Kharagpur Data Science Hackathon

TEAM ID- 24KTJKHAP751341

Table of Contents

Problem Statement

The problem consists of developing algorithmic trading models for the BTC/USDT cryptocurrency market, aiming to outperform benchmark returns.

Overview

In the rapidly evolving landscape of cryptocurrency markets, the demand for effective trading strategies has never been more crucial. This report delves into the development and implementation of a sophisticated predictive model tailored for Bitcoin trading. Leveraging cutting-edge technologies and comprehensive market analysis, our team has engineered a model designed to enhance decision-making processes in the volatile world of cryptocurrency.

As cryptocurrency markets continue to evolve, the development of a robust and predictive trading model becomes instrumental for traders seeking a competitive edge

Key Objectives:

1. Market Insight and Analysis:
   - Provide an in-depth analysis of the current state of the Bitcoin market.
   - Identify key trends, challenges, and opportunities that influence trading dynamics.
2. Model Development:
   - Detail the methodology and algorithms employed in constructing the predictive model.
   - Highlight the key features and indicators considered for accurate market predictions.
3. Performance Evaluation:
   - Present comprehensive performance metrics of the developed model.
   - Compare and contrast the model's predictions against historical market data.
4. Risk Management Strategies:
   - Discuss the incorporation of risk management mechanisms within the model.
   - Explore how the model addresses potential pitfalls and market uncertainties.
   - Provide insights into how traders and investors can leverage the model for informed decision-making.

Data acquisition

Data for the model training and testing has been acquired from kaggle (Ref: https://www.kaggle.com/datasets/jkraak/bitcoin-price-dataset)

Dataset Details:

Total Number of Entries: 3126000

Attributes:

- **Timestamp**: Dataset has timestamps for each minute starting from 2023/08/17 till 2023/07/31. We have distributed the dataset into three parts insample_1 (2018/01/01 till 2022/01/31), out_of_sample_1 (2022/02/01 till 2022/12/31) and out_of_sample_2 (2023/01/01 till 2023/12/31), for training, backtesting and testing.
- **Open Price**: The initial price of a financial instrument at the beginning of a specific time period, such as a day or trading session.
- **High Price**: The highest price reached by a financial instrument during a particular time period, often within a day or trading session.
- **Low Price**: The lowest price reached by a financial instrument during a specific time period, typically within a day or trading session.
- **Close Price**: The last traded price of a financial instrument at the end of a specified time period, reflecting the final consensus between buyers and sellers.
- **Volume**: The total number of shares or units of a financial instrument traded during a specific time period, indicating market activity and liquidity.
- **Quote Asset Volume**: The total value of a financial instrument traded in terms of the quote currency, providing the overall value in the second currency of a trading pair.
- **Number of Trades**: The total count of individual transactions (buys or sells) during a given time period, offering insights into the frequency of trading activity.
- **Taker Buy Base Asset Volume**: The total amount of the base currency bought by market takers, where a taker places an order at the current market price.
- **Taker Buy Quote Asset Volume**: The total value of the base currency bought by market takers, expressed in terms of the quote currency.

All of these attributes except Timestamp were used for model training. Timestamp isn't taken by the model because DL models always takes integers as input

Data Preprocessing

The data preprocessing involved a transformation of the sampling frequency, transitioning from minute-level to hourly sampling. This adjustment was implemented to streamline data management and optimize the model training process. By aggregating the data at an hourly granularity, the dataset became more manageable, facilitating a more effective training experience for the model. This strategic modification enhances the model's ability to capture meaningful patterns and insights within the broader context of hourly fluctuations, contributing to improved overall performance. It aggregates the data within each hourly interval using various methods for different columns:

- 'open': Takes the first value in the interval.
- 'high': Takes the maximum value in the interval.
- 'low': Takes the minimum value in the interval.
- 'close': Takes the last value in the interval.
- 'volume', 'quote_asset_volume', 'number_of_trades', 'taker_buy_base_asset_volume', 'taker_buy_quote_asset_volume': Takes the mean (average) value in the interval.

Normalization was employed in the model-building process, specifically utilizing scalar normalization. This approach has proven to be highly beneficial in our methodology. By scaling the data through normalization, we ensure that all features contribute uniformly to the model training, preventing any particular variable from dominating the learning process. This aids in achieving a balanced and stable model, enhancing its ability to generalize patterns and make accurate predictions across various data ranges. The application of scalar normalization emerges as a pivotal step in optimizing the model's performance and ensuring robustness in the face of diverse input conditions

Model Architecture and Training

The implementation crucially uses the following aspects:

1. Window-based Model:
    a. Utilizes a window of 24 data points, each data point representing hourly sampled data.

    b. Predicts the value for the 25th sample point based on the closing real value of the previous 24 points.
    c. Achieves sequential prediction by sliding the window across the entire dataset.
2. Input Configuration:
    a. Input for the model, referred to as `trainX`, is structured with three dimensions:
        i. Size of sample data.
        ii. Size of the window (24 data points in this case).
        iii. Number of columns/attributes, totaling 9 in our dataset.
3. Output Labeling - `trainY`:
    a. The output of the model is denoted as `trainY`.
    b. `trainY` represents a singular output for the entire length of the sample data.
    c. This output encapsulates the predicted closing values for the 25th data across all processed windows.

As we delved into working with recurrent neural networks (RNNs), we found them struggling with capturing long-term dependencies due to the vanishing gradient problem. The shift to Long Short-Term Memory networks (LSTMs) became imperative. LSTMs, with their advanced architecture featuring memory cells and gating mechanisms, excel at retaining information over extended sequences. This transition has proven valuable, particularly in tasks requiring a nuanced understanding of context, such as natural language processing. The stability in training LSTMs, thanks to their gating mechanisms, has further solidified their role in our workflow, addressing the challenges posed by traditional RNNs.

Also in our work with recurrent neural networks, we found that Long Short-Term Memory networks (LSTMs) excelled at capturing temporal dependencies but struggled with information. Hence, we are transitioning to ConvLSTM, an extension that incorporates convolutional operations. ConvLSTMs offer improved handling of patterns, enhancing our models' performance in more feature extraction by using filters and scenarios where both temporal and spatial dependencies are crucial.

RNN Model:

Model Architecture:

- The model begins with an LSTM layer, which processes sequential data with `lstm_units` number of memory cells. The `dropout` and `recurrent_dropout` parameters introduce regularization to prevent overfitting.
- A Dense layer (`FC`) follows the LSTM layer with `dense_units` units, serving as a fully connected layer.
- The final Dense layer (`Output`) produces the model output with a single unit (for regression tasks).

Training Configuration:

- The model is compiled with Mean Squared Error (MSE) loss and RMSProp optimizer.
- Early Stopping callback is employed to halt training if the validation loss does not improve for a certain number of epochs (`patience=10`), restoring the weights associated with the best validation loss.

This architecture is suitable for regression tasks, where the goal is to predict a continuous output. Ensure that the input data (`window_size`, `n_features`) aligns with your dataset. Adjust hyperparameters as needed based on the characteristics of your data and the nature of your task.

Model Configuration:

- RNN Layer:
  - Units: 12
  - Dropout: 0.1 (input dropout)
  - Recurrent Dropout: 0.1 (recurrent dropout)
  - Input Shape: (window_size, n_features)
  - Activation: Default (linear)
  - Return Sequences: False (output at the last time step)
- Dense Layer (FC):
  - Units: 6
  - Activation: Default (linear)
- Output Layer:
  - Units: 1

        - Activation: Default (linear)

Training Configuration:

- Loss Function: Mean Squared Error (MSE)
- Optimizer: RMSProp
- Early Stopping:
    - Monitors: Validation Loss
    - Patience: 10 epochs
    - Restores Best Weights: Yes

Model Summary:

- The model begins with an RNN layer, employing 12 memory cells with dropout regularization.
- A dense layer follows, containing 6 units to capture complex relationships in the learned features.
- The output layer, with a single unit, produces continuous predictions for the regression task.

Training Strategy:

- The model is trained using historical Bitcoin trading data, with a window size of 24 data points and 9 features.
- The mean squared error is minimized using the RMSProp optimizer.
- Early stopping is implemented to prevent overfitting and halt training if validation loss stagnates for 10 consecutive epochs.

Code snippets

```
lstm_units = 12
dense_units = 6
window_size = 24
n_features = 9  #  9 features in our input data
output_size = 1
from tensorflow.keras.callbacks import EarlyStopping

early_stopping = EarlyStopping(monitor='val_loss',
                               patience=10,
                               restore_best_weights=True)
#rnn
model = Sequential([
    LSTM(units=lstm_units,
```

```
        dropout=.1,
        recurrent_dropout=.1,
        input_shape=(window_size, n_features), name='LSTM',
        return_sequences=False),
    Dense(dense_units, name='FC'),
    Dense(output_size, name='Output')
])
model.compile(loss='mse', optimizer='RMSProp')
model.summary()
```

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 LSTM (LSTM)                 (None, 12)                1056

 FC (Dense)                  (None, 6)                 78

 Output (Dense)              (None, 1)                 7

=================================================================
Total params: 1141 (4.46 KB)
Trainable params: 1141 (4.46 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

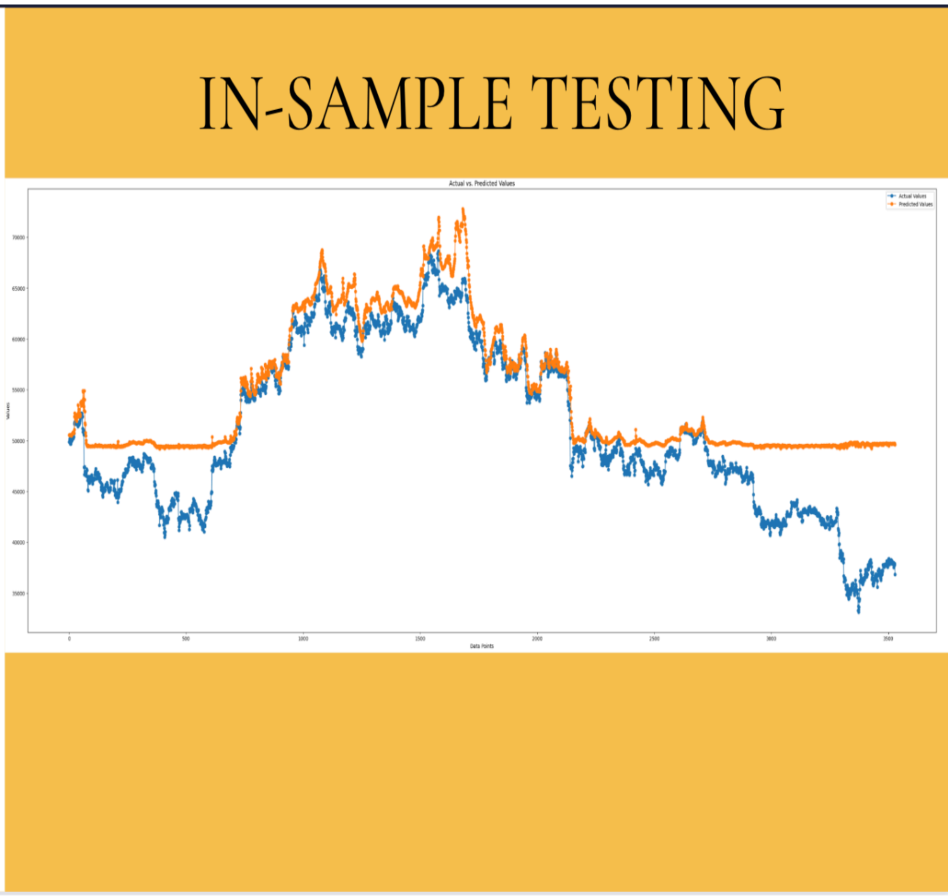
```
# Reshape trainX to match the expected input shape of the LSTM layer
trainX_reshaped = trainX.reshape((trainX.shape[0], window_size, n_features))

# Training the model
history = model.fit(
    trainX_reshaped,      # Training data input (reshaped to match LSTM input shape)
    trainY,               # Training data target
    epochs=50,            # Number of training epochs
    batch_size=20,        # Batch size
    shuffle=False,        # Whether to shuffle the training data before each epoch (here
set to False)
    validation_split=0.1, # Fraction of training data to be used as validation set
    callbacks=early_stopping,  # Callbacks for early stopping (if provided)
    verbose=1             # Verbosity mode (1: progress bar, 0: silent)
)
```

- `trainX_reshaped`: The input training data, reshaped to match the expected input shape of the LSTM layer. It now has the shape `(batch_size, window_size, n_features)`.
- `trainY`: The target training data.

- `epochs=50`: The number of times the entire training dataset is passed forward and backward through the neural network.
- `batch_size=20`: The number of samples used in each iteration during training.
- `shuffle=False`: It indicates whether to shuffle the training data before each epoch. In this case, it is set to `False`.
- `validation_split=0.1`: The fraction of the training data to be used as a validation set.
- `callbacks=early_stopping`: Callbacks are functions that can be applied at different stages of the training process. Here, `early_stopping` is presumably an early stopping callback that would stop training if a certain condition is met.
- `verbose=1`: Display training progress in a progress bar.

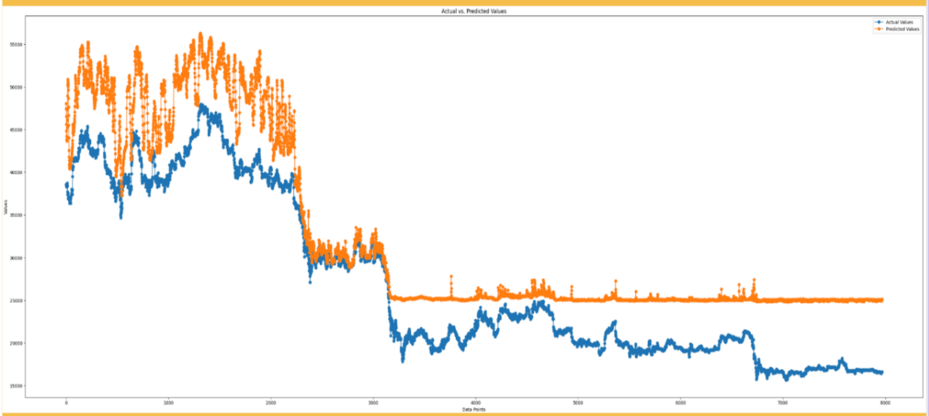
Results

```
BACKTESTING RMSE: 2664.67
Testing RMSE: 4851.21
Initial Capital : 100000

Returned Capital : 233845.8679256572

Total Gross Profit : 156621.19387807095

Total Net Profit : 156386.26208725385

Total Gross Loss (Predicted): -22775.325952413754

Total Closed Trades (Predicted): 26

Win Rate (%)(Predicted): 92.3076923076923

Loss Rate (%)(Predicted): 7.6923076923076925

Max Drawdown (Predicted): -17485.98591386623

Max Drawdown (Predicted) as a percentage: -10.16%

Average Winning Trade (in USDT)(Predicted): 6525.883078252956

Average Losing Trade (in USDT)(Predicted): -11387.662976206877

Buy and Hold Return of BTC (%)(Predicted): -1.8404491245746613

Average Annual Return of BTC (Predicted): -4.51%
Largest Losing Trade (in USDT)(Predicted): -17485.98591386623

Largest Winning Trade (in USDT)(Predicted): 14791.349877269968

Sharpe Ratio (Predicted): 0.8444958716104244

Sortino Ratio (Predicted): 1.0734531537961867

Average Holding Duration per Trade:  99.8076923076923 hours

Maximum Holding Duration of a Trade: 461 hours

Max Dip in Running Trade (Predicted): 26.294256119198568

Average Dip in Running Trade (Predicted): 4.407554390349892
```



```
BACKTESTING RMSE: 3182.95
Testing RMSE: 6177.77
Initial Capital : 100000

Returned Capital : 94488.08842765611

Total Gross Profit : 2104.163241299597

Total Net Profit : 2101.0069964376476

Total Gross Loss (Predicted): -7616.074813643485

Total Closed Trades (Predicted): 6

Win Rate (%)(Predicted): 50.0

Loss Rate (%)(Predicted): 50.0

Max Drawdown (Predicted): -5618.954620031975

Max Drawdown (Predicted) as a percentage: -5.59%

Average Winning Trade (in USDT)(Predicted): 701.3877470998656

Average Losing Trade (in USDT)(Predicted): -2538.6916045478283

Buy and Hold Return of BTC (%)(Predicted): -47.769567370414734

Average Annual Return of BTC (Predicted): -51.03%
Largest Losing Trade (in USDT)(Predicted): -5618.954620031975

Largest Winning Trade (in USDT)(Predicted): 1291.31974394212

Sharpe Ratio (Predicted): -0.4053009580228215

Sortino Ratio (Predicted): -0.41677802048031803

Average Holding Duration per Trade:  892.0 hours

Maximum Holding Duration of a Trade: 308 hours

Max Dip in Running Trade (Predicted): 10.118870576367755

Average Dip in Running Trade (Predicted): 4.337211272922269
```

OUT_OF_SAMPLE_I_TESTING




```
BACKTESTING RMSE: 1200.89
Testing RMSE: 2704.53
Initial Capital : 100000

Returned Capital : 286967.52014417003

Total Gross Profit : 202657.3242111444

Total Net Profit : 202353.33822482772

Total Gross Loss (Predicted): -15689.804066974379

Total Closed Trades (Predicted): 40

Win Rate (%)(Predicted): 90.0

Loss Rate (%)(Predicted): 10.0

Max Drawdown (Predicted): -9408.15079274858

Max Drawdown (Predicted) as a percentage: -6.34%

Average Winning Trade (in USDT)(Predicted): 5629.3701169762335

Average Losing Trade (in USDT)(Predicted): -3922.4510167435947

Buy and Hold Return of BTC (%)(Predicted): 15.656186640262604

Average Annual Return of BTC (Predicted): 28.68%
Largest Losing Trade (in USDT)(Predicted): -9408.15079274858

Largest Winning Trade (in USDT)(Predicted): 25036.264300417504

Sharpe Ratio (Predicted): 0.8360154050821368

Sortino Ratio (Predicted): 1.0582473556266432

Average Holding Duration per Trade:  74.65 hours

Maximum Holding Duration of a Trade: 373 hours

Max Dip in Running Trade (Predicted): 12.569096341742783

Average Dip in Running Trade (Predicted): 2.7202369854776878
```

OUT_OF_SAMPLE_II_TESTING

LSTM Model:

1. Sequential Model Design:
    a. The use of a `Sequential` model facilitates the creation of a linear stack of layers, ensuring a structured flow of information through the network.
2. LSTM Layers for Temporal Modeling:
    a. Two LSTM layers are incorporated into the model. The initial layer, with 64 units and ReLU activation, captures intricate temporal patterns by returning sequences. The subsequent layer, with 32 units and ReLU activation, processes the obtained sequences for further analysis.
3. Dropout for Overfitting Mitigation:
    a. A Dropout layer, with a dropout rate of 0.2, is strategically placed to curb overfitting during the training process. This enhances the model's ability to generalize patterns beyond the training data.
4. Dense Layer for Output Prediction:
    a. The addition of a Dense layer, with the number of units matching the shape of the training output, facilitates the final prediction based on the processed information.
5. Compilation with Adam Optimizer:
    a. The model is compiled using the Adam optimizer, a popular choice for its adaptive learning rates, and the mean squared error (MSE) loss function, which aligns with the regression nature of the Bitcoin trading prediction task.

Model Summary:

The model summary provides a concise overview of the architecture, detailing the configuration of each layer, the output shape, and the number of parameters. This comprehensive setup aims to empower traders and investors with a potent tool that harnesses LSTM networks to navigate the complexities of Bitcoin trading, offering enhanced predictive capabilities and a nuanced understanding of market dynamics.

```
Model: "sequential"

_________________________________________________________________

 Layer (type)            Output Shape            Param #

=================================================================

 lstm (LSTM)              (None, 24, 64)          18944


 lstm_1 (LSTM)            (None, 32)              12416


 dropout (Dropout)        (None, 32)              0


 dense (Dense)            (None, 1)               33


=================================================================

Total params: 31393 (122.63 KB)

Trainable params: 31393 (122.63 KB)

Non-trainable params: 0 (0.00 Byte)

_________________________________________________________________
```

Code Snippet

```
model = Sequential()
```

```
model.add(LSTM(64, activation='relu', input_shape=(trainX.shape[1], trainX.shape[2]), return_sequences=True))
model.add(LSTM(32, activation='relu', return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(trainY.shape[1]))

model.compile(optimizer='adam', loss='mse')
model.summary()
```

```
history = model.fit(
    trainX,           # Training data input
    trainY,           # Training data target
    epochs=5,         # Number of training epochs
    batch_size=32,    # Batch size
    validation_split=0.1,  # Fraction of training data to be used as validation set
    verbose=1         # Verbosity mode (1: progress bar, 0: silent)
)
```

- `trainX`: The input training data with shape (32182, 24, 9).
- `trainY`: The target training data with shape (32182, 1).
- `epochs=5`: The number of times the entire training dataset is passed forward and backward through the neural network.
- `batch_size=32`: The number of samples used in each iteration during training.
- `validation_split=0.1`: The fraction of the training data to be used as validation data. In this case, 10% of the training data is used for validation.
- `verbose=1`: Display training progress in a progress bar.

Results

IN-SAMPLE TESTING

BACKTESTING RMSE: 317.03
Testing RMSE: 1030.23
Initial Capital : 100000

Returned Capital : 5122467.559161452

Total Gross Profit : 5900871.923554571

Total Net Profit : 5892020.61566924

Total Gross Loss (Predicted): -878404.36439312

Total Closed Trades (Predicted): 202

Win Rate (%)(Predicted): 91.58415841584159

Loss Rate (%)(Predicted): 8.415841584158416

Max Drawdown (Predicted): -420360.216603104

Max Drawdown (Predicted) as a percentage: -8.87%

Average Winning Trade (in USDT)(Predicted): 31896.604992186873

Average Losing Trade (in USDT)(Predicted): -51670.84496430118

Buy and Hold Return of BTC (%)(Predicted): -23.253001272678375

Average Annual Return of BTC (Predicted): -48.16%
Largest Losing Trade (in USDT)(Predicted): -420360.216603104

Largest Winning Trade (in USDT)(Predicted): 335654.6452726023

Sharpe Ratio (Predicted): 0.7291464454419013

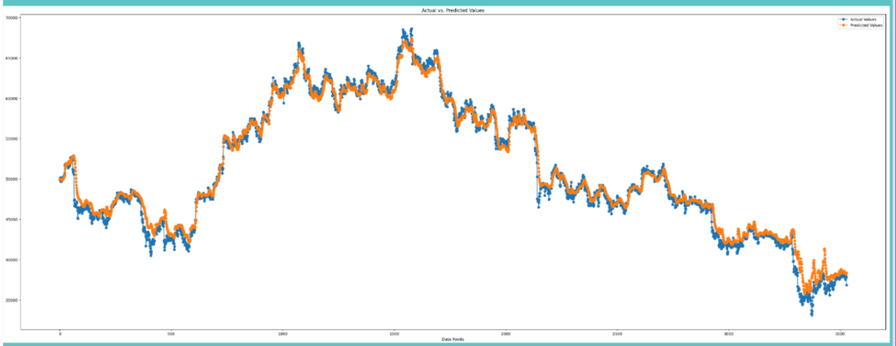
Sortino Ratio (Predicted): 0.5340971610160286

Average Holding Duration per Trade: 9.549504950495049 hours

Maximum Holding Duration of a Trade: 183 hours

Max Dip in Running Trade (Predicted): 17.86192430576182

Average Dip in Running Trade (Predicted): 1.5716033896354884



OUT_OF_SAMPLE_I_TESTING

BACKTESTING RMSE: 378.69
Testing RMSE: 1550.87
Initial Capital : 100000

Returned Capital : 84238.50371069257

Total Gross Profit : 68722.46965988142

Total Net Profit : 68619.3859553916

Total Gross Loss (Predicted): -84483.96594918886

Total Closed Trades (Predicted): 115

Win Rate (%)(Predicted): 64.34782608695652

Loss Rate (%)(Predicted): 35.65217391304348

Max Drawdown (Predicted): -16471.19935906332

Max Drawdown (Predicted) as a percentage: -19.00%

Average Winning Trade (in USDT)(Predicted): 928.68202243083

Average Losing Trade (in USDT)(Predicted): -2060.58453534607

Buy and Hold Return of BTC (%)(Predicted): -52.04668045043945

Average Annual Return of BTC (Predicted): -55.42%
Largest Losing Trade (in USDT)(Predicted): -16471.19935906332

Largest Winning Trade (in USDT)(Predicted): 6022.811119610022

Sharpe Ratio (Predicted): -0.04354085245316547

Sortino Ratio (Predicted): -0.038520044974069315

Average Holding Duration per Trade: 43.07826086956522 hours

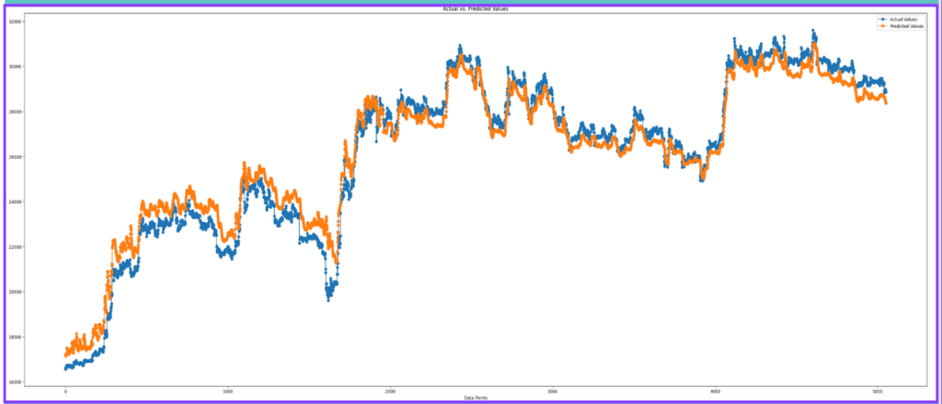
Maximum Holding Duration of a Trade: 208 hours

Max Dip in Running Trade (Predicted): 27.34725927338108

Average Dip in Running Trade (Predicted): 3.0630273916228083

```
BACKTESTING RMSE: 142.87
Testing RMSE: 679.88
Initial Capital : 100000

Returned Capital : 884484.1191290705

Total Gross Profit : 787243.6163336198

Total Net Profit : 786062.7509091194

Total Gross Loss (Predicted): -2759.4972045492614

Total Closed Trades (Predicted): 70

Win Rate (%)(Predicted): 92.85714285714286

Loss Rate (%)(Predicted): 7.142857142857142

Max Drawdown (Predicted): -2435.976099921332

Max Drawdown (Predicted) as a percentage: -1.42%

Average Winning Trade (in USDT)(Predicted): 12111.440251286458

Average Losing Trade (in USDT)(Predicted): -551.8994409098523

Buy and Hold Return of BTC (%)(Predicted): 65.2678370475769

Average Annual Return of BTC (Predicted): 138.92%
Largest Losing Trade (in USDT)(Predicted): -2435.976099921332

Largest Winning Trade (in USDT)(Predicted): 45604.16474473628

Sharpe Ratio (Predicted): 0.6039530139747283

Sortino Ratio (Predicted): 5.903590164284295

Average Holding Duration per Trade:  29.214285714285715 hours

Maximum Holding Duration of a Trade: 1076 hours

Max Dip in Running Trade (Predicted): 31.299445241052553

Average Dip in Running Trade (Predicted): 1.5357637276998177
```

OUT_OF_SAMPLE_II_TESTING



ConvLSTM

- Conv1D Layer: The model starts with a 1D Convolutional layer (`Conv1D`) with 64 filters and a kernel size of 1. The `input_shape` is set to `(trainX.shape[1], trainX.shape[2])`, which corresponds to the time steps and features of the input data.

- LSTM Layer: After the Conv1D layer, an LSTM layer with 50 units and ReLU activation is added. The LSTM layer processes the sequential information captured by the Conv1D layer.

- Dense Layer: A Dense layer with 1 neuron is added, assuming this is a regression task. If it's a classification task, you might use a softmax activation function and an appropriate number of neurons in the output layer.

- Compilation: The model is compiled with the Adam optimizer and Mean Squared Error (MSE) as the loss function. Additionally, Mean Absolute Error (MAE) and MSE are specified as metrics to monitor during training.

- Training: The model is trained using the `model.fit` method. It runs for 15 epochs with a batch size of 8. The training data is split for validation with 10%, and the learning rate reduction callback is applied.

Code snippets:

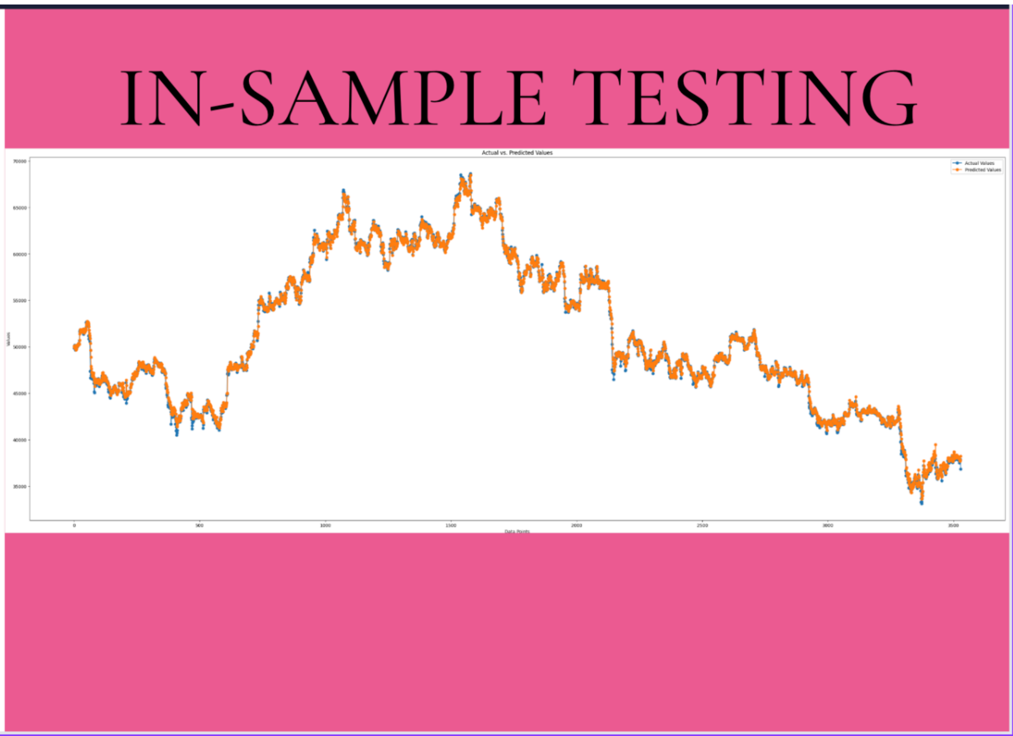
```
 from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, LSTM, Dense

# Create a Sequential model
model = Sequential()

# Add Conv1D layer with LSTM
model.add(Conv1D(filters=64, kernel_size=1, input_shape=(trainX.shape[1],
trainX.shape[2])))
model.add(LSTM(50, activation='relu'))

# Add a Dense output layer with 1 neuron (assuming it's a regression task)
model.add(Dense(1))

# Compile the model
model.compile(optimizer='adam', loss='mse', metrics=['mae', 'mse'])

# Train the model
history = model.fit(
    trainX,              # Training data input
    trainY,              # Training data target
    epochs=15,           # Number of training epochs
    batch_size=8,        # Batch size
    validation_split=0.1, # Fraction of training data to be used as a validation
set
    callbacks=[learning_rate_reduction]  # List of callbacks to apply during
training
)
```

- `trainX`: The input training data.
- `trainY`: The target training data.
- `epochs=15`: The number of times the entire training dataset is passed forward and backward through the neural network. In this case, it's set to 15 epochs.
- `batch_size=8`: The number of samples used in each iteration during training.

- `validation_split=0.1`: The fraction of the training data to be used as a validation set. Here, 10% of the training data is used for validation.
- `callbacks=[learning_rate_reduction]`: A list of callbacks to apply during training. In this case, it includes a callback named `learning_rate_reduction`. Callbacks are functions that can be applied at different stages of the training process. They can be used for various purposes, such as adjusting the learning rate, saving the best model, or early stopping.

Results

```
BACKTESTING RMSE: 130.53
Testing RMSE: 422.56
Initial Capital : 100000

Returned Capital : 108351058914.47334

Total Gross Profit : 108756244365.3919

Total Net Profit : 108593109998.84383

Total Gross Loss (Predicted): -405285450.91856694

Total Closed Trades (Predicted): 650

Win Rate (%)(Predicted): 92.15384615384615

Loss Rate (%)(Predicted): 7.846153846153846

Max Drawdown (Predicted): -199526828.6634369

Max Drawdown (Predicted) as a percentage: -0.26%

Average Winning Trade (in USDT)(Predicted): 181563012.29614675

Average Losing Trade (in USDT)(Predicted): -7946773.547422881

Buy and Hold Return of BTC (%)(Predicted): -23.652619123458862

Average Annual Return of BTC (Predicted): -48.82%
Largest Losing Trade (in USDT)(Predicted): -199526828.6634369

Largest Winning Trade (in USDT)(Predicted): 7327962432.282982

Sharpe Ratio (Predicted): 0.8922495451532831

Sortino Ratio (Predicted): 1.3180623584311322

Average Holding Duration per Trade:  1.5246153846153847 hours

Maximum Holding Duration of a Trade: 45 hours

Max Dip in Running Trade (Predicted): 19.281910066188754

Average Dip in Running Trade (Predicted): 0.5583539827113205
```



OUT_OF_SAMPLE_I_TESTING

BACKTESTING RMSE: 155.92
Testing RMSE: 331.03
Initial Capital : 100000

Returned Capital : 543739.9719185287

Total Gross Profit : 1300402.564699817

Total Net Profit : 1298451.9608527673

Total Gross Loss (Predicted): -856662.5927812883

Total Closed Trades (Predicted): 924

Win Rate (%)(Predicted): 67.20779220779221

Loss Rate (%)(Predicted): 32.7922077922078

Max Drawdown (Predicted): -110709.74861510092

Max Drawdown (Predicted) as a percentage: -20.28%

Average Winning Trade (in USDT)(Predicted): 2094.0459979063075

Average Losing Trade (in USDT)(Predicted): -2827.269283106562

Buy and Hold Return of BTC (%)(Predicted): -57.04585909843445

Average Annual Return of BTC (Predicted): -60.50%
Largest Losing Trade (in USDT)(Predicted): -110709.74861510092

Largest Winning Trade (in USDT)(Predicted): 36191.09162082599

Sharpe Ratio (Predicted): 0.12062020996044429

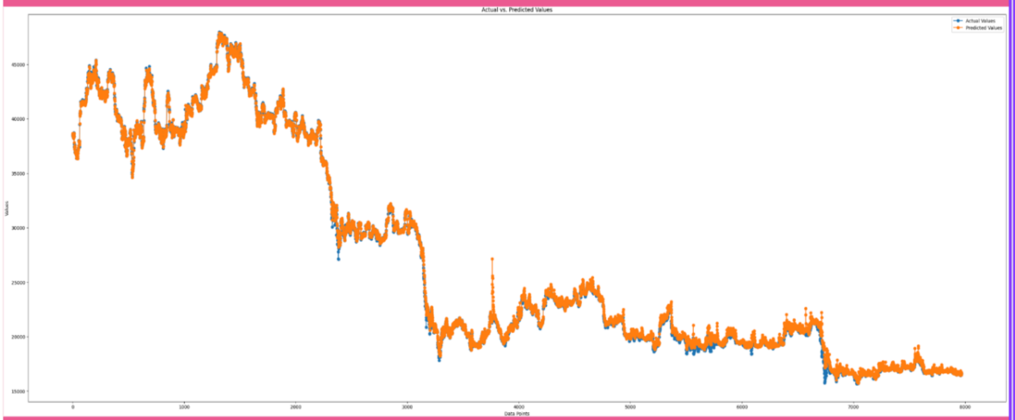
Sortino Ratio (Predicted): 0.10167097883253254

Average Holding Duration per Trade: 3.561688311688312 hours

Maximum Holding Duration of a Trade: 165 hours

Max Dip in Running Trade (Predicted): 26.178575136309128

Average Dip in Running Trade (Predicted): 1.0506603453369896



OUT_OF_SAMPLE_II_TESTING

BACKTESTING RMSE: 58.83
Testing RMSE: 161.31
Initial Capital : 100000

Returned Capital : 1754721554.6235278

Total Gross Profit : 1762338107.6319788

Total Net Profit : 1759694600.470531

Total Gross Loss (Predicted): -7716553.008450936

Total Closed Trades (Predicted): 658

Win Rate (%)(Predicted): 91.48936170212765

Loss Rate (%)(Predicted): 8.51063829787234

Max Drawdown (Predicted): -2642396.7292848825

Max Drawdown (Predicted) as a percentage: -0.26%

Average Winning Trade (in USDT)(Predicted): 2927471.9395880047

Average Losing Trade (in USDT)(Predicted): -137795.58943662385

Buy and Hold Return of BTC (%)(Predicted): 74.87246990203857

Average Annual Return of BTC (Predicted): 163.50%
Largest Losing Trade (in USDT)(Predicted): -2642396.7292848825

Largest Winning Trade (in USDT)(Predicted): 58147136.51806325

Sharpe Ratio (Predicted): 0.6767161355195948

Sortino Ratio (Predicted): 0.8598397702895488

Average Holding Duration per Trade: 2.9711246200607904 hours

Maximum Holding Duration of a Trade: 185 hours

Max Dip in Running Trade (Predicted): 17.790167774964882

Average Dip in Running Trade (Predicted): 0.4562516827877513

Backtesting:

We test our model onto in-sample data to see the fit over it. This in-sample testing helps in knowing potential mistakes in our model built.

Overfitting, a common challenge in backtesting potentially leads to poor performance in live markets. So we need to get an optimum fit for the in sample data such that it doesn't overfit it.
Its values for corresponding models are shown in results for individual models.

Risk Management

Our integrated risk management and performance evaluation approach, anchored in Sharpe and Sortino ratios, stands as a testament to our commitment to innovation and excellence in algorithmic trading strategies within the BTC/USDT market. The meticulous balance between risk and return reflects our dedication to achieving optimal risk-adjusted returns in the ever-evolving landscape of cryptocurrency trading. This report encapsulates our continuous pursuit of excellence in algorithmic trading methodologies.

1. Contextual Background:
Algorithmic trading in the BTC/USDT market demands a robust risk management strategy, considering the volatile nature of cryptocurrency markets. Our approach is anchored in a commitment to balance risk and return, ensuring adaptive strategies in response to evolving market conditions.
2. Risk Management Mechanisms:
Our risk management mechanisms are tailored to address general market risks and the specific challenges presented by the BTC/USDT market. Fundamental components include:
a. **Diversification**: Deploying diversified portfolios to spread risk across various assets, mitigating exposure to specific market movements.
b. **Position Size Limits**: Setting stringent position size limits to control the capital allocated to each trade, preventing disproportionate impact on the overall portfolio.
c. **Stop-Loss Rules**: Tactical utilization of stop-loss orders to limit potential losses during adverse market conditions. Experimentation has refined parameters to align with the unique volatility of the BTC/USDT market.
3. Backtesting and Performance Metrics:

Our risk management strategies have undergone comprehensive backtesting, assessing their effectiveness across historical data. Key performance metrics, including the Sharpe and Sortino ratios, are employed to gauge the impact of risk management on overall portfolio performance.

4. Performance Metrics Integration:

The Sharpe ratio, measuring the risk-adjusted return by considering both upside and downside volatility, provides a comprehensive view of the overall performance. Simultaneously, the Sortino ratio, focusing solely on downside risk, offers insights into performance adjustments concerning adverse market movements.

5. Experimental Findings and Verdicts:

Our experiments with risk management strategies, incorporating Sharpe and Sortino ratios, have yielded insightful outcomes:

a. **Diversification Resilience**: Diversified portfolios showcased resilience during turbulent market phases, contributing to risk mitigation and positively impacting both Sharpe and Sortino ratios.

b. **Optimized Stop-Loss Rules**: Fine-tuned stop-loss parameters strike a balance between limiting losses and avoiding premature exits, positively impacting Sortino ratios.

c. **Enhanced Risk-Adjusted Returns**: Integrated risk management strategies resulted in improved risk-adjusted returns, reflected in heightened Sharpe ratios, indicating superior overall performance.

6. Adaptive Adjustments:

Our risk management approach remains adaptive, allowing for real-time adjustments based on market conditions and ongoing performance assessments. This adaptability is crucial for navigating the dynamic nature of cryptocurrency markets.

Optimization

In our pursuit of refining algorithmic trading strategies for the BTC/USDT market, we conducted a series of experiments focusing on hyperparameter optimization for RNN, LSTM, and ConvLSTM models employed in Bitcoin price prediction. The objective was to maximize returns while maintaining acceptable risk levels. Leveraging extensive backtesting, performance metrics analysis, and systematic adjustments, we present our experimental findings and verdicts.

1. Model Overview:

Our algorithmic trading strategy relies on cutting-edge deep learning models: RNN, LSTM, and ConvLSTM. These models capitalize on sequential dependencies in historical Bitcoin price data to make accurate predictions, guiding strategic decision-making in the volatile BTC/USDT market.

2. Hyperparameter Optimization:

Our hyperparameter optimization process is characterized by a rigorous and iterative approach, encompassing the following key steps:

a. **Backtesting**: Through thorough backtesting using historical data, we evaluated model performance under diverse hyperparameter settings. This involved simulating trades based on historical data to gauge the models' historical performance.

b. **Performance Metrics**: We employed an array of performance metrics, including accuracy, precision, recall, F1 score, return on investment (ROI), and drawdown. These metrics provided a comprehensive view of both predictive accuracy and financial performance.

c. **Grid Search and Random Search**: Utilizing grid search and random search techniques, we systematically explored hyperparameter combinations to identify optimal configurations. Grid search evaluated predefined combinations, while random search explored a subset of the hyperparameter space.

d. **Cross-Validation**: To ensure model robustness, we implemented cross-validation techniques, preventing overfitting and promoting generalization across different data subsets.

e. **Validation Set Performance**: A dedicated validation set was reserved to assess model performance on unseen data, ensuring adaptability to new market conditions.

3. Experimental Findings and Verdicts:

Our experiments yielded noteworthy findings and verdicts:

a. **Optimal Configurations**: We identified hyperparameter configurations that demonstrated superior performance in terms of both predictive accuracy and financial metrics.

b. **Performance Improvements**: The optimized models showcased significant improvements in key performance indicators, translating to enhanced returns while managing risk effectively.

c. **Generalization**: The optimized models exhibited robust generalization capabilities, performing well on unseen data and diverse market conditions.