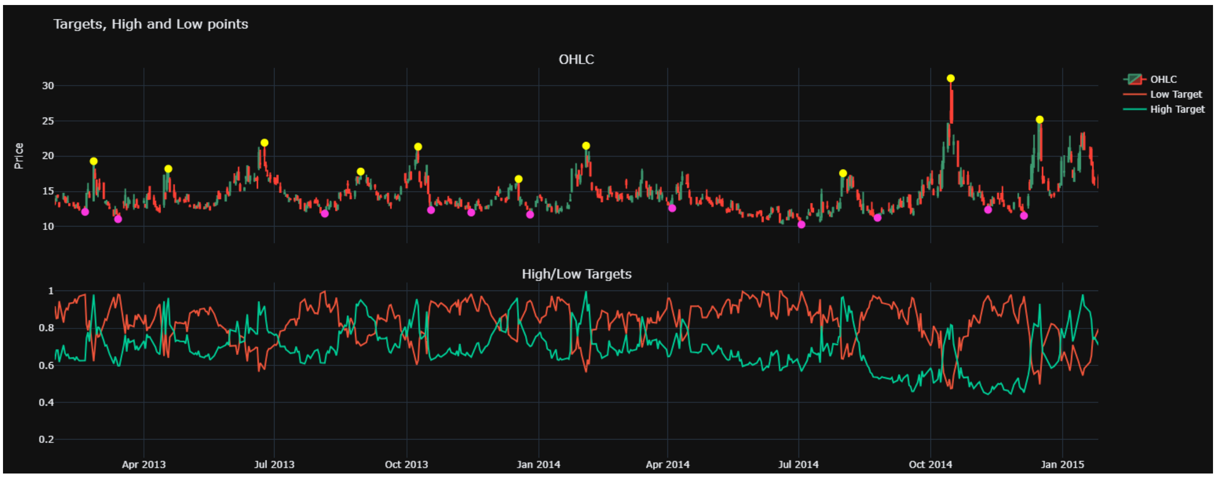


Figure 11: Target creation step 2.

To define the High Target...

Let C_t denote the closing price at time t,

Let P_t denote the previous high point at time t,

Let $\Delta T(P_t, C_t)$ denote the distance of the previous high P_t to the current time t as a positive integer

Let F_t denote the future high point at time t,

Let $\Delta T(C_t, F_t)$ denote the distance of the future high P_t to the current time t as a positive integer

$$\text{High Target}_t = \frac{C_t}{\left(\frac{F_t \cdot \Delta T(P_t, C_t) + P_t \cdot \Delta T(C_t, F_t)}{\Delta T(P_t, F_t)}\right)}$$

Which simplifies to:

$$\text{High Target}_t = \frac{C_t \cdot \Delta T(P_t, F_t)}{F_t \cdot \Delta T(P_t, C_t) + P_t \cdot \Delta T(C_t, F_t))}$$

The formula for High Target_t has a theoretical range of 0 to 1, as illustrated by the green line in figure 11

To define the Low Target...

Let C_t denote the closing price at time t,

Let P_t denote the previous low point at time t,

Let $\Delta T(P_t, C_t)$ denote the distance of the previous low P_t to the current time t as a positive integer

Let F_t denote the future low point at time t,

Let $\Delta T(C_t, F_t)$ denote the distance of the future low P_t to the current time t as a positive integer

$$\text{Low Target}_t = \frac{F_t \cdot \Delta T(P_t, C_t) + P_t \cdot \Delta T(C_t, F_t))}{C_t \cdot \Delta T(P_t, F_t)}$$

Likewise, the formula for Low Target_t has a theoretical range of 0 to 1, as illustrated by the red line in figure 11

Concerns regarding information leakage are definitely needed in time series forecasting, however, the nature of forecasting inherently requires the target to be placed in the future, arguably making the approach consistent with standard practices in financial forecasting seen in (Zou, 2022).

This alternative approach to target creation allows the model to focus on predicting significant market peaks and valley's, offering potential improvements on forecasting a mean-reverting asset or index, such as VIX. By defining targets that capture the extremes of market extremes, the model is hopefully better positioned to provide actionable insights.

3.7 Time series model (LSTM)

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, have gained significant traction in the field of time series analysis due to their ability to capture temporal dependencies and patterns in sequential data. LSTMs are particularly adept at handling the complexities inherent in financial time series data, such as non-linearity and high volatility, with documented examples with their suitability for forecasting in financial markets (Paliari, 2021), (Cao et al., 2019), (Siami-Namini and Namin, 2018).

The choice of using an LSTM is motivated by its demonstrated capability to effectively model and forecast time-dependent data. In the introduction it is hypothesized that the LSTM, by it's inherent design, has the capability to create its own relevant linear combinations from the OHLC data figure 12. If true, this suggests that the process of manually creating features could be redundant.

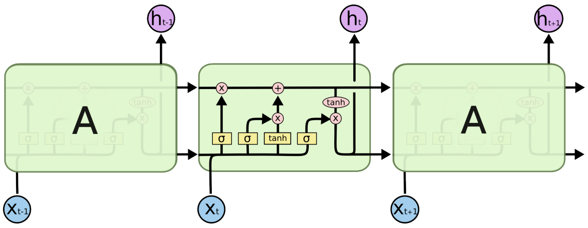


Figure 12: LSTM cell from Colah's Blog (Olah, 2015)

It's is important to note that the primary focus of this research is not on hyperparameter tuning of the LSTM model, nor a deeper assessment of it's intricate functionality in time series predictions. While hyperparameter optimization is a crucial aspect in many machine learning applications, our research takes a different approach, as the increase of moving parts for the scope of this project would make it complicated to derive conclusions from.

Instead, this project concentrates it's exploration by comparing the impacts of different features and loss functions on the model's performance. More specifically, we compare the effectiveness of using basic Open-High-Low-Close (OHLC) data versus a more comprehensive set of features that include various technical indicators. This comparison aims to uncover whether the inclusion of these additional features enhances the model's ability to forecast market movements.

Furthermore, this paper also investigates the performance differences between models using Mean Squared Error (MSE) as a loss function and those employing a custom-designed loss function seen in section 3.8. The custom loss function is designed to the specific objectives of this project, potentially offering better results when attempting to capture the nuances of the VIX's movements.

The specific experimental setup includes training 100 models of each configuration of loss function and feature data;

1. MSE, using OHLC data
2. MSE, using All Features
3. Custom Loss, using OHLC data
4. Custom Loss, using All Features

As mentioned, the focus is diverged from hyper parameter tuning, hence, all of the LSTM models share the same hyper parameters. These parameters were chosen based on computational availability, and best estimation, creating a large space for improvement. The specific hyper parameters can be found in table 3.

Table 3: Hyperparameters of the LSTM Model

LSTM Hyper Parameters	**Value**	**Info**
Optimizer	Adamax	
Drop out rate	0.1	* included
Learning rate	10^{-5}	
Max Epochs	500	
Patience	15	
Batch Size	32	
Window Size	20	
Dense Layer act-Func	Elu	
LSTM Layer Type	**Output Shape**	**Params**
LSTM Input Layer	(20, 256)	402,432
Batch Normalization	(20, 256)	1,024
LSTM Layer 2*	(256)	525,312
Dense Layer*	(256)	65,792
Output Nodes	(2)	514
Total Trainable Params		994,562
Total Non-Trainable Params		512

3.8 Custom Loss Function

In our research, we introduce a custom loss function that stands in contrast to the commonly used Mean Squared Error (MSE). The custom loss function is designed to particularly emphasize errors in predictions where either the targets or predictions are high, which are critical for the model's accuracy in forecasting significant market movements. The formula for this custom loss function is as follows:

$$\text{MSE Loss} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{5}$$

$$\text{Custom Loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y}_i)^2 \cdot e^{Y_i} \cdot e^{\hat{Y}_i}}{2} \tag{6}$$

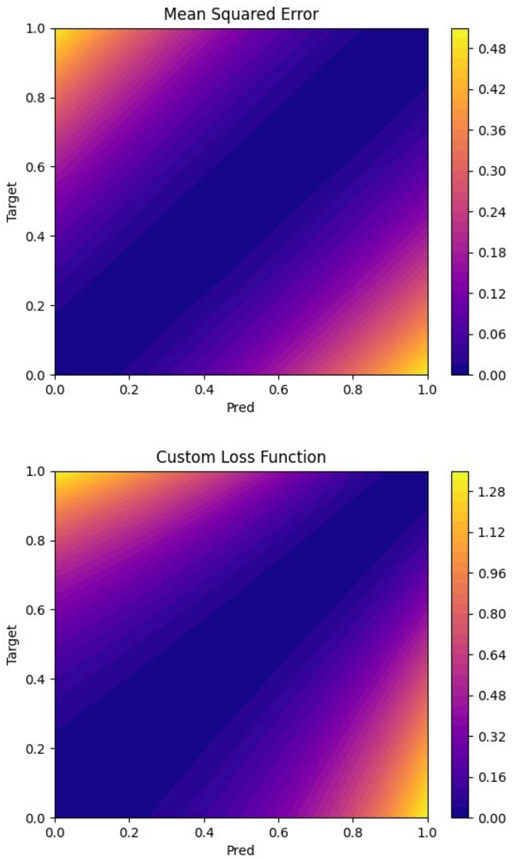


Figure 13: MSE and custom loss function plots

The intention behind the expression is to increase the priority of the model's accuracy in instances where the targets and predictions have higher values, which, in the context of the project, represent crucial points for prediction, I.E peaks and valleys. This is contrast to MSE, which treats all errors uniformly regardless of the target value, our custom loss function introduces a dynamic and exponential scaling factor, as seen in figure 13. The loss function is divided by 2 for previous visualization aspirations, however in hindsight this division doesn't carry any meaningful functionality and should be discarded.

Given that the target range theoretically spans from 0 to 1, the custom loss function ensures that as the target and prediction values increase, the model is penalized more heavily for inaccuracies. Combinations of loss values are visualized in a heat map in figure 11. This functionality is hypothesized to be crucial as the primary objective of the function is to accurately predict high targets.

The effectiveness of this custom loss function will be evaluated in comparison to the standard MSE in subsequent sections, particularly in terms of its impact on the predictive accuracy of our LSTM model in capturing the critical peaks and valleys of the VIX.

3.9 Trading Simulations

Trading simulations form a critical part of the evaluation process, providing practical insights into the applicability of our LSTM model predictions for the VIX. These simulations were specifically conducted on the **validation data** set and were structured around a set of predefined rules and thresholds. The following section not only tests the model's predictive accuracy but also explores it's potential in guiding trading decisions on unseen data.

The buying actions in the simulations inlcude both ”call” and ”puts”, where call positions profit when the VIX increases, and put positions profit when the VIX decreases. They are governed by three key thresholds: High_threshold for high predictions, Low_threshold for low predictions, and the Diff_threshold for the difference between high and low predictions. The decision to initiate trades was based on the following criteria:

A **call** position is initiated when all of the following conditions are met:

1. The prediction difference ”pred_diff” is greater than the Diff_threshold,
2. The previous high prediction ”previous_high_pred” is greater than the current high prediction ”current_high_pred”,
3. The high prediction ”high_pred” surpasses the High_threshold.

A **put** position is initiated when all of the following conditions are met:

1. The prediction difference ”pred_diff” is less than the negative Diff_threshold,
2. The previous low prediction ”previous_low_pred” is greater than the current low prediction ”current_low_pred”,
3. The low prediction ”low_pred” exceeds the Low_threshold.

The specific implementation can be viewed in the function `trading_simulation()` in the script `s3_2_0_simulate_trades.py`[3]

Each trade, once initiated, could remain open for a maximum of 50 days. Positions were closed if the trade duration exceeded 50 days or if the profit exceeds 50% based on the closing price and the trade type. It's important to note that only one trade could be active at any given time to mimic a realistic trading scenario. Although a take profit mechanism (Investopedia, Accessed 2023c) is included, a stop loss mechanism (Investopedia, Accessed 2023b) is not included. The stop-loss mechanism was intentionally excluded as it would further increase the number of moving components, potentially obfuscating the results. The take profit is included to emulate the intended use case, namely to use this implementation for buying call and put options (Investopedia, Accessed 2023a) and executing said options at the desired point in time.

To evaluate the performance of each trading simulation, there are a handful of results which are of interest, namely the **average returns**, **trades placed** and the **cumulative returns** (NorthstarRisk, 2023). Ideally, an investor would like to maximise their cumulative returns, however, high returns often follow with high risk. To emulate the slightest aspect of risk management, cumulative returns are based on investing 20% of the agents total capital available pr. time, to avoid loosing all liquidity in the event of a loss. This resembles a possible portion of a portfolio one would invest with. Additionally, investing with 20% of the total available cash means that a big loss in the beginning doesn't alter future performance as heavily.

For the returns, we chose to use the definition by (Investopedia, 2023b), which will be used for all values regarding returns in the section 4:

$$\text{Returns (\%)} = \frac{100 \cdot (\text{Current Value} - \text{Initial Value})}{\text{Initial Value}}$$

$$\text{Returns} = \frac{(\text{Current Value} - \text{Initial Value})}{\text{Initial Value}}$$

Let the returns of the investment i be denoted as R_i, and the total amount of investments be denoted n, then...

$$\text{Cumulatile Returns} = \left(\prod_{i=1}^{n} 1 + R_i\right) - 1$$

The simulations were exhaustively conducted for all permutations of threshold values within the specified ranges:

- Diff_thresholds: $\{0, 0.05, 0.1, ..., 0.45\}$
- High_thresholds: $\{0.3, 0.35, ..., 0.95\}$
- Low_thresholds: $\{0.3, 0.35, ..., 0.95\}$

In total, 15,680 simulations were conducted, with each combination of loss functions and input data configurations undergoing 3,920 simulations (1960 for the validation data, and 1960 for the test data). This exhaustive approach ensures a comprehensive analysis of the model's performance across a wide spectrum of scenarios and thresholds, providing a robust assessment of its predictive capabilities in a trading context. Each simulations parameters and the corresponding results were stored in an SQLite database for easy of access.

For a reasonable comparison, we will deploy a baseline simulation, which is based on the simple rule-set "Buy low and sell high". As the volatility index is mean reverting as seen in figure 10 and, this strategy will always place a call position at a fixed value, and a put position at a fixed value.

To define the baseline thresholds, we will be basing them of the the 75th and 25th percentile, of the high and low values in the OHLC training dataset.

$$\text{Baseline\_Low} = 12.760$$

$$\text{Baseline\_High} = 21.4625$$

The baseline simulation will follow the same rules as previously stated, however, instead of using a time series neural network, we will simply initiate a call position if the price is below Baseline_Low, and initiate a put position if the price is above Baseline_High.

[3] Link to full GitHub repository; https://github.com/EmilHaldan/VIX_vertex_prediction

Table 4: Loss scores mean and standard deviation

features used	loss type	mean val loss	mean test loss	epochs trained
all_features	MSE	0.0128 ±0.0013	0.0166 ±0.0021	162.1 ±43.1
OHLC	MSE	0.0110 ±0.0014	0.0149 ±0.0025	44.7 ±31.5
all_features	Custom	0.0479 ±0.0061	0.0634 ±0.0078	163.4 ±46.4
OHLC	Custom	0.0427 ±0.0033	0.0523 ±0.0059	41.8 ±28.4

4 Results

This section presents the findings of our analysis, focusing on the performance in trading simulations of the Long Short-Term Memory (LSTM) model under various configurations in predicting the peaks and valleys of the Volatility Index (VIX). The results are derived from an exhaustive series of simulations and analyses, utilizing the validation dataset for tuning the thresholds of the trading simulation agent. Simulation on the test set were based on optimal thresholds found from the simulations of the validation set. The results are conveyed through a series of data visualizations, including tabular data, violin/box plots, and naturally line charts.

4.1 Loss scores

We start with a tabular presentation of loss scores, with different combinations of loss functions (Mean Squared Error vs. Custom Loss) and feature sets (OHLC vs. All Features). In general, the models using OHLC data as features tend to have a better loss score, and also tend to train for a much shorter amount of epochs than the models using all features table 4. Notably, the standard deviation of the loss' indicates that most models must have converged to similar weights. The best performing models (in terms of loss scores) table 5 tended to train for a longer time than the mean training training time. In terms of loss scores, the models only using OHLC data as features performed better than the models using all features.

In figure 14 the targets and predictions of best performing models from table 5 are visualized on the test set. If we focus our attention to the peaks of the high and low targets, we are interested in seeing if the predictions are similar. Noticably, the models using OHLC data (LSTM_46 and LSTM_17) fluctuate much less in their predictions in contrast to the models using feature engineered data (LSTM_70 and LSTM_78). Interestingly, when the Covid-19 crisis sparked economic turmoil in 2020, the low predictions appear to be at their lowest point (indicating it's a bad time to buy the VIX according to the set of rules defined in section 3.9).

4.2 Trading Simulations

The following section will use the four models which had the lowest validation loss values shown in table 5. This is due to an interest in keeping the test set isolated until final examination. The following sections will illustrate the performance of the LSTM's in trading simulations, using an exhaustive search for the optimal thresholds for the trading simulation. Once found, these parameters will be used to simulate trades on the test set.

4.2.1 Distribution of returns

The results of the trading simulations using different thresholds are visualized with two violin plots with integrated boxplots, to visualize the distribution representing the cumulative and average returns for each configuration figure 16. These distributions, show interesting results, as the maximum cumulative returns are of the OHLC, MSE combi-

Table 5: Best model's loss scores

features used	loss type	val loss	test loss	epochs trained	model name
all_features	MSE	0.0106	0.0151	224	LSTM_70
OHLC	MSE	0.0102	0.0129	120	LSTM_46
all_features	Custom	0.034	0.0716	112	LSTM_78
OHLC	Custom	0.0362	0.0465	113	LSTM_17

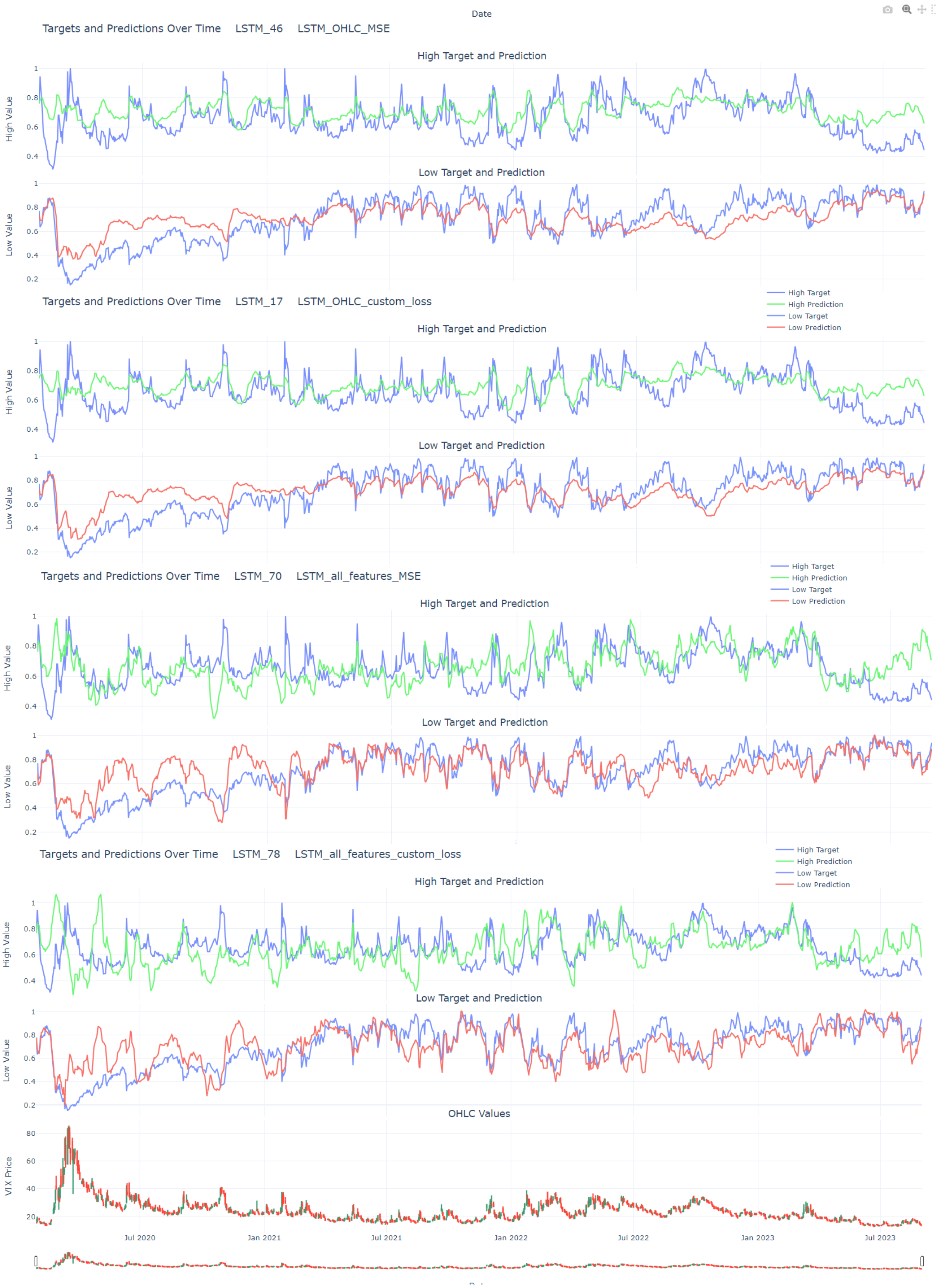


Figure 14: Predictions (red and green) and Targets (blue) on the test set of the models from table 5

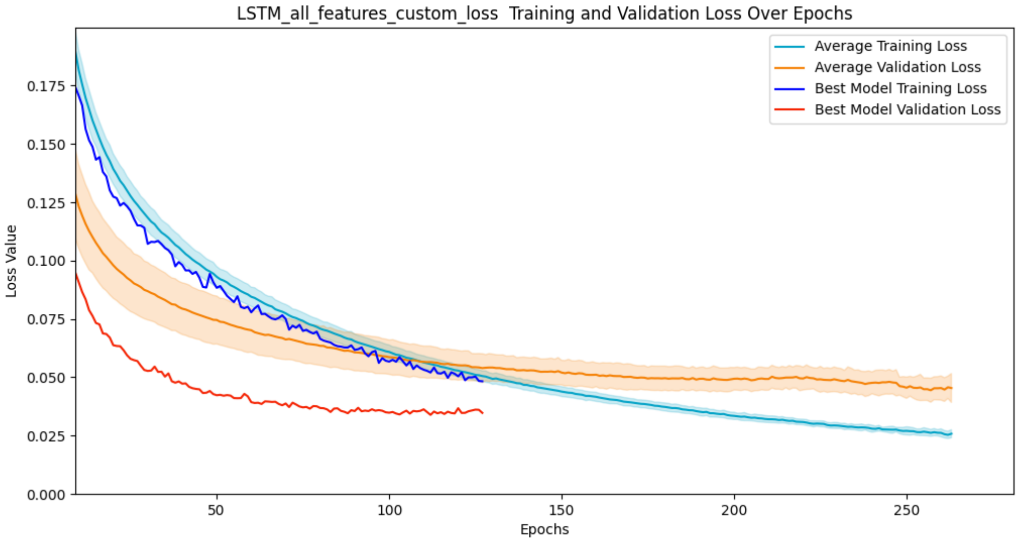


Figure 15: Training and Validation loss of the models using All Features and the Custom Loss function

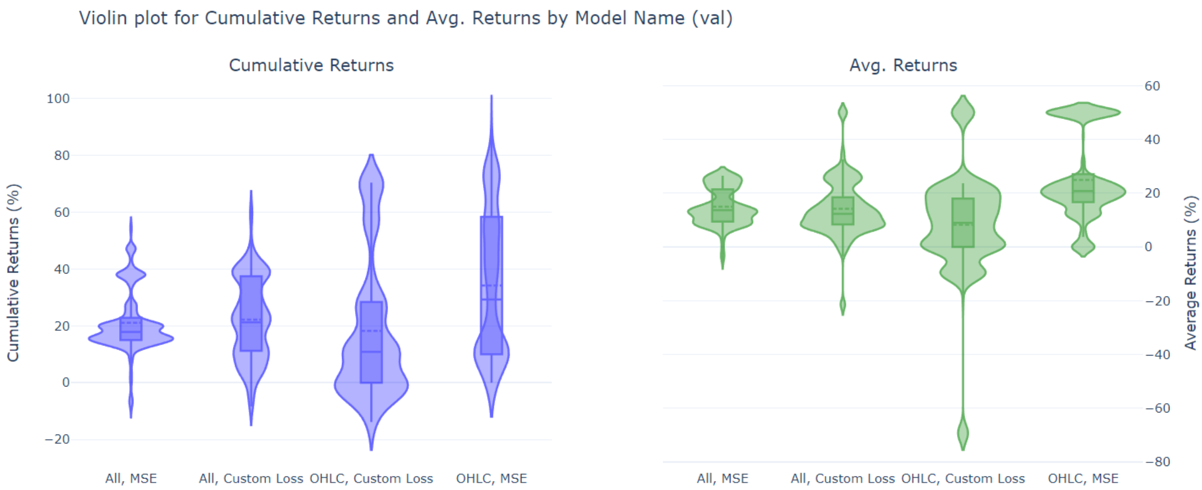


Figure 16: Simulation results Violin and Boxplot (Validation Set)

Table 6: Best model's trading simulations, based of validation **Average Returns**

Metric	OHLC, MSE	All, MSE	OHLC, Custom Loss	All, Custom Loss	Baseline
LSTM number	46	70	17	78	N/A
High Threshold	0.8	0.95	0.7	0.95	1
Low Threshold	0.95	0.95	0.95	0.85	1
Diff Threshold	0.45	0.45	0.4	0.45	0
Validation Results					
Cumulative	10 %	16.03 %	10%	10 %	117.47 %
Avg. Returns	**50 %**	**26.49 %**	**50%**	**50%**	**33.94 %**
Trades Placed	1	3	1	1	12
Test Results					
Cumulative	0.0 %	0.0%	3.94 %	19.70 %	39.15 %
Avg. Returns	**0.0%**	**0.0%**	**19.71%**	**31.07 %**	**10.27 %**
Trades Placed	0	0	1	3	17

Table 7: Best model's trading simulations, based of validation **Cumulative Returns**

Metric	OHLC, MSE	All, MSE	OHLC, Custom Loss	All, Custom Loss	Baseline
LSTM number	46	70	17	78	N/A
High Threshold	0.75	0.55	0.8	0.6	1
Low Threshold	0.8	0.85	0.7	0.9	1
Diff Threshold	0	0	0.1	0.25	0
Validation Results					
Cumulative	**88.91 %**	**55.43 %**	**70.35 %**	**60.83 %**	**117.47 %**
Avg. Returns	22.72%	19.55 %	22.04 %	25.24%	33.94 %
Trades Placed	15	12	13	10	12
Test Results					
Cumulative	**-9.94 %**	**15.10%**	**52.20 %**	**78.85 %**	**39.15 %**
Avg. Returns	-0.89 %	5.26%	14.64 %	23.41 %	10.27 %
Trades Placed	16	18	15	13	17

nation, while the 3rd quartile of the (All, Custom Loss) combination is significantly larger. Taking a look at the Avg. Returns plot, almost the complete opposite.

4.2.2 Best simulation performances for each of the model combinations

To enhance our understanding of the simulation outcomes, the best performing models from figure 16 were meticulously analyzed. Employing these top-performing thresholds from the validation set, we conducted further simulations on the test set. This approach was borrowed from the methodology used in machine learning, where the validation is used to identify the most effective training parameters. By applying these thresholds to the test set, as detailed in table 7 and table 6, we achieved some interesting results.

Using this approach we can see that the thresholds determined on the validation data are naturally "over fitted". The results on the test set, which spans over two and a half years, were notably less impressive. Interestingly, the adoption of conservative parameters that had shown the best average returns on the validation set resulted in either passive or underwhelming performance in the actual trading simulations. The set of thresholds when using the model which was configured with all features and the custom loss function performed much better than it's counter part using OHLC and MSE on the test set when using the best threshold for cumulative returns.

The predictions of the model using the custom loss function, and all features can be seen in figure 17, where the top chart visualizes the test data set, along with the locations of the actions. A green dotted line indicates a profitable trade, while a red dotted line indicates a loss. The red and green silos in the chart indicate whether the position was a call or put. The chart below the price data is an illustration of unrealized cumulative returns on a day to day basis. The areas where the chart is flat corresponds to the agent not having an active trade.

5 Discussion

The results of our study present a nuanced picture of LSTM model performance in financial time series forecasting, particularly in predicting the peaks and valleys of the VIX. This discussion delves into the implications of our findings, considering various factors such as the chosen data, the features, modeling choices, risk profiles, and more.

5.1 Time Frame of Data (1D vs. 1H Candles)

To start with, we will highlight the use of 1-day candles as opposed to 1-hour candles in our analysis and which complications it may introduce. A finer granularity of the data could potentially have impacted the model's predictive performance and trading decisions. Additionally, the use of finer granularity also significantly increases the amount of available training data.

5.2 Risk Profile and Returns

The interpretation of our results is significantly influenced by the investor's risk profile. Whether the goal is to maximize compounded returns or to prioritize average returns per trade, the choice alters the perception of the model's success. This variability underscores the necessity for investors to align model usage with their individual risk tolerance and investment strategy. This project did not investigate any methods or aspect of risk management,

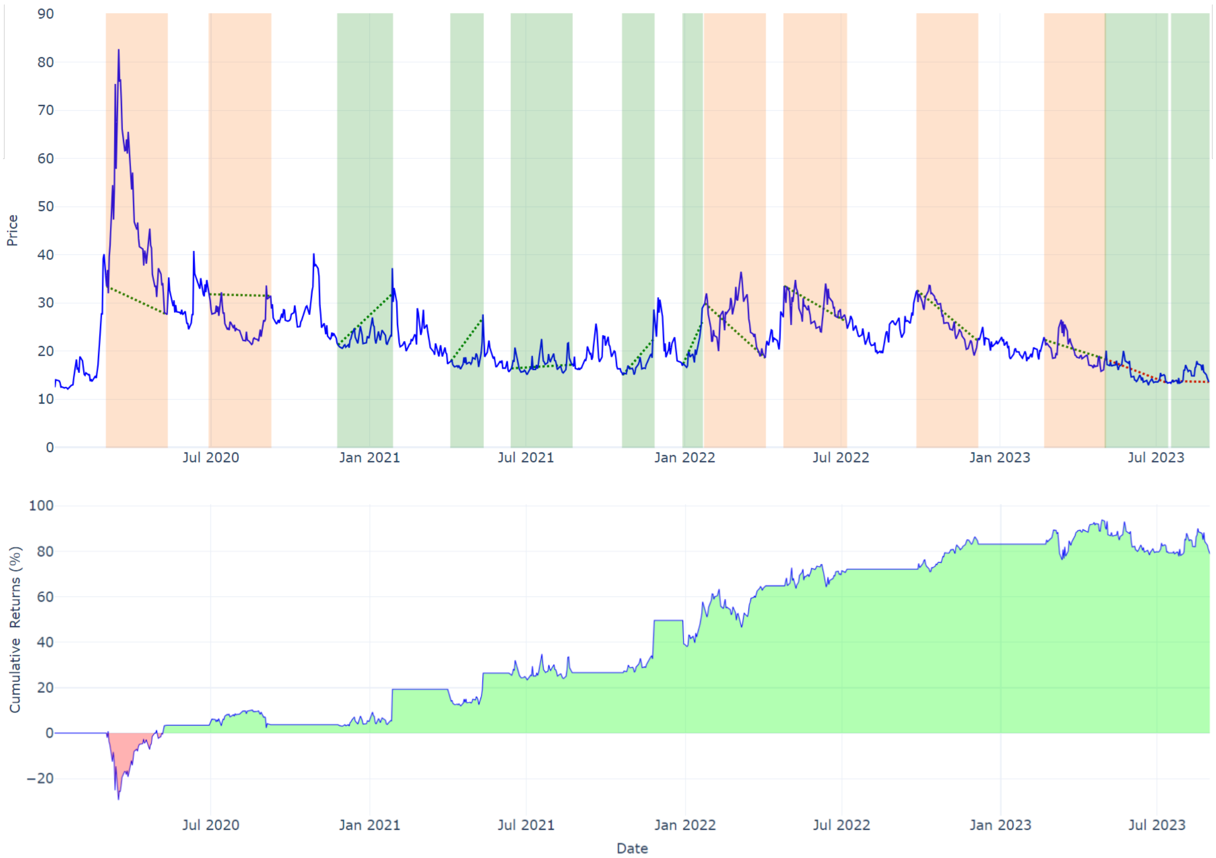


Figure 17: Test Set Simulation:
All Features custom loss function, threshold tuned on validation set

as the intention was to keep the trading simulator as simplified as possible, primarily focusing upon the trading simulation agent's decision making based on the specific predictions of the model.

5.3 Pitfalls in Feature Selection

The comparison between OHLC data and extended feature data is difficult to compare fairly. This is due to techincaly indicators by definition having a window of historical data to use for calculations. At most, some indicators had information up to 60 candles back, which in contrast to the LSTM windows of 20, highlights a potential unfair comparison. While the extended feature set offers a broader data window, this may not always translate into better performance as seen in section 4. In some cases, the additional feature engineering could potentially introduce noise rather than insight, as suggested by the longer average training times for these models. Unfortunately, this research did not include an exploratory phase of the features, in order to reduce the selection of input data, thus, reducing noise. This process was intended, but was discard due to the project increasingly growing in size.

The relatively better performance on the test data, of models using feature engineered data, might partly stem from this aspect.

5.4 Impact of Market Crashes in Test Set

The inclusion of a market crash within the test set raises questions about the fairness and representatives of the testing ground. Following the market crash in early 2020, the model's performance appeared to significantly increase figure 17. Addressing these "what-if" scenarios is crucial for a comprehensive understanding of the model's robustness across various market conditions, hence, it was included even as it could easily have been considered an outlier in the data cleaning process section 3.1.

5.5 Trading Simulations

The trading simulation in this study, while insightful, does have limitations due to its static design. Key constraints include the absence of dynamic elements like stop-loss or optimized take-profit lev-

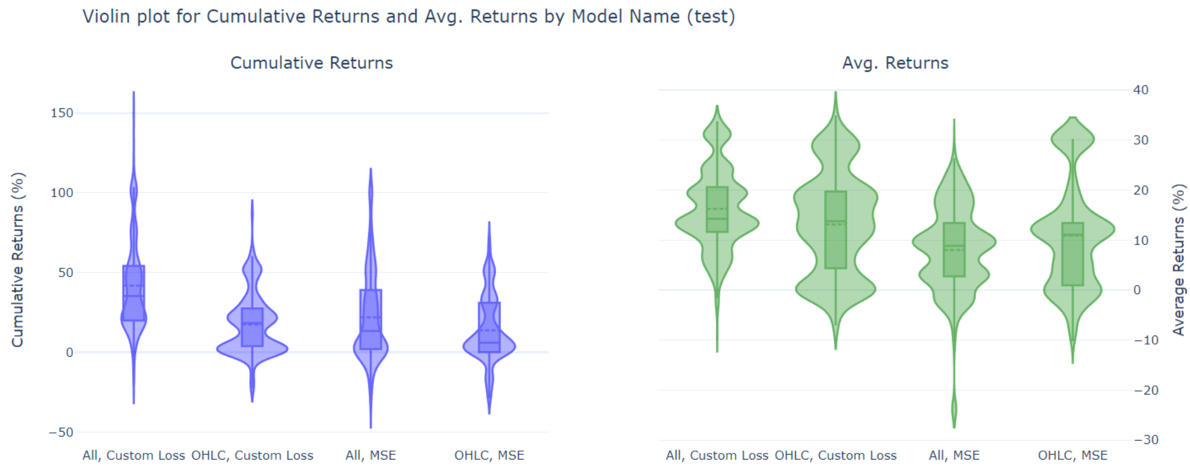


Figure 18: Simulation results Violin and Boxplot (Test set)

els, potentially limiting the simulation's reflection of real-world trading dynamics. Additionally, the choice to not implement daily trading actions, in favor of the implementing cumulative returns, may not accurately represent the predictive capabilities of the models. The approach to only have one position open at a time, emphasizes the interest in the quality of individual decisions, in contrast to the quantity of correct predictions over the entire time period. The trading simulation developed intends to evaluation the LSTM model's practical applicability. Furthermore, the thresholds chosen were constrained to intervals of 0.5, leaving out the possibility for better predictions. In a more robust setup, a machine learning algorithm could be deployed as a replacement to the current threshold agent. Lastly, the trading simulation assumes that the VIX is a tradable asset through a broker which does not require any fee pr. trade taken. In a real life scenario, options or futures would be the instrument being used, and trading fee's and premiums would add a significant strain to the results.

(DeGroot and Schervish, 2012)

5.6 Hyper parameter Tuning

The decision not to focus on hyperparameter tuning in our study leaves open the possibility that different results might have been obtained with alternative configurations. Furthermore, the choice of a 20-candle window for analysis is somewhat arbitrary and may not be optimal for capturing market dynamics. Adjusting the window size could potentially influence the model's performance significantly, for the different models, thus altering the conclusion of this research.

5.7 Custom Loss Function and Room for Improvement

The custom loss function, derived from visual intuition, shows promise but also suggests substantial room for improvement. Its current form may not optimally capture the targets as intended, which suggests the need for a more nuanced formulation, and some more research of previous work.

5.8 Subjectivity in Target Creation

The process of the target creation, also primarily based on visual intuition, introduces a degree of subjectivity in defining High and Low Targets. Pitfalls within the method late in the research process were found, specifically at instances of market crashes such as the one in 2022 figure 14. The subjective definition could affect the model's predictive capabilities, indicating that a more objective or quantitatively-driven approach might yield different outcomes.

6 Conclusion

This research embarked on a journey to unravel the following two research questions in the realm of algorithmic trading on the Volatility Index:

1. Do technical indicators have any information gain in comparison to only utilizing OHLC indicators, when used with time series neural networks?

2. Will a custom loss function for the training process be more suitable to predict data points of interest, namely peaks and valley's of the volatility index?

Our findings present a picture of mixed results. Initially, technical indicators demonstrated a negative impact in terms of the loss score, over the use of OHLC values alone. When evaluating the models performance in trading simulations, models employing just OHLC data generally achieved better performance on the validation set. Surprisingly, when these parameters were applied to the test set, the model using technical indicators as input data, and the custom loss function, significantly outperformed its counterparts. This outcome contradicts the validation and test loss scores of the models, where statistical evidence clearly favored models utilizing OHLC data, irrespective of the loss function employed table 4.

These mixed results make it challenging to draw definitive conclusions. Yet, within the confinement of our specific experimental setup, the performance of the model combining technical indicators with the custom loss function on the test set suggests a potential information gain in its predictive capabilities table 7 and figure 17. Furthermore, the statistics of results, specifically in terms of cumulative returns, were favorable in test sets for this model configuration.

These findings highlight the volatile nature of financial market predictions and the need for careful consideration of model configurations, and decision making based on their predictions. They further demonstrate that while statistical metrics provide essential insights, the ultimate test of a model's utility may differ in its real-world applicability, as seen in the trading simulations table 7. Future research could build on these insights, opening up for hyper parameter tuning, refining loss functions, and altering the feature and target engineering all together, in an attempt to enhance predictive accuracy for the volatility index.

While our research raises as many questions as it answers, it contributes to the broader dialogue about the effectiveness of neural network models in financial forecasting. Basing the conclusion on the test set results, applying technical analysis as feature's appeared to had limited effect on the predictive capability. Lastly, a clear performance increase was found when applying the optimal thresholds for the models using the custom loss function, suggesting that the loss function worked as intended.

References

Bottom Street. 2023. How to use technical analysis to predict stock market trends. https://bottomstreet.com/market-trend-prediction/. Accessed: 2023-12-13.

Jian Cao, Zhi Li, and Jian Li. 2019. Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical Mechanics and its Applications*, 519:127–139.

Chicago Board Options Exchange. 2023. VIX Index Methodology. https://cdn.cboe.com/api/global/us_indices/governance/VIX_Methodology.pdf. Accessed: 2023-12-03.

Morris H. DeGroot and Mark J. Schervish. 2012. *Probability and Statistics*, 4 edition. Pearson. See chapter 6.2 on the Law of Large Numbers.

Investopedia. 2023a. Aroon indicator. https://www.investopedia.com/terms/a/aroon.asp. Accessed: 2023-12-03.

Investopedia. 2023b. Bollinger bands. https://www.investopedia.com/terms/b/bollingerbands.asp. Accessed: 2023-12-03.

Investopedia. 2023c. Exponential moving average - ema. https://www.investopedia.com/terms/e/ema.asp. Accessed: 2023-12-03.

Investopedia. 2023. Futures definition. https://www.investopedia.com/terms/f/futures.asp. Accessed: [Insert access date here].

Investopedia. 2023a. Price rate of change (roc). https://www.investopedia.com/terms/p/pricerateofchange.asp. Accessed: 2023-12-03.

Investopedia. 2023b. Rate of return. https://www.investopedia.com/terms/r/rateofreturn.asp. Accessed: 2023-12-03.

Investopedia. Accessed 2023a. Option. https://www.investopedia.com/terms/o/option.asp. Accessed: [Insert Date of Access].

Investopedia. Accessed 2023b. Stop-loss order. https://www.investopedia.com/terms/s/stop-lossorder.asp. Accessed: [Insert Date of Access].

Investopedia. Accessed 2023c. Take-profit order. https://www.investopedia.com/terms/t/take-profitorder.asp. Accessed: [Insert Date of Access].

NorthstarRisk. 2023. Cumulative return. https://www.northstarrisk.com/cumulative-return. Accessed: 2023-12-03.

Christopher Olah. 2015. Understanding lstm networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: [Insert access date here].

Iliana Paliari. 2021. A comparison of the optimized lstm, xgboost, and arima in time series forecasting. *ResearchGate*.

Quora User. 2023. What percentage of traders use technical analysis? https://www.quora.com/What-percentage-of-traders-use-technical-analysis. Accessed: [Insert access date here].

Saxo Bank. 2023. Inspiration disclaimer. https://www.home.saxo/en-ch/legal/inspiration/inspiration-disclaimer. Accessed: 2023-12-03.

Sima Siami-Namini and Akbar Siami Namin. 2018. Forecasting economics and financial time series: Arima vs. lstm.

SteadyOptions. 2023. Using vix options to hedge your portfolio. https://steadyoptions.com/articles/hedge-portfolio-vix-options. Accessed: 2023-11-09.

Jinan Zou. 2022. Stock market prediction via deep learning techniques: A survey. *arXiv*.

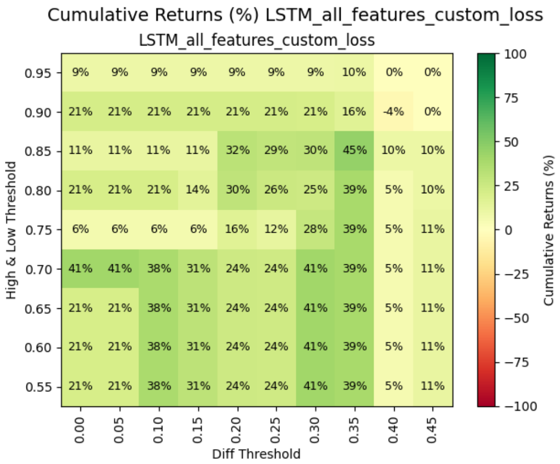


Figure 19: All Features: Cumulative returns investing 20% of available capital , on the Validation Set

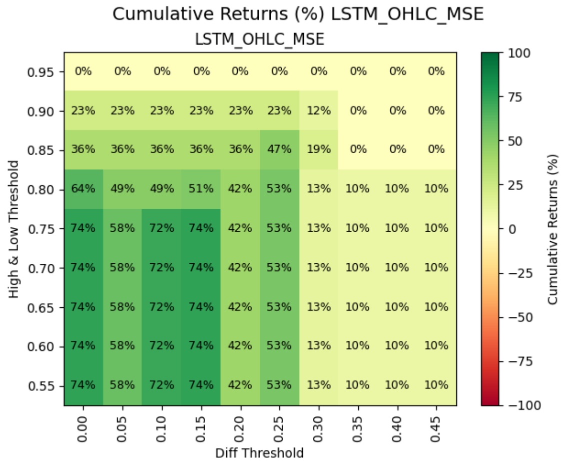


Figure 20: Open, High, Low, Close: Cumulative returns investing 20% of available capital, on the Validation Set

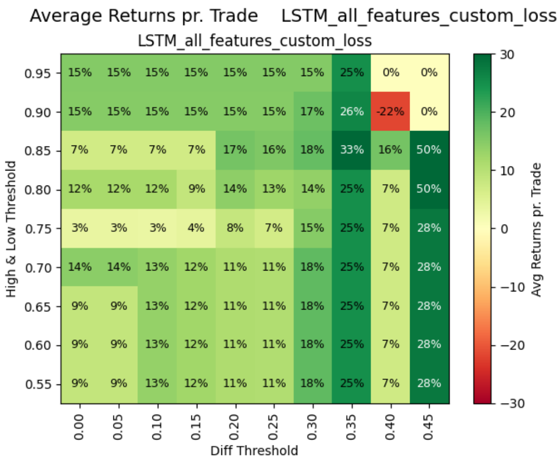


Figure 21: All Features custom loss function, on the Validation Set Validation Set

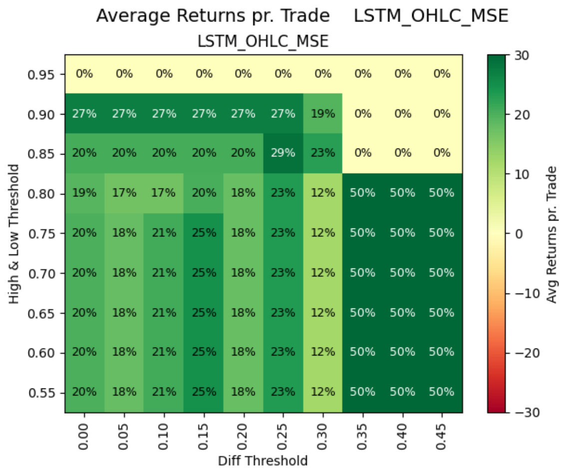


Figure 22: Open, High, Low, Close, on the Validation Set

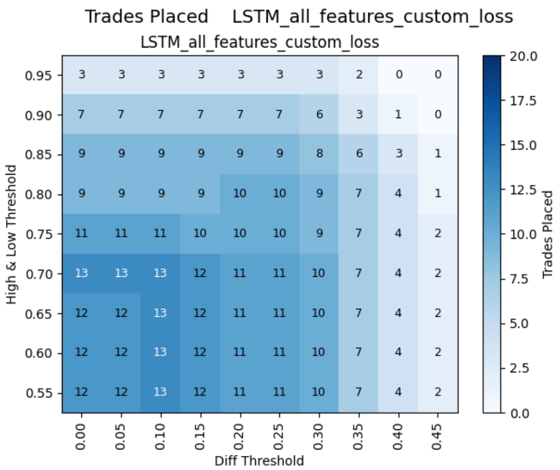


Figure 23: All Features custom loss function, on the Validation Set

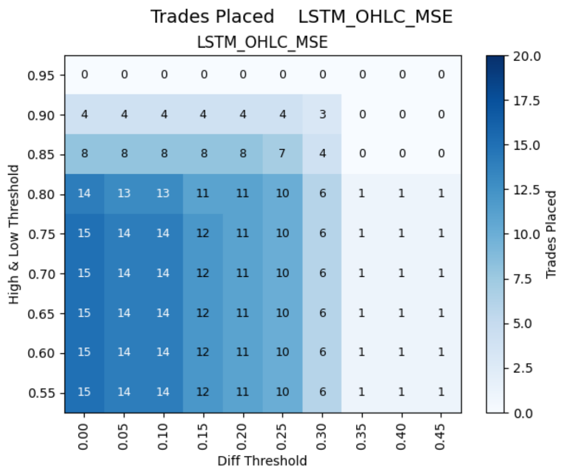


Figure 24: Open, High, Low, Close, on the Validation Set

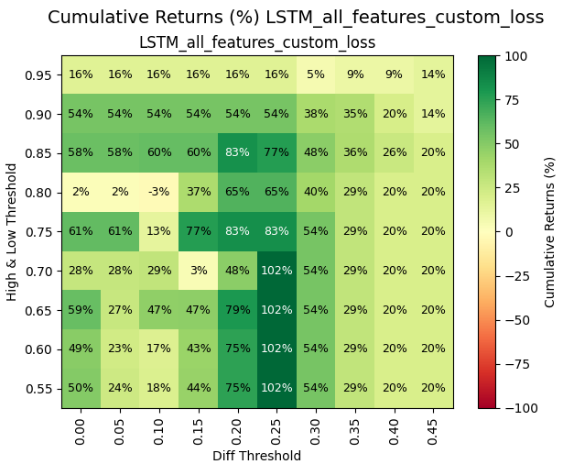


Figure 25: All Features: Cumulative returns investing 20% of available capital , on the Test Set

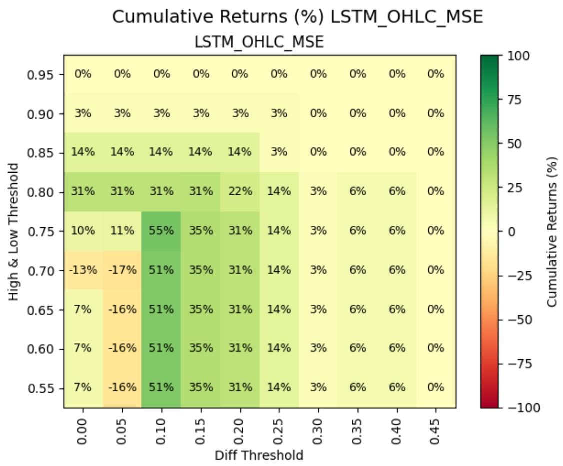


Figure 26: Open, High, Low, Close: Cumulative returns investing 20% of available capital, on the Test Set

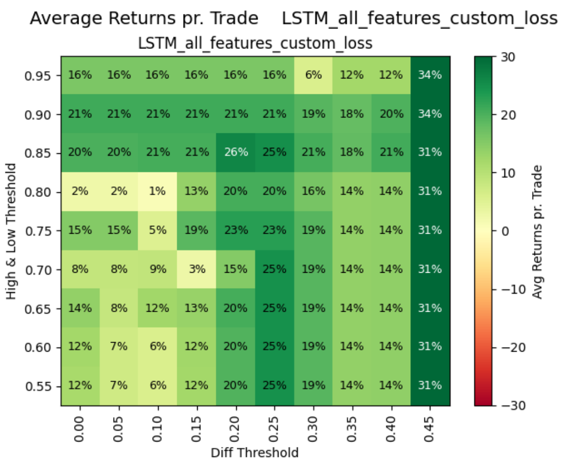


Figure 27: All Features custom loss function, on the Test Set

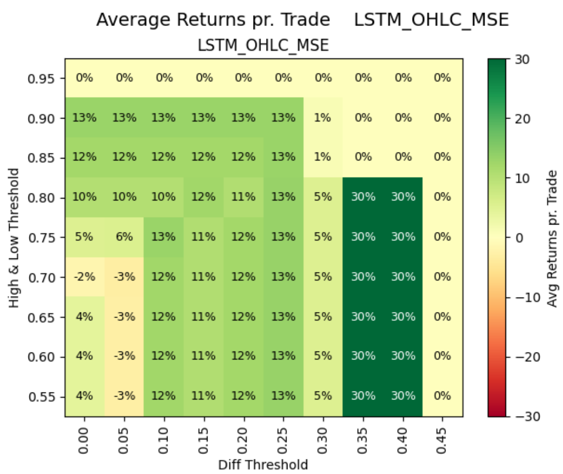


Figure 28: Open, High, Low, Close, on the Test Set

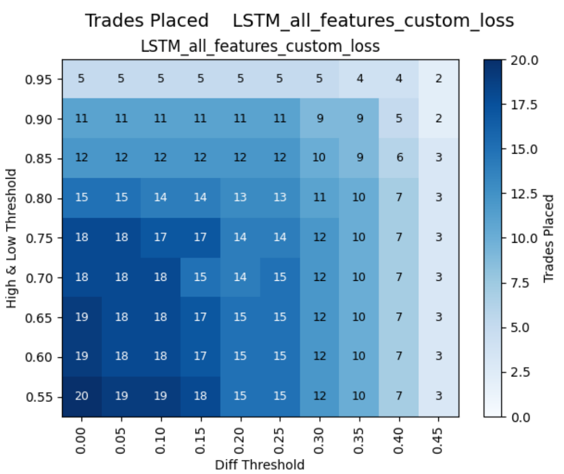


Figure 29: All Features custom loss function, on the Test Set

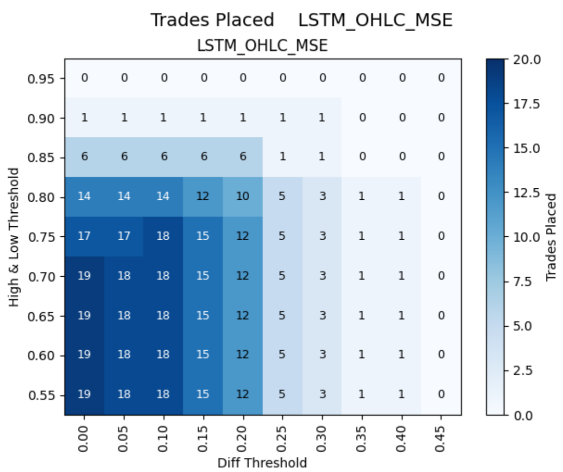


Figure 30: Open, High, Low, Close, on the Test Set