Strike Price Recommendations for NIFTY 50 Index Options

Dr Arul Anand N[1], Ashwin Kumar T G[2], Aditya Varma[3], Jeyam Palaniappan D[4], Sanjeeva Shekar[5], Sashti Amar R A[6]

Department of Computer Science and Engineering
PSG College of Technology
Coimbatore, Tamil Nadu 641004, India

[1]naa.cse@psgtech.ac.in, [2]tgashwinkumar@gmail.com, [3]adityavarma1404@gmail.com, [4]jeyam1001@gmail.com, [5]sanjeevashekar1@gmail.com, [6]sasti244@gmail.com

***Abstract*—The burgeoning complexity and inherent uncertainty of stock market trading strategies, particularly for Index Options, necessitates the need for innovative solutions. Deep learning techniques hold immense promise in navigating this dynamic landscape, offering potent tools for optimizing option selections and strike prices. This project delves into the potential of harnessing a novel Convolutional Neural Network (CNN) architecture to predict optimal price ranges for the NIFTY 50 index, thereby enhancing the effectiveness of index options trading strategies. Drawing upon historical data and rigorous empirical analysis, this study demonstrates the feasibility of this approach, achieving an accuracy rate of 63.22%. Beyond improved forecasting accuracy, the system also provides actionable buy or sell recommendations, empowering investors to navigate the options market with greater confidence and precision.**

Keywords—Index options, Trading strategy, Convolutional Neural Networks, Stock Market

I. INTRODUCTION

In tandem with the growing significance of derivatives within financial markets, Futures and Options serve as pivotal tools for investors and traders, enabling them to amplify returns significantly and adeptly manage risk exposure. Within the Indian financial landscape, the National Stock Exchange (NSE) and Bombay Stock Exchange (BSE) stand as the principal exchanges. The annual turnover of futures and options on the NSE during FY 23-24 reached approximately Rs. 95 trillion, underscoring a remarkable surge in transaction volume and monetary involvement. Consequently, this necessitates more robust risk management techniques, prompting the utilization of deep learning-based algorithms to fortify the safety of derivative trading.

The Prediction system implemented in this paper employs a CNN-based architecture trained on two years of historical data, the system predicts price ranges for the next 2, 5, 7, and 10 days, generating "call buy/sell" or "put buy/sell" signals. This will help derivative traders make better informed trading decisions.The dataset used will be described in the next sub-section

A. Dataset Description

The NIFTY 50 index is a portfolio consisting of 50 publicly traded companies that function as a derivative of the underlying stock as mentioned in Mondal et al. 2021 [1]. The dataset employed in this study encompasses historical records of the NIFTY 50 Index from 2018 to 2023 from NSE. It comprises key parameters vital for a thorough analysis of the NIFTY 50 Index, serving as a robust foundation for precise strike price recommendations. The dataset delineates crucial indicators, including

- Close: The closing price of the NIFTY 50 Index at the end of a trading day, a pivotal reference point for market sentiment and pricing trends.
- Open: The opening price of the NIFTY 50 Index at the beginning of the trading day, reflecting early market sentiment.
- High: The highest price reached by the NIFTY 50 Index during a trading day, indicating peak price levels.
- Low: The lowest price reached by the NIFTY 50 Index during a trading day, signifying minimum price levels.
- Volume: The total number of contracts traded during a trading day, a measure of market liquidity and interest.
- FII Call and Put : Track Foreign Institutional Investors' positions in put and call options, providing valuable institutional sentiment on market direction and potential gains.
- Momentum: A measure of the rate of change in the NIFTY 50 Index's price over a specific period, essential for understanding price trends.
- Volatility: A measure of the variability or risk in the NIFTY 50 Index's price, often expressed as the standard deviation, critical for assessing market risk.
- Index Momentum: Similar to individual stock momentum, this metric measures the overall momentum of the entire NIFTY 50 Index.
- FX Price: Foreign exchange rates, reflecting currency values that may influence the NIFTY 50 Index.
- Gold Price: The price of gold, a key safe-haven asset, which can affect investor sentiment and trading strategies.

B. Related Work

Several papers have effectively utilized ensemble models, deep learning architectures, and diverse machine learning methodologies to enhance the precision of stock price predictions.

Long et *al* introduced convolutional neural networks (CNN) and long short-term memory networks (LSTM), which are recognized as deep learning models that are accurate at capturing intricate temporal dependencies, thus enhancing the precision of stock price predictions [2]. Furthermore, Vijh et al harnessed artificial neural networks and random forest methodologies, consequently underscoring the latent potential of ensemble and deep learning models within the context of stock price prediction [3].

As data forms the bedrock for recommendation systems, using the right set of features is of paramount importance to improve accuracy. The impact of Foreign Institutional Investors (FIIs) on stock market volatility is explored by Joo and Mir. Their research employed the augmented Dickey–Fuller test to assess data stationarity and employed the GARCH (Generalised Autoregressive Conditional Heteroskedasticity) model to analyze market volatility. Utilizing time series data, they reveal a substantial correlation between FIIs and market volatility [4].

In [5], the authors Rezaei et al propose a hybrid approach combining empirical mode decomposition (EMD) with Convolutional Neural Networks (CNN) and LSTM. According to the experimental findings, combining CNN with LSTM gives better results than those obtained using other approaches. However, this approach does not incorporate the process of selection.

A short-term prediction model was presented by Sable et al employs evolutionary techniques and the Genetic Algorithm to estimate the prices of eight scripts, each having six variables (Opening Price, Closing Price, Highest Price, Lowest Price, Volume, and Adjusted Closing Price). US-based enterprises are reflected in the eight scripts. Other foreign markets are not covered in this report. It also makes no indication of the precise rationale underlying the selection of the six qualities [6].

The idea of using Gramian angular fields [7] was introduced by Islam and Shuvo in their study, which converted the time-series data into images using Dual Tree Complex Wavelet Transform (DTCWT). Leveraging time series-to-image encoding via Gramian angular fields, Barra et al. employed an ensemble of CNNs to predict S&P 500 index trends. Their multi-resolution approach, outperforming the buy-and-hold strategy, demonstrates the potential of CNNs for financial forecasting even with highly volatile data [8].

Mehtab and Sen used regression and classification algorithms to predict stock values and movements. It highlights the importance of deep learning techniques, showing that CNN-based approaches outperform traditional machine learning models. During analysis they used eight regression and eight classification algorithms to demonstrate numerous stock-value and movement (up/down) prediction approaches on a weekly-forecast prospect. These models were constructed, fine-tuned, and then evaluated using daily historical data from the NIFTY 50 for 5 years. This prediction context is further enhanced by building three CNN models, using univariate and multivariate techniques with varied input data sizes and network formations. This CNN-based approach outperformed machine-learning-based prediction models by a wide margin. The disadvantage is that there are many fluctuations in real-time stock prices [9].

A stock market price trend prediction model was put forth by Shen and Shafiq [10]. The idea is based on deep learning and feature engineering customisation. The proposal relies on many feature engineering techniques with an optimized system, rather than solely relying on a deep learning model. Although the article has a high scientific value, it is still too early to judge how well the same strategy will work in other markets as this was modeled only on the Chinese markets.

II. METHODOLOGY

The implementation works on a Classification-based Convolutional Neural Network (CNN) to predict the range of index prices in terms of returns over n days, relative to the current NIFTY 50 Index price.The classification-based Convolutional Neural Network (CNN) used in this study was inspired by the work of Dixon, Matthew [11]. A primary challenge lies in addressing the sequential nature of index values within a specific time window to enable the model to discern underlying patterns within the sequence. To comprehend the sequential characteristics of NIFTY Index value's closing prices, two distinct system architectures were attempted.

The first architecture utilizes a CNN model to classify index price returns based on closing price. This model was inspired from the work of Haiyao Wang [12]. The second architecture, in addition to the CNN model classification, incorporates external data. This external data encompasses all other parameters in the dataset.

A. Pre-processing data

The important parameters to the proposed model are :

- plot_window: Plot window refers to the time frame between the current day and k training days before today.
- after_n: Refers to the number of training days, after today, to which the returns are calculated
- class_range: Refers to the width of class bands on which the returns can be fit. For example, if class range = 300, then return values from 0 to 300 might lie in one target class and 300 to 600 in another.
- target_extr: Refers to the target classes to which the model can consider data points. Any points outside the target_extr range is considered as an anomaly.

After initiating these parameters, the target classes are changed to a numerical format for the model to understand.

$$target\_class = \begin{cases} floor\left(\frac{return}{class\_range}\right) + 1, if\ return \geq 0 \\ floor\left(\frac{return}{class\_range}\right), \quad otherwise \end{cases}$$

To remove anomaly target classes, it is later filtered to that range as follows:

$$target\_class_{normal} \in \{c \mid c \in target\_class \,\&\, c \geq target\_extr_0 \,\&\, c \leq target\_extr_1$$

After getting the target_class normal (without the anomalies), the target classes are shifted in such a way that the least numerical target is 0. This shift is necessary as when the classification layer trains on the data, it can label only from 0 to a positive number.

$$target\_class_{final}[i] = target\_class_{normal}[i] - min\,(target\_class_{normal})$$

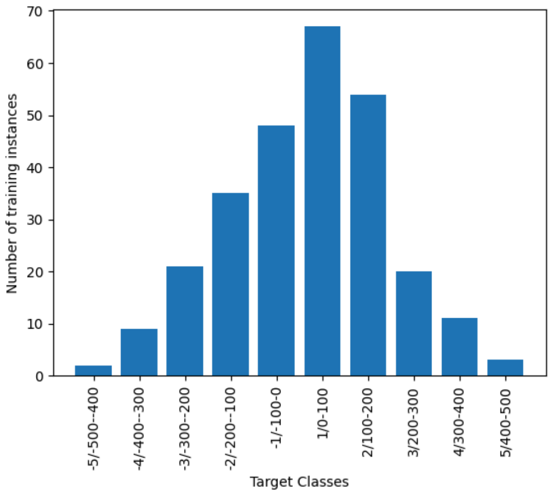

Fig.1 Distribution of training instances w.r.t Target Classes

Provided on the value of plot_window, the data snippet for every date to another date is extracted and the time-series data is converted to GASF-encoded image. The image is created only on the access of a record in a batch, and not stored. Concurrently, the rest of the dataset is formatted to numerical values through one-hot encoding for originally categorical features and other numerical features retained.

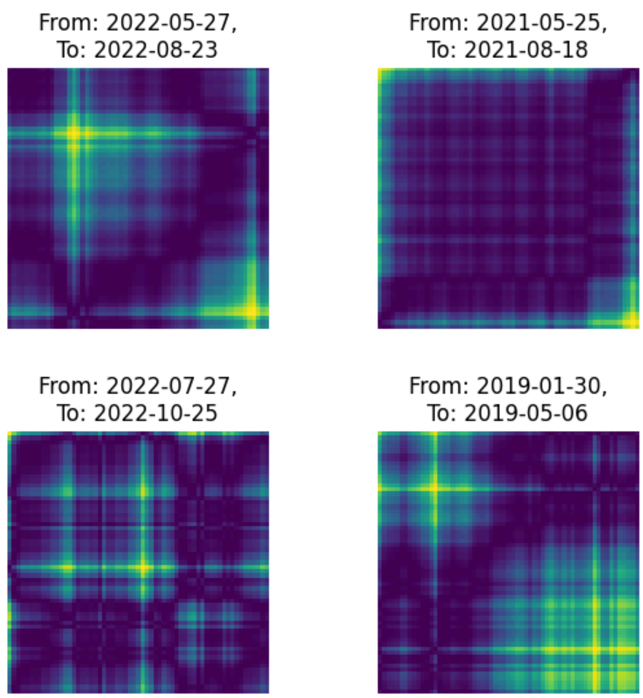

Fig.2 GASF Encoded image representation of NIFTY values from respective time window

B. Architecture and Model Building

The convolutional neural network (CNN) comprises multiple dense layers, max pool 2D operations, and dropout layers, each serving distinct roles in feature extraction and model regularization. The dense layers contribute to the network's capacity to learn complex patterns, while max pool 2D operations downsample spatial dimensions, aiding in the extraction of key features. Dropout layers mitigate overfitting by randomly deactivating a fraction of neurons during training. The CNN's architecture allows for flexibility as its structure can be dynamically adjusted based on specified parameters, facilitating adaptability to diverse input data and task requirements.

Within the framework of the GASF encoded image, the convolutional neural network (CNN) undertakes the crucial task of feature extraction. Progressing through the network layers, the final layer assumes the role of a classifier, generating a label corresponding to the output class. Subsequently, the output class information is decoded to obtain values within the required range, completing the processing pipeline. This hierarchical approach encapsulates the CNN's dual functionality in extracting meaningful features and performing classification, culminating in the derivation of interpretable output values from the encoded representation.

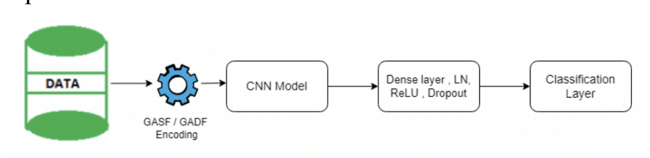

Fig.3 System Architecture 1

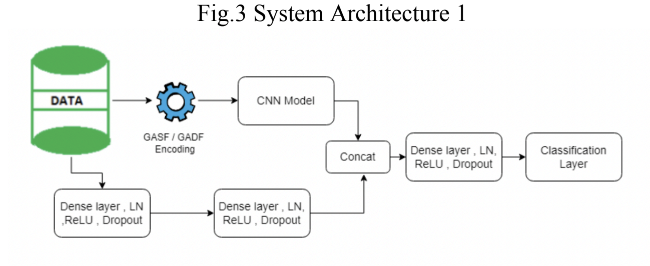

Fig.4 System Architecture 2

III. RESULT

A. Results from Architecture 1

A Convolutional Neural Network (CNN) classifier model is designed for a specific task. The model takes input data with dimensions 1 x plot_window x plot_window, applies convolutional layers with ReLU activations and max-pooling to extract features, and then flattens the output. Subsequently, fully connected layers are employed to perform classification. The forward function processes the input data and computes the logits, providing the model's output.

TABLE 1

ACCURACY VALUES AGAINST VARIOUS INPUT PARAMETERS IN ARCHITECTURE 1

after_n/ plot_window*	**40**	**50**	**60**
2	23.76	16.74	16.43
5	21.07	21.71	18.26

7	18.01	10.90	18.80
10	12.61	16.36	17.88

*Here the *plot_window* refers to the period of Close values the GASF encoding is created on, and *after_n* refers to the N days after which the returns from the current close value is predicted.

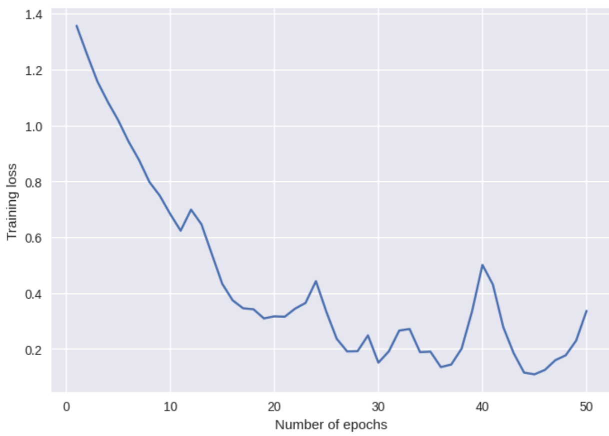

Fig.5 Plot for class_range = 100, plot_window = 40, after_n = 2, epochs = 50

B. Results from Architecture 2

A Convolutional Neural Network (CNN) classifier is implemented for a hybrid model that combines image data with tabular data for classification tasks. The CNN model takes image data, in this case, a 1-channel image with dimensions 40x40, and applies multiple convolutional layers with ReLU activations and dropout to extract features. The tabular model processes tabular data containing selected columns. The outputs of both the CNN and tabular models are concatenated and passed through fully connected layers with activation functions and layer normalization. The final model is designed for a multi-class classification problem with a specified number of classes. The forward function combines image and tabular data, computes the logits, and returns the classification results.

TABLE 2

ACCURACY VALUES AGAINST VARIOUS INPUT PARAMETERS IN ARCHITECTURE 2

after_n/ plot_window*	**40**	**60**	**75**	**100**
2	21.05	31.68	10.80	16.53
5	19.26	15.41	19.06	21.55
7	16.74	17.93	23.18	24.65
10	22.85	24.87	25.61	22.72

*Here the *plot_window* refers to the period of Close values the GASF encoding is created on, and *after_n* refers to the N days after which the returns from the current close value is predicted.

TABLE 3

ACCURACY VALUES FOR DIFFERENT class_range VALUES for plot_window=100, after_n=5*

100	**150**	**200**	**300**
23.56%	36.88%	42.38%	63.22%

*Here, the *class_range* value refers to the value roundoff to the respective class. For example, if the return after 10 days (y) is 545.00 and it is rounded off to class_range=200, then 545 lies in the 400-600 range and thus it is +3.

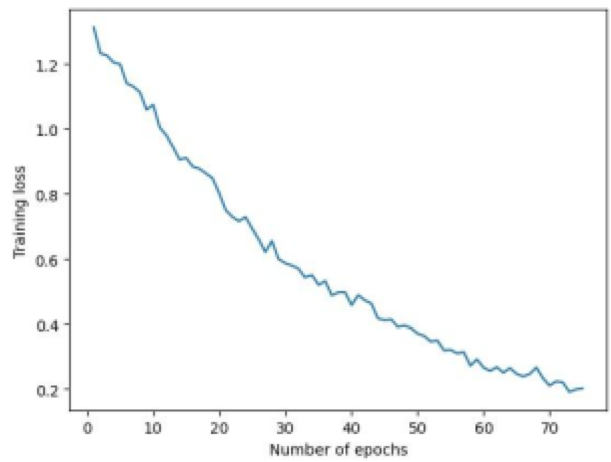

Fig.6 For class_range = 300, plot_window = 100, after_n = 5, epochs = 75

IV. RESULT ANALYSIS

From Table 1 and 2, it can be observed that System Architecture 2 translates to a better overall accuracy, with a maximum of 31.68% accuracy in the plot window of 60 days and returns predicted after 2 days.

In Table 3, it can be inferred that the accuracy percentages increase linearly with the class range values, achieving a maximum of 63.22% in the 300 points range.

In the experimental evaluation, Adam optimizer yielded a better accuracy compared to Adamax and SGD. Hyperparameter tuning revealed that the selection of the optimizer exerted a more pronounced influence on performance outcomes than other parameters. The introduction of weight decay via L2 regularization proved to be an effective strategy, contributing to improved model performance. Furthermore, variations in accuracy were observed with different combinations of learning rates and the number of epochs, underscoring the significance of careful tuning of these parameters for optimal results.

The application of shuffling was instrumental in mitigating the variance in sample mean across batches, thereby contributing to an amelioration of the model's performance by alleviating the risks associated with overfitting.

The utilization of batch normalization was observed to be less effective when compared to layer normalization as used in [13]. The diverse sample mean across batches, even after shuffling, was identified as a limiting factor for batch normalization, whereas layer normalization demonstrated efficacy in normalizing features consistently.
Additionally, in scenarios where neurons across different layers exhibited tendencies to memorize specific values, the integration of dropout mechanisms was introduced as a countermeasure to mitigate overfitting effects. This augmentation proved beneficial in enhancing the generalization capabilities of the model.

V. CONCLUSION

In conclusion, this paper has presented a robust and innovative Strike Price Recommendation system leveraging advanced artificial intelligence, specifically a 3-component CNN model, for forecasting the NIFTY 50 index's prices over the next 2, 5, 7 and 10 trading days. The implemented CNN model has demonstrated remarkable efficacy, aligning closely with actual closing prices and showcasing its potential as a powerful tool in the realm of financial forecasting.

Table 3 highlights the system's commendable performance, achieving a notable 63.22% accuracy in classifying price bands within an optimal 300-point range. The generated trading signals, employing a straightforward parameter to trigger "call buy" or "put sell" for positive percentage changes (> 1%) and "put buy" or "call sell" for negative percentage changes (< 1%), provide valuable insights for decision-making based on support and resistance levels derived from the predicted price range.

It's important to note that the simplicity of the current parameter-based approach is intentional for the purpose of clarity and initial evaluation. However, we recognize the potential for further sophistication in signal generation. Future iterations of this system could explore the incorporation of more intricate parameters or advanced algorithms to refine and enhance the precision of trading signals, adding a layer of complexity that aligns with the evolving landscape of financial forecasting methodologies.

While the current project has successfully met its objectives in predicting NIFTY 50 index prices through CNN models, the journey doesn't end here. The paper acknowledges the vast potential for further exploration and improvement.

Future endeavors could delve into refining the system through the development of specialized deep learning algorithms tailored to the intricacies of financial forecasting. Targeted enhancements in algorithmic architecture, such as exploring Convolutional LSTM (CLSTM) techniques as mentioned in [14]. This could offer improved classification and control, addressing the unique challenges posed by dynamic market conditions. Furthermore, the strategic expansion of datasets, incorporating not only larger volumes but also diverse sources, has the potential to significantly augment precision and boost data generalization. This multifaceted approach holds promise for ushering in a new era of even more accurate predictions, firmly establishing the system as a cutting-edge tool in the landscape of financial decision support.

In essence, this study not only underscores the significant strides made in utilizing AI for financial forecasting but also sets the stage for promising future applications. The positive outcomes observed in this research contribute to the growing body of knowledge on the intersection of artificial intelligence and the financial sector, encouraging continued exploration and innovation in this dynamic field. The potential for further advancements highlights the transformative impact that AI can have on decision-making processes within financial markets.

REFERENCES

[1] Mondal, Bhaskar, Om Patra, Ashutosh Satapathy, and Soumya Ranjan Behera. 2021. A Comparative Study on Financial Market Forecasting Using AI: A Case Study on NIFTY. In Emerging Technologies in Data Mining and Information Security. Singapore: Springer, vol. 1286, pp. 95–103.

[2] Long, Wen, Zhichen Lu and Lingxiao Cui. (2019). Deep learning-based feature engineering for stock price movement prediction. Knowledge-Based Systems 164: 163–73.

[3] Vijh, Mehar, Deeksha Chandola, Vinay Anand Tikkiwal, and Arun Kumar. (2020). Stock closing price prediction using machine learning techniques. Procedia Computer Science 167: 599–606.

[4] Bashir, Ahmad & Joo, Bashir & Mir, Zahoor. (2015). Impact of FIIs Investment on Volatility of Indian Stock Market: An Empirical Investigation. 1. 2375-774.

[5] Rezaei, Hadi, Hamidreza Faaljou, and Gholamreza Mansourfar. (2021). Stock price prediction using deep learning and frequency decomposition. Expert Systems with Applications 169: 114332.

[6] Sable, Sonal, Ankita Porwal, and Upendra Singh. 2017. Stock price prediction using genetic algorithms and evolution strategies. Paper presented at the International conference of Electronics, Communication and Aerospace Technology (ICECA) Proceedings, Coimbatore, India, April 20–22; pp. 240–53.

[7] Islam, Md Monirul & Shuvo, Md. Maruf Hossain. (2019). DenseNet Based Speech Imagery EEG Signal Classification using Gramian Angular Field. 10.1109/ICAEE48663.2019.8975572.

[8] Barra, Silvio & Carta, Salvatore & Corriga, Andrea & Podda, Alessandro & Reforgiato Recupero, Diego. (2020). Deep Learning and Time Series-to-Image Encoding for Financial Forecasting. IEEE/CAA Journal of Automatica Sinica. 7. 683-692. 10.1109/JAS.2020.1003132.

[9] Mehtab, Sidra, and Jaydip Sen. (2020). Stock price prediction using convolutional neural networks on a multivariate time series. arXiv arXiv:2001.09769.

[10] Shen, Jingyi, and M. Omair Shafiq. 2020. Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data 7: 1–33.

[11] Dixon, M., Klabjan, D., & Bang, J. H. (2017). Classification-based financial markets prediction using deep neural networks. Algorithmic Finance, 6(3-4), 67-77.

[12] Wang, H., Wang, J., Cao, L., Li, Y., Sun, Q., & Wang, J. (2021). A Stock Closing Price Prediction Model Based on CNN-BiSLSTM.

[13] J. L. Ba, J.R.Kiros, and G. E. Hinton, "Layer Normalization." arXiv, 2016. doi: 10.48550/ ARXIV.1607.06450.

[14] X. Shi, Z. Chen, H. Wang, D. Yeung, W. C. Wong, and W.-C. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," arXiv (Cornell University), Jun. 2015, doi: 10.48550/arxiv.1506.04214.