Performance Analysis of Bollinger Bands and Long Short-Term Memory(LSTM) models based Strategies on NIFTY50 Companies

Varun Seshu
Dept of Computer Science Engg
PES University
Bangalore, India
seshuvarun@gmail.com

Hritik Shanbhag
Dept of Computer Science Engg
PES University
Bangalore, India
hritik.24shanbhag@gmail.com

Samrudhi R Rao
Dept of Computer Science Engg
PES University
Bangalore, India
raosamrudhi@gmail.com

Disha Venkatesh
Dept of Computer Science Engg
PES University
Bangalore, India
dishacvenkatesh@gmail.com

Pooja Agarwal
Dept of Computer Science Engg
PES University
Bangalore, India
https://orcid.org/0000-0003-1684-2408

Arti Arya
Dept of Computer Science Engg
PES University
Bangalore, India
https://orcid.org/0000-0002-4470-0311

***Abstract*—Hedge funds and financial institutions strive to build consistently profitable automated trading strategies which can provide higher returns than stock market index baselines. This work aims to beat the stock market baselines with higher returns by building a system capable of evaluating the performance of different automated trading strategies on various different metrics. An automated strategy is a set of rules according to which a computer makes buy/sell decisions in the stock market. This work leverages predictions generated by two strategies namely Bollinger Bands and Long Short-Term Memory to aid in decision making. The LSTM strategy makes use of predictions from 250 LSTM neural networks (5 models per company), while the Bollinger Bands strategy uses close price, simple moving averages and standard deviations to make buy or sell decisions. The strategies' performance has been evaluated against historic data (backtesting) and continuous real time data (paper trading) of the stocks in the NIFTY50 index being sourced from the stock market. Upon analysis of the backtested data, it was observed that various strategy configurations have beaten the market baselines over different time periods. The results obtained by this work show that the custom strategies proposed have beaten the market baselines in 35.93% (a third) of all time periods back tested. Thus, investing in the proposed custom strategies produces higher returns than investing in the stock market index for the same time periods.**

***Index Terms*—automated strategies, baseline, paper trading, back testing, portfolio, Long Short-Term Memory(LSTM), Models, Simple Bollinger Bands, algorithmic trading, orders, signals, trades, alpha, Artificial Intelligence(AI)**

I. Introduction

A majority of the orders executed in today's stock exchanges are generated by computers executing automated strategies. An automated trading strategy is a set of rules according to which a computer makes buy/sell decisions, it allows for faster execution of orders with minimum slippage, but most trading systems on the market do not provide substantial flexibility in terms of modification of parameters to build the desired strategy. Thus it was decided to build a system to develop, backtest and paper trade custom-built strategies that involve technical indicators and machine learning.

The system has been structured to accommodate multiple features that are relevant and necessary for the functioning of an online trading system and is designed to assist any individual that harbors a basic understanding of trading on the stock market. These features include custom-built automated strategies for trading, backtesting and paper trading.

In this work, two automated trading strategies are proposed - a bollinger bands strategy and a LSTM models-based strategy [1]. Each strategy is parameterized and is initialized with randomized configurations. The dataset used in this work consists of daily stock ticker data with daily high, low, open and close prices for the past three years. Furthermore, data on stock volume was collected for the same time period.

Backtesting is used to evaluate the performance of the two proposed strategies on historical data [9]. The results of the backtest indicate how well the given strategy would likely have performed over that time period. The system's backtesting component evaluates a strategy's performance by taking in two inputs - namely the parameterized strategy and historical data over a certain time period. Historical orders are generated based on the strategy's rules. This component also outputs a detailed report providing insights into the performance of that particular strategy for the specified time period. The system runs 40 backtests on each company in the NIFTY50 index, which consists of 50 companies, resulting in 2000 backtests overall. These backtests have resulted in the generation of 598823 trades, 598823 signals and 1197064 orders.

Paper trading is a process used to evaluate the performance of a strategy against continuous historical and real-time data being sourced from the stock market.The paper trading component of the system tracks orders generated by the backtested strategy configurations and evaluates their performance in live market conditions.

Alpha refers to the profit or potential profit from an investment that takes into account the baseline increase or decrease (in percentage) of that particular investment over the same time period. By Analyzing the data produced by backtests, various strategy configurations have produced alpha over different time periods. In this analysis, the baselines are first calculated for a stock using a simple buy and hold strategy, then time periods (such as quarters, months and years) where the strategies outperform the calculated stock market baselines are found. The proposed custom strategies have been able to outperform the baselines (produce alpha) in a majority of the time periods measured.

The rest of the paper is organised as follows: Section II describes the literature survey done on the same above mentioned problem. Section III chronicles the proposed methodology used in this work highlighting all the important steps that includes data collection, building automated strategies, running backtests, tracking orders real time, baseline calculations and Alpha calculations. Section IV talks about the results obtained in the conducted experiments. The conclusion of this work is discussed in Section V.

II. Literature Survey

Moghar et.al [1] performed Stock market prediction using a LSTM recurrent neural network.These networks were used to generate long term forecasts of GOOGL and NKE stock ticker's prices.The network consists of 4 LSTM layers, each with 96 neurons with a dropout layer between each of them, followed by a single dense neuron for the output layer. They compares predictions of the model after training the network for 12, 25, 50 and 100 epochs - MSE of the model decreases with an increase in the number of training epochs. Their visualizations show that the LSTM model traces the evolution of opening prices of both assets.

Huang et.al [2] used statistical and machine learning methods in automated trading systems. They reviewed trading systems built by various methods and empirically evaluated the methods by grouping them into three types: technical analyses, textual analyses and high-frequency trading. The authors discussed statistical methods like Pairs Trading(PST) and machine learning applications like Genetic Algorithm (GA), Artificial Neural Networks (ANN) and Support Vector Machines(SVM). The field of High Frequency Trading(HFT) was also discussed.

Ni et.al [3] presents an efficient implementation of the backtesting of a trading strategy using Simplified Trading Strategy (STS) algorithm and Parallel Genetic Algorithm (PGA) which is fine tuned based on a thorough analysis of the trading strategy. They observed that the refined Parallel Genetic Algorithm (PGA) took less running time than the Simplified Trading Strategy(STS) algorithm. In real world scenarios, STS algorithm was observed to take nearly 40 hours whereas PGA hardly took 1-2 hrs. However, the authors have not taken brokerage fees into account. Furthermore, they assume that the same concepts used can be extended to the backtesting of many other trading algorithms.

Zhang et.al [4] used Quantitative media (blogs and news) data, generated by NLP to perform a comprehensive and comparative study on how company related news variables anticipate or reflect the company's stock trading volumes and financial returns. Based on the study, a market-neutral strategy was built which gave consistently favorable returns with low volatility over a long period. Their findings show that the positive sentiments of today's news has almost no ability to predict short term returns (such as the next day). Furthermore, they observed that large and small firms showed greater returns than medium-size firms. They concluded by emphasising that a considerable amount of stock market metrics are substantially associated with raw or derived blog/news variables.

Adebiyi et.al [5] used daily candlestick stock prices in their dataset. The hybridized approach combines technical and fundamental analysis variables. They have used neural networks to predict the stock price values at different time periods. They have compared the predictions of various different configurations of neural networks with different architectures and different input variables. All networks used sigmoid as activation function and were trained for 10000 epochs. The hybridized approach has the potential to enhance the quality of decision making of investors in the stock market by offering more accurate stock prediction compared to existing technical analysis based approaches. Faizul F et.al [6] suggests a neural network model to formulate trading decisions such as buying or selling of shares by removing subjectivity as most trading activities are based on market sentiment, technical analysis and imitative behaviour.The model they proposed, comprises of two parts - technical analysis and fundamental analysis. The model proposed by them does not consider the behavioral factors while modeling in lieu of imparting subjectivity. The model they proposed uses technical analysis to seek out price patterns for profitable short runs and fundamental analysis for valuation. The fundamental analysis aspect of their model may be prone to certain subjectivity due to dissimilar holding periods and discount rates by different investors.

Dacorogna et.al [7] used the main ingredients of simple trading models in genetic algorithm optimization. They have combined technical indicators used with respect to the trading model with a genetic algorithm. Operations such as *, +, - and / were performed on the indicators to generate a sensible trading model. They use a mapping algorithm to generate the right genes for the genetic algorithm.

Przemysław Ryś et.al [8] explains machine learning used in algorithmic trading strategy optimization. Their paper discusses sensitivity of a strategy's performance to parameter changes. The hypothesis verified in their paper indicated that the machine learning methods selected strategies with highest evaluation criterion but did so in significantly shorter execution

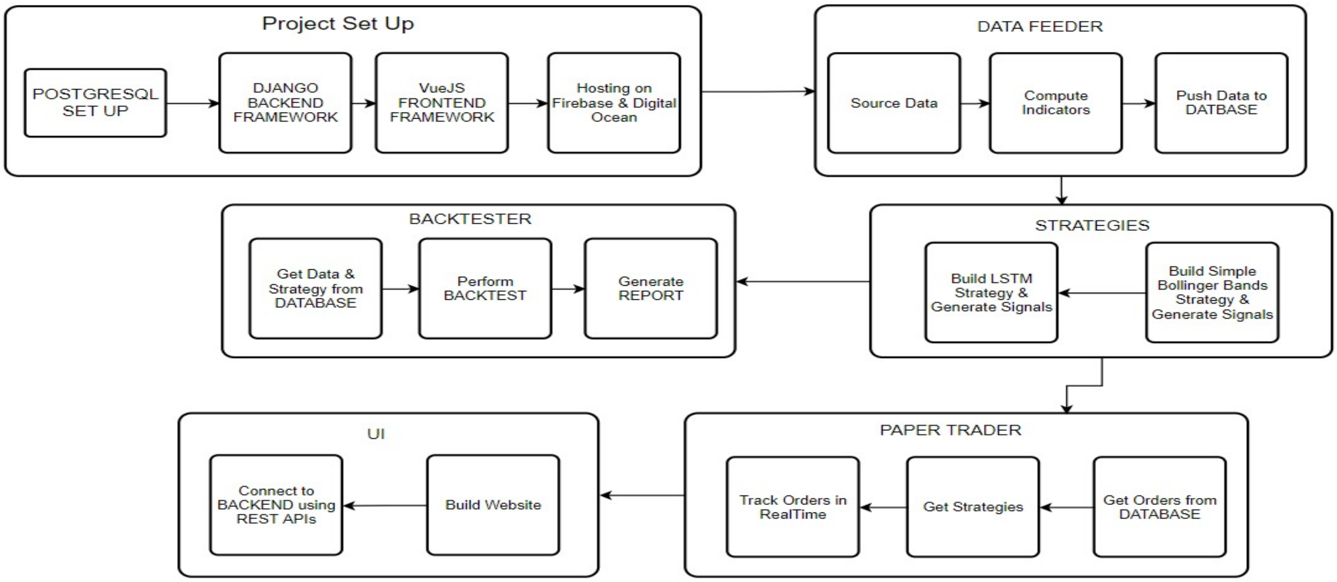

Fig. 1. Design of the proposed system

time as compared to the brute force method. They have created and analyzed machine learning methods, modeled to best fit the specified optimization criteria.

III. Proposed Methodology

The first component of the system is the configuration where the postgresql database is set up. Django is chosen to be the backend framework and the frontend framework is VueJS. The system is hosted on Firebase and Digital Ocean.

Furthermore, in the data feeder component, the data is sourced,indicators are computed and data is pushed to the database. Next, the Simple Bollinger Bands (SBB) and LSTM Strategies are built. These two strategies generate signals. Furthermore, the strategies component is branched out into two components, backtester and paper trader.

The backtester component takes historical data from the database and a strategy as inputs to perform backtests. Backtesting is a process where a strategy is run on historic data to generate historical signals, orders and trades. Furthermore, it generates a backtest report based on the strategies performance on the historical data.

In the paper trader component, the orders and strategies are taken from the database and the orders are tracked real time i.e the orders are evaluated and tracked in real time market scenario. An interactive UI was built. Using REST APIs the website was connected to the backend as shown in Fig.1.

A. Data collection

The input data for the system comes from the data feeder component. The time series data is sourced from Yahoo Finance. Three years of historical stock market daily candlestick data has been collected for each company in the NIFTY50 index of the National Stock exchange in India. Further, the system sources up to date data for each company in the NIFTY50 index. Candlestick data for a particular period of time consists of the open, high, low and close prices of that stock in that particular time period. The features of the data set are date, daily open, high, low, close and adjusted close prices and volume of shares.

B. Building Automated Strategies

An Automated Strategy is a set of rules according to which a computer makes a decision on whether to generate a buy or sell signal. In this work, two automated Strategies were created - Simple Bollinger Band (SBB) Strategy and LSTM Strategy.

```
if Close_price > n standard deviations above Simple Moving Average (with time
period t)
        Generate SELL Signal

else if Close_price < n standard deviations below Simple Moving Average (with time
period t)
        Generate BUY Signal

else
        Generate FLAT Signal
```

Fig. 2. Pseudocode of simple bollinger bands strategy

Fig. 2 shows the Simple Bollinger Band Strategy Pseudocode where the strategy generates a BUY signal if the close price of a stock is lesser than n standard deviations below the simple moving average.The strategy generates a SELL signal if close price of stock is greater than n standard deviations above simple moving average. In all other cases strategy generates a FLAT signal.

```
if predicted_return_percentage > threshold
        Generate BUY Signal

if predicted_return_percentage < threshold
        Generate SELL Signal

else
        Generate FLAT Signal
```

Fig. 3. Pseudocode of LSTM strategy

Fig. 3 illustrates the pseudocode for the LSTM strategy. Overall, the system has 250 LSTM neural networks (5 models per company). The 5 models of each company have been trained on that particular company's data. The hyperparameters of these models have been randomized. Each strategy configuration of the LSTM strategy corresponds to an unique model. This strategy generates predictions on the return percentage of a stock for a particular time period. This strategy generates a BUY signal if the predicted return percentage is greater than a parameterized threshold. The strategy generates a SELL signal if the predicted return percentage is lesser than a parameterized threshold. In all other cases the strategy generates a FLAT signal.

C. Running Backtests

The backtesting component takes a parameterized strategy and data over a period of time as input. It generates orders on historical data provided as input according to the strategy's rules and outputs a comprehensive report on the strategy's results over a predefined period of time. 2000 backtests were conducted resulting in approximately 1.2 million orders being placed, 6 lakh signals being generated and 6 lakh trades being completed.

```
Steps:
  • Calculate indicators and generate predictions on historical data
  • Generate signals using strategy (3 possible types - BUY, SELL, FLAT)
  • Calculate take profit and stop loss prices for all BUY and SELL signals
  • For each signal
      ○ if at least max_holding_period number of days are ahead of the signal
        in the historical dataset
          ■ Generate order from signal on the next day
          ■ For max_holding_period number of days ahead of the day of
            signal generation
              • If take_profit or stop_loss price is hit
                  ○ Generate exit order
                  ○ Complete trade
      ○ else
          ■ go to next signal
```

Fig. 4. Backtesting Algorithm

Fig. 4 depicts the backtesting algorithm. First, the indicators are calculated and predictions are generated on historical data. Next, the backtested strategy configuration generates signals (such as Buy, Sell or Flat) on the historical data. After signal generation, the take profit and stop loss prices are calculated for all Buy or Sell signals. For every signal, if at least the maximum holding period (a parameter) number of days is ahead of the signal in the historical dataset, then an order is generated on the next day. For every maximum holding period (a parameter) number of days ahead of the day of signal generation, the exit order is generated and the trade is completed if and only if take profit or stop loss price is hit. If the maximum holding period (a parameter) number of days is not ahead of the signal in the historic dataset then the algorithm skips to the next buy/sell signal until all signals are exhausted.

D. Tracking orders real time

The next important component in this work is paper trading, where strategies are tested in real-time market scenarios after a strategy has performed well in back testing. The paper trading algorithm runs once a day. It tracks and evaluates orders generated by SBB Strategy and LSTM Strategy in real-time market conditions and provides various performance metrics for tracked and historical orders.

```
Steps:
  • For each live paper order
      ○ Generate Exit order if
          ■ max_holding_period or take profit or stop loss limit is reached
          ■ Save paper trade
  • For each paper signal in DB
      ○ Execute signal (Generate a paper order) at today's close price
  • For each company
      ○ Source latest data
      ○ For each strategy
          ■ Calculate indicators and generate predictions
          ■ Generate paper signals using strategy (3 possible types - BUY,
            SELL, FLAT)
```

Fig. 5. Paper Trading Algorithm

Fig. 5 shows the paper trading algorithm. The algorithm states that for each live paper order, if the maximum holding period (a parameter) or take profit or stop loss limit is reached, then an exit order is generated and the paper trade is saved. Next, for each paper signal in the database at today's close price, the signal is executed and a paper order is generated. For every company, the latest data is sourced and furthermore for every strategy the indicators are calculated and predictions are generated. The algorithm produces Buy, Sell or Flat paper signals using a strategy configuration, which are later evaluated to complete a trade.

E. Baseline calculation

The calculation of baselines helps in understanding whether the two proposed strategies beat the stock market baselines. The success of a strategy configuration is determined largely by the amount (percentage points) by which it beats market baseline returns. To calculate the baseline return for a particular time period, a simple buy and hold strategy was implemented where a share of the stock at the beginning of the time period was bought and sold at the end of the time period. The return percentage of this trade is the baseline return for that particular stock at that particular time period. The above process was repeated for all 50 stocks and 69 time periods (such as quarters, months and years).

F. Alpha calculation

Alpha is a metric that helps to determine the success of a strategy and compare different strategies. The Calculation of Alpha values is done across all time periods mentioned above for all backtests. For a particular time period, alpha is calculated by taking the difference between company backtest returns and company baseline returns if a company's backtest returns are greater than 0 and greater than company baseline.

IV. Experimental Results and Discussion

A total of 2000 backtests were conducted using the system, which has generated 5,98,823 trades, 5,98,823 signals and 11,97,064 orders. Furthermore, 1513 live paper trades were generated with 1600 strategy configurations that are being traded live. Of 1,38,000 backtest time periods analyzed, 46,714 have produced alpha, which implies that custom strategies have beaten the baselines in 35.93% (a third) of all backtest time periods.

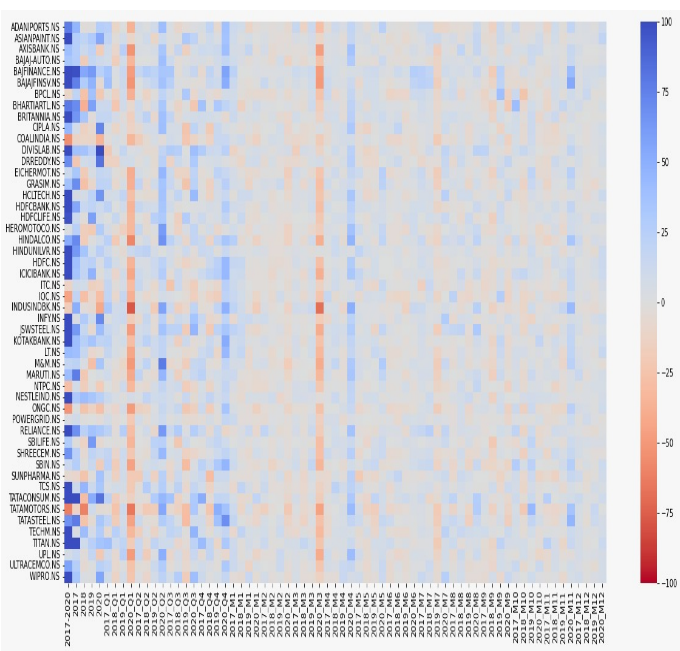

Fig. 6. Company baseline returns

From a simple analysis of baseline data as given in Fig. 6, shows companies' baseline returns against different time periods given in the x-axis. Darker blues indicate higher positive returns, while darker reds indicate higher negative returns. It can be seen that in the first quarter of 2020 and the month of March in 2020 have streaks of red, from which it can be inferred that those were loss-making time periods for most companies. Further, it was concluded that this was partly caused due to increased market volatility induced by the effects of covid 19 restrictions on the economy at that time.

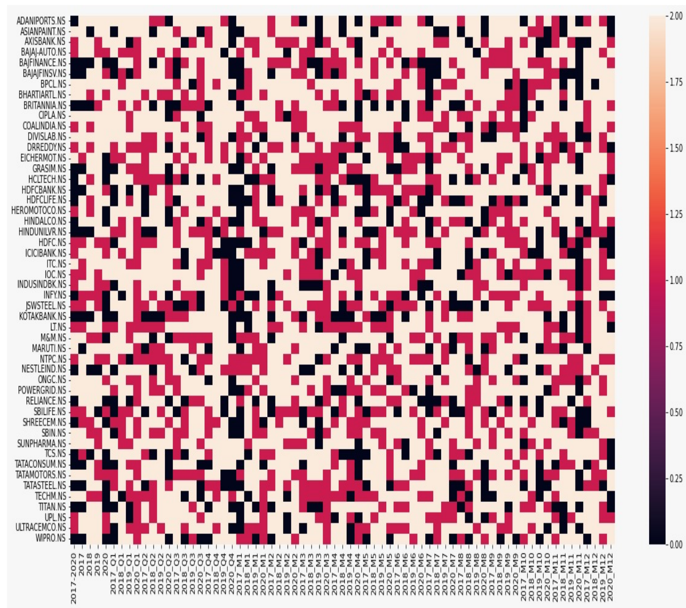

Fig. 7. Best performing strategies across all companies and time period

Fig. 7 shows the best performing strategy for every company and every time period. The black color shows the company time periods where the company baselines were not beaten by any strategy, while darker pink color denotes the company time periods where the bollinger bands strategy outperformed all other strategies and the lighter pink shows the company time periods where the LSTM strategy performed the best. From this visualization, it is apparent no singular strategy has been able to consistently outperform the market baselines. This figure determines the best strategy for a given company and time period.

Next, the custom built strategies i.e Simple Bollinger Bands and LSTM-based strategy are combined and their returns are compared with the baselines across the NIFTY50 index. Fig. 8 shows the results for the same. The light pink color denotes the company time periods when the custom strategies have outperformed the baseline, while the black denotes the contrary.

Table I compares the performance of the custom-built strategies to that of the baselines. The column named "Profitable

TABLE I
PERFORMANCE OF CUSTOM-BUILT STRATEGIES

Strategy	Profitable TP	Alpha TP	Percentage of Profitable TP	Percentage of Alpha TP	Total TP
LSTM	29766	23074	43.14	33.44	69000
SBB	30195	23640	43.76	34.26	69000
Combined	59961	46714	43.45	35.93	138000

TP" refers to the total count of time periods during which the corresponding strategy was profitable, while the "Alpha TP" column indicates the number of time periods during which the strategy outperformed the baselines. Similary "Percentage of Profitable TP" and "Percentage of Alpha TP" refer to the percentage of time periods that are profitable and alpha generating respectively. The last column "Total TP" refers to the total number of time periods backtested for a corresponding strategy.

Out of a total of 69,000 time periods it was observed that the LSTM strategy was profitable in 43.14% of the time periods and outperformed baselines in 33.44% of the time periods. Similarly, the SBB strategy has a total profitable time periods count of 30,195 or 43.76% of the total time periods and outperformed the baselines in 34.26% of the time period. When both strategies are combined, the percentage of time periods that produce alpha increases to 35.93%.

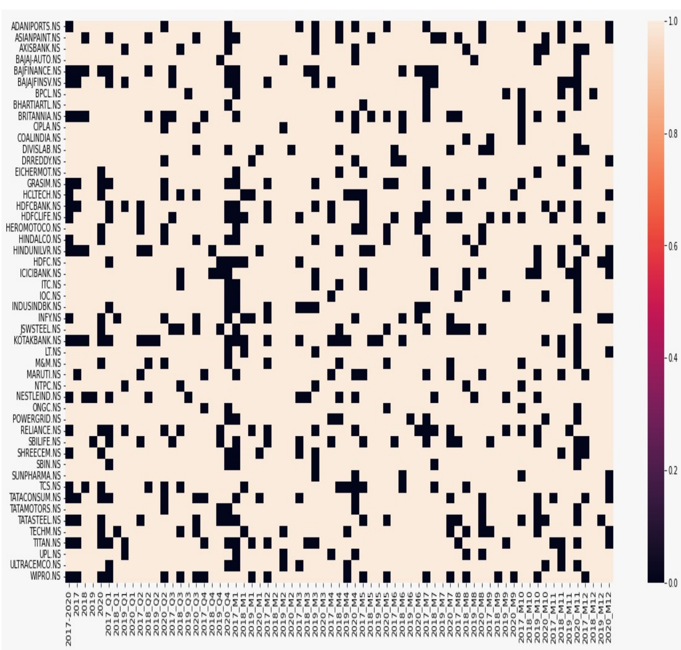

Fig. 8. Baselines vs Custom Strategies performance across NIFTY50

The heatmap in Fig. 8, there are a few backtest time periods where the baselines have outperformed the custom strategies (denoted by the black squares) in a sea of light pink, which clearly depicts that the custom strategies beat the baselines for most of the time periods. In February 2019, custom strategies have outperformed and consistently given profits across companies. From 2017 to 2020, the custom strategies have performed exceptionally well for many companies like Divislab and Sunpharma.

Fig. 9. Best performing Strategy Configurations across all companies and time periods

As depicted in Fig. 9, a different color is assigned to each of the 40 strategy configurations, with black for the baseline. For each company and time period combination, Fig. 9 highlights which strategy configuration has performed the best. From the Fig. 9, it can be clearly seen that there is no discernible pattern. This helps visualize the dynamic nature of the stock market, which is an emergent system where different strategy configurations can outperform others at different market conditions.

V. CONCLUSION

The stock market is a dynamic system and building consistently profitable strategies which can beat the stock market baselines is a continuous effort. Over 2000 backtests and 69 time periods have resulted in 1,38,000 total backtested time periods, this work finds that the custom strategies proposed have beat the market baselines in 46,713 time periods or 35.93% (a third) of all the backtested time periods. Taking the best performing strategy configurations for each time period, custom strategies are able to outperform stock market baselines for a majority of the time periods.

The future scope of this work includes live trading in the market, simulate slippage and brokerage in backtests and paper trades. Additionally, better simulations of the exchange's order book can provide better orders entries and exits than the close price, as currently orders enter the day after the entry signal is generated and exit the day when exit conditions are met. Furthermore, paper orders can be executed and evaluated more accurately with market data that is updated every minute, rather than updation on a daily basis.

REFERENCES

[1] "Moghar, Adil and Mhamed Hamiche. "Stock market prediction using lstm recurrent neural network." Procedia Computer Science 170 (2020): 1168-1173."

[2] "Huang, Boming, Yuxiang Huan, Li Da Xu, Lirong Zheng and Zhuo Zou. "Automated trading systems statistical and machine learning methods and hardware implementation: a survey." Enterprise Information Systems 13, no. 1 (2019): 132-144."

[3] "Ni, Jiarui and Chengqi Zhang. "An efficient implementation of the backtesting of trading strategies." In International Symposium on Parallel and Distributed Processing and Applications, pp. 126-131. Springer, Berlin, Heidelberg, 2005."

[4] "Zhang, Wenbin and Steven Skiena. "Trading strategies to exploit blog and news sentiment." In Proceedings of the International AAAI Conference on Web and Social Media, vol. 4, no. 1. 2010."

[5] "Adebiyi, Ayodele A., Charles K. Ayo, Marion O. Adebiyi and Sunday O. Otokiti. "Stock price prediction using neural network with hybridized market indicators." Journal of Emerging Trends in Computing and Information Sciences 3, no. 1 (2012): 1-9."

[6] "Noor, Faizul F. and Hossain, Mohammad Farhadahangirnagar Review, Part II: Social Science, Vol. XXIX, pp. 177-194, 2005."

[7] "Dacorogna, Michel M. "The Main ingredients of simple trading models for use in genetic algorithm optimization." Internal document 2500 (1993): 03-22".

[8] "Ryś, Przemysław, and Robert Ślepaczuk. "Machine Learning Methods in Algorithmic Trading Strategy Optimization–Design and Time Efficiency." Central European Economic Journal 5.52 (2018): 206-229."

[9] "Chen, Cathy WS, Tsai-Yu Lin, and T. Y. Huang. "Incorporating volatility in tolerance intervals for pair-trading strategy and backtesting." Journal of Risk Model Validation 13.1 (2019): 63-94."

[10] "Du, Zaichao, and Pei Pei. "Backtesting portfolio value-at-risk with estimated portfolio weights." Journal of Time Series Analysis 41.5 (2020): 605-619."

[11] "Deng, Kaihua, and Jie Qiu. "Backtesting expected shortfall and beyond." Quantitative Finance 21.7 (2021): 1109-1125."

[12] "Nelson, David MQ, Adriano CM Pereira, and Renato A. de Oliveira. "Stock market's price movement prediction with LSTM neural networks." 2017 International joint conference on neural networks (IJCNN). IEEE, 2017."

[13] "Carlier, Francesco. "A Simple Options Trading Strategy based on Technical Indicators." International Journal of Economics and Financial Issues 11.2 (2021): 88-91."

[14] "Jirapongpan, Ronnachai, and Naragain Phumchusri. "Prediction of the Profitability of Pairs Trading Strategy Using Machine Learning." 2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA). IEEE, 2020."

[15] "Pang, X., Zhou, Y., Wang, P., Lin, W., & Chang, V. (2020). An innovative neural network approach for stock market prediction. The Journal of Supercomputing, 76(3), 2098-2118."