Comparative Analysis of CNN Pre-trained Model for Stock Market Trend Prediction

Jitendra Kumar Chauhan[1,2](✉), Tanveer Ahmed[1], and Amit Sinha[2]

[1] Computer Science Department, Bennett University, Plot Nos 8, 11, TechZone 2, Greater Noida 201310, Uttarpradesh, India
{E19SOE817,tanveer.ahmed}@bennett.edu.in

[2] Department of Information Technology, ABES Engineering College, Ghaziabad, India
amit.sinha@abes.ac.in

Abstract. This research offers an in-depth comparative analysis of various pre-trained Convolutional Neural Network (CNN) models such as VGG16, ResNet50, InceptionV3, MobileNetV2, and Xception to predict stock market trends. Our approach involves the conversion of time-series financial data into 2D image-like structures through the application of two distinct techniques: the Gramian Angular Field (GAF) and the Markov Transition Field (MTF). By applying this transformation, we leverage the power of CNNs. We utilize the ideas of transfer learning and try to evaluate the performance of each model using several measures including predictive accuracy, precision, recall, F1-score, and computational efficiency. The analysis highlights the unique advantages and limitations of each model, thereby offering valuable insights into their suitability for stock market prediction tasks. This study is a significant contribution to the current body of literature on financial time series forecasting, providing a novel perspective on using pre-trained CNN models in the Indian Financial Sector. It carries important implications for future work and practitioners in the finance and investment sectors, offering a tool for more e-market predictions.

Keywords: Stock trend classification Deep learning · Trading · Convolutional neural network GAF · MTF

1 Introduction

Stock price prediction is at the heart of the financial world and the broader global economy. Investors, traders, economists, and policymakers closely monitor the fluctuation of stock prices to make informed decisions. Understanding where inventory costs are headed can cause quite a few rewarding opportunities, consisting of strategic funding, hazard management, and macroeconomic making plans. The economic markets are complicated and multifaceted structures

KC Santosh et al. (Eds.): RTIP2R 2023, CCIS 2026, pp. 110–129, 2024.
https://doi.org/10.1007/978-3-031-53082-1_10

where countless variables have interaction. Global occasions, governmental rules, company decisions, economic signs, mental elements, and climate situations can have an effect on inventory prices. This elaborate interplay of things contributes to the non-linear and non-desk-bound conduct of economic time series, making inventory charge prediction one of the most difficult tasks in finance. Stock fee prediction isn't always a brand-new phenomenon. It dates returned to the early days of inventory exchanges when investors trusted rudimentary evaluation, intuition, and revel in. Over the decades, statistical models, econometric evaluation, technical signs, and quantitative strategies have evolved to research historic charge facts and identify potential trends. Despite advances in mathematics, statistics, and computing, accurate stock price prediction stays elusive. Factors contributing to the mission encompass:

1. Market Noise: Financial markets are rife with noise and extraneous facts that could confound evaluation.
2. Volatility: Markets are regularly having difficulty with surprising and unpredictable adjustments.
3. Emotional Factors: Investor psychology and sentiment play a big role in marketplace actions.
4. Regulatory Changes: Unexpected modifications in governmental or regulatory policies can dramatically affect stock fees.
5. Global Events: Unforeseen international occasions which include herbal disasters, pandemics, or geopolitical tensions can result in abrupt market shifts.

With the advent of computer technology, more sophisticated patterns emerged of Simple linear regression including complex machine learning algorithms, computational methods are irrelevant in the world of finance. However, traditional methods often fail to capture the complex non-linear dynamics of the market. Researchers have been seeking ways to incorporate the non-linear and non-desk bound behavior of monetary time series, yet the problem remains a persistent challenge (Abhyankar et al.,1997) [1]; (Hartman and Hlinka, 2018) [2]. The persistent difficulty in predicting stock prices underscores the need for innovative approaches. There is a continuous quest for models that can better interpret the complexities of financial data, learn from historical trends, and predict future price movements with greater accuracy. Stock price prediction remains a critical yet enigmatic component of financial analysis. The potential rewards are substantial, but so are the risks and challenges.

In this paper, we harness the capabilities of deep learning, notably, Convolutional Neural Networks (CNNs), to forecast stock prices. Focusing on 20 equities from the Indian stock exchange, our methodology employs multiple pre-trained CNN architectures such as VGG16, ResNet50, and InceptionV3 [Simonyan and Zisserman, 2014; He et al., 2015 [3,4]; Szegedy et al., 2016] [5]. Recognizing the inherent temporal sequences in stock prices, we convert a stock's price into two-dimensional imagery via Markov Transition Fields (MTF) and Gramian

Angular Fields (GAF) [Wang and Oates, 2015] [6]. These techniques are adept at spatially encoding temporal patterns, enhancing the CNNs' analytical performance. The resultant 2D images are processed through the pre-trained CNN models. Through extensive numerical examination, spanning over five years of stock data, we have found that the method shows good results in terms of prediction accuracy. This is considering the case of MTF and GAF simultaneously. Leveraging these pre-trained CNN models, we not only demonstrate their potential in real-world stock market scenarios but also highlight their scalability benefits in demanding computational contexts. The rest of the paper is organized as follows:

2 Literature Review

The landscape of economic market forecasting keeps evolving, with various research papers investigating various theories, strategies, and technology. Understanding financial market forecasting begins with the basic tasks of time series analysis and trading regulation. Box and Jenkins (1970) [7] proposed a paradigm shift by introducing the ARIMA model, an important technique in time series forecasting. However, this model operates under the assumptions of linearity and stationarity - conditions that data on financial markets often violate. Subsequently, Nelson (1991) [8] and Taylor (1986) [9] proposed ARCH and GARCH models, respectively, to deal with the inherent volatility commonly found in economic time series data. To bridge the gap between theory and practice, Brock, Lakonishok, and LeBaron (1992) [10] empirically tested the utility of simple industrial trading rules, a cornerstone of financial trading and they emphasized.

Turning to the realm of data transformation techniques, various methods have been proposed to align time series data with advanced machine learning techniques Wang and Oates (2015) [11] use Gramian Angular Field (GAF) - a new technique for time series transformation data for convolutional neural networks (CNNs). A Markov Transition Field (MTF) is introduced, which is a new technique for transforming time series data into a 2D matrix. Guo et al. (2016) [12] engaged visibility graph networks, developing a different approach for visualizing time series data, while Xie, Xu, and Wang (2016) [13] and Lahmiri and Boukadoum (2019) [14] investigated wavelet transform and empirical mode decomposition (EMD) as preprocessing techniques for economic time series data, respectively.

Building a solid foundation of theoretical and variable methodologies, recent research has used pre-trained CNN models to forecast various applications in the financial domain. Sezer and Ozbayoglu (2018) [15] showed the potential of deep learning techniques, especially CNN, in finance for tasks such as credit card fraud detection. In another application, Zhang, Zhou, and Yang (2019) [16] use CNN to predict stock price movements based on financial media, which provided a new perspective on the use of inputs to predict stock prices and Ding, Zhang, Liu, and Duan (2015) [11] demonstrated the forecasting capability of CNN based on textual financial reports. Patel, Shah, Thakkar, and Kotecha (2015)[17] used pre-trained CNN models to evaluate the effectiveness of transfer learning for predicting financial market dynamics. In addition, Li, Li, and Li (2020) [18] demonstrated the capability of transfer learning using previously trained models to predict stock market movements using large-scale classification image models to be used for financial time series forecasting.

To tie the whole thing together, several researchers have conducted comparative analyses and meta-analyses to assess the overall performance of various forecasting techniques. Tsantekidis et al. (2017) [19] supplied a comparative take look at machine studying techniques, including CNNs, for inventory rate prediction. Bao, Yue, and Rao (2017) [20]and Hoseinzade and Haratizadeh (2019) [21] proposed hybrid models that integrate distinct device-gaining knowledge of techniques, providing strong and flexible strategies for inventory prediction. Makridakis, Spiliotis, and Assimakopoulos (2018) [22] provided a complete comparison of statistical and system studying techniques in time series forecasting, providing insights into the strengths and weaknesses of those extraordinary approaches. Lastly, Gu, Kelly, and Xiu (2020) [23] tested a whole lot of gadget learning models for their capability in forecasting stock returns, evaluating their performance based totally on a big dataset of inventory marketplace records.

3 Proposed Methodology

This section expands upon the details of the proposed approach. The overall methodology has been divided into four main parts. They are as follows:

1. The first subsection describes the model architecture in detail.
2. In the second subsection, we discuss the CNN architecture in detail.
3. In the third section a discussion on labeling of stock's close-price.
4. In the last section describe the image generation.

3.1 Model Architecture

In the first step, we download the historical data for 20 distinct shares from Yahoo Finance. These statistics consist of daily data like open price, close fee, excessive, low, quantity, etc., for a selected duration. These prices serve as the basis for our next analysis and version education. Once we have the prices, we rework them into images using distinct techniques: Gramian Angular Fields (GAF) and Markov Transition Fields (MTF). These techniques encode time-collection information into images that could then be processed via convolutional neural networks. GAF captures the temporal correlations among one-of-a-kind time steps, whilst MTF illustrates the probabilistic transitions between unique states inside the time-collection information. We use five different pre-trained Convolutional Neural Network (CNN) models for our purpose: VGG16, ResNet50, Inception, MobileNetv2, and Xception. Each model offers a prediction based totally on the input images. This prediction step is executed one at a time for the images generated via each of the GAF and MTF techniques. To get our very last prediction, we compute a weighted common of the predictions from all five models for each of the 20 shares. This method permits us to leverage the strengths of every model and minimizes the influence of any single version's weaknesses. By doing this separately for the GAF-generated images and the MTF-generated snapshots, we are able to evaluate the performance of the two picture encoding strategies within the context of our particular undertaking (Fig. 1).

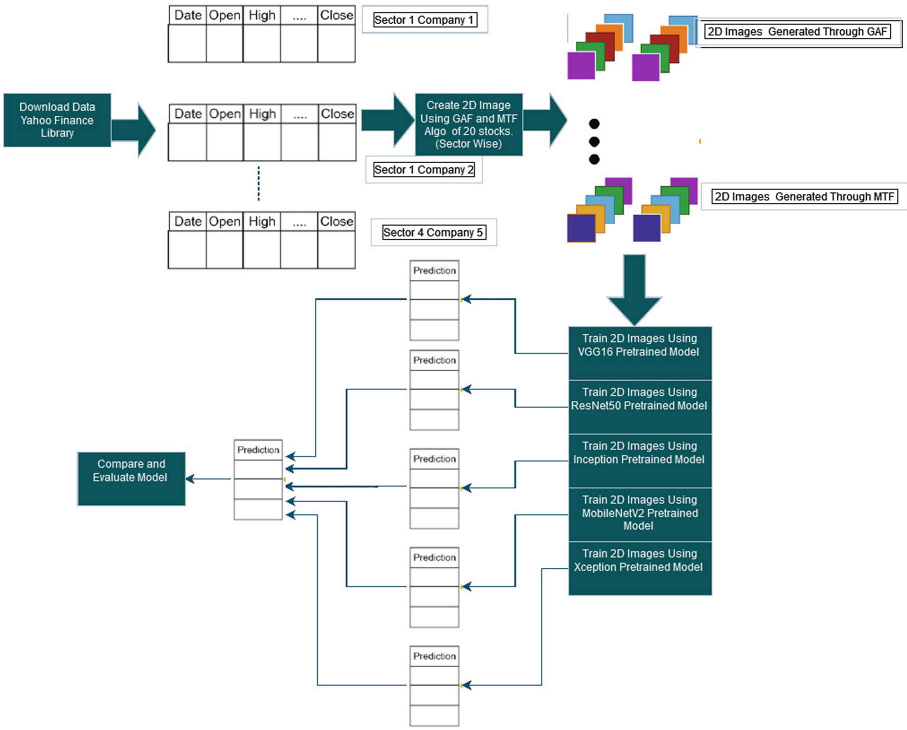

Fig. 1. Proposed framework for Stock trend classification

3.2 CNN Architecture

Convolutional Neural Networks, commonly referred to as CNNs, are a breakthrough technology in deep learning, particularly geared toward visual recognition tasks. These models have played a pivotal role in achieving state-of-the-art results in areas such as image classification, object detection, and facial recognition. The beauty of CNNs lies in their ability to automatically and adaptively learn spatial hierarchies of features from the input data. The convolution operation involves sliding a kernel across the input data and performing element-wise multiplication followed by summation. Each kernel is tuned to recognize specific characteristics, such as edges, corners, textures, or more complex patterns in higher layers. The output of this operation is known as a feature map or activation map. Strides and padding are critical concepts in convolutional layers. A stride determines how many steps a kernel takes when sliding across the input. Padding, on the other hand, adds extra pixels around the input to control the spatial dimensions of the output feature maps. By tuning these parameters, one can achieve a desirable level of control over the layer's operation. Pooling layers play a vital role in down-sampling or reducing the spatial dimensions of the feature maps. By doing so, they make the model more robust to variations and reduce the computational complexity. Max Pooling selects the maximum value from a group of values within a window, while Average Pooling takes the average. Both methods are effective in preserving essential features while discarding redundant information. The choice between Max and Average Pooling depends on the specific requirements of the task. Fully connected layers act as the final stages of a CNN. They combine the features extracted from previous layers into a more abstract and high-level understanding of the input data. The final fully connected layer often uses a softmax activation function, especially for multi-class classification tasks. The softmax function converts the raw output into probabilities, providing a clear and interpretable prediction for each class.

Various designs of CNNs were proposed over the years, each introducing new thoughts and techniques for enhancing performance. Choosing the right pre-trained version for a specific challenge is hard. For the challenge of stock market trend prediction, the following models are used in this article: VGG 16, ResNet50, InceptionV3, MobileNetV2, Xception.

3.3 Labeling of Stock's Close-Price

The methodology for labeling stock trends is central to our proposed research, as it lays the foundation for the predictive capabilities of our models. This method,

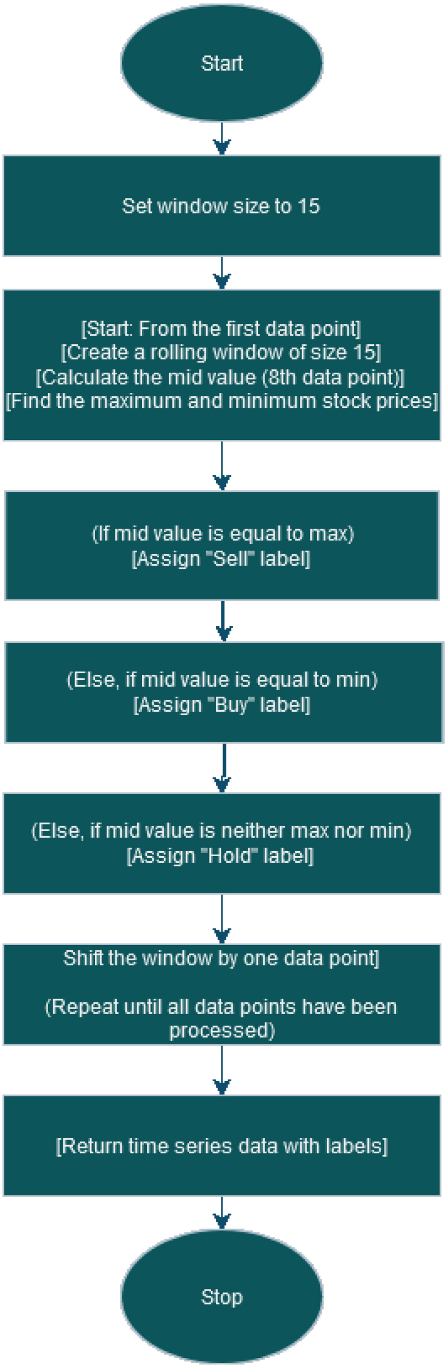

Fig. 2. Stock Label Algorithm

though simple in theory, proves to be highly effective in practice. It employs a window-based algorithm, considering a specific number of data points to assign labels to stock prices. The procedure to accomplish the goal is highlighted in the following text.

Algorithm Description:

1. Window Size Selection: Initially, the window size is set to encompass 15 data points. This size is not rigid and can be adjusted to meet the specific needs of any analysis.
2. Rolling Window Creation: For each stock price dataset, we commence with the first data point and proceed by creating a rolling window of the set size.
3. Midpoint Calculation: Within each window, we identify the midpoint, or the eighth data point in this case.
4. Maximum and Minimum Identification: The maximum and minimum stock prices within the window are identified, and the labeling of the midpoint is executed based on these values.
   a. Sell Label: If the midpoint equals the maximum price, it is labeled as "Sell," indicating that the stock price has reached a peak and may represent an opportune selling moment.
   b. Buy Label: Conversely, if the midpoint corresponds to the minimum price, it is labeled as "Buy," signifying a low in stock price and possibly a favorable buying opportunity.
   c. Hold Label: If the midpoint is neither the maximum nor minimum price, the label "Hold" is assigned, suggesting no significant high or low in stock price and implying a recommendation to defer transactions.
5. Rolling Window Progression: The window is then shifted forward by one data point, and the process is repeated until all data points have been processed.

The final product is a time series of stock prices with corresponding "Buy," "Sell," or "Hold" labels. These labels are primed for further analysis and model training. This methodology, by converting raw stock prices into actionable labels, contributes to the decision-making process in investment strategies. The flexibility in window size and the structured approach to labeling provide a robust tool for financial analysts and researchers. Further exploration may include optimizing the window size based on specific stock behaviors and integrating additional variables into the algorithm to enhance predictive accuracy.

3.4 Image Generation

In this article, we focused our efforts on forecasting price trends. To do this, we converted stock prices into pictures. In this case, we used Gramian angular fields (GAF) and Markov transition fields (MTF). Both methods put time-collection data into picks in a way that gives a time cost based on the total pattern embedded in the series and makes those patterns more visible to algorithms and detection devices

3.4.1 Grammian Angular Field(GAF)

The Gramian angular fields (GAF) approach is an effective means of transforming time-series data into images, which can be employed for machine learning examination. Here is a precise explanation of the steps involved.

1. **Regularization:** Here, we must regularize the time series data $X = [x_1, x_2, \ldots, x_P]$, such that it falls between -1 and 1. This can be achieved by conducting the min-max regularization technique:

$$x'_i = 2\left(\frac{x_i - \min(X)}{\max(X) - \min(X)}\right) - 1.$$

   This regularization step is paramount as the GAF method relies on arccosine and cosine functions, which necessitate inputs that are within a given boundary.
2. **Polar Coordinate Transformation:** After normalization, each element of x'_i in the temporal dataset is transformed to a polar coordinate. The angle ϕ is then calculated from this data point $\phi_i = \arccos(x'_i)$, and the radius r is simply the moment in time at which the datum is recorded.
3. **Gramian Angular Field:**Next, we evaluate the Gramian Angular Field G, a $P \times P$ matrix that chronicles the temporal links of the sequence. This can be either a Gramian Angular Summation Field (GASF) or a Gramian Angular Difference Field (GADF). The elements of the GASF are calculated using the formula:

$$GASF_{i,j} = \cos(\phi_i + \phi_j),$$

   whereas the GADF is calculated as:

$$GADF_{i,j} = \sin(|\phi_i - \phi_j|).$$

4. **Image Representation:** The GAF is visualized as an image, where the color intensity of each pixel corresponds to the value in the GAF. The resulting image retains the complete temporal correlation information from the original time series.

These images, generated from the GAF transformation, can be input into image-based machine learning models, like convolutional neural networks (CNNs), for further analysis. A significant advantage of GAF is its ability to maintain the temporal relationship of the time series, enabling CNNs to recognize sequential patterns within the data, potentially improving predictive performance (Fig. 3).

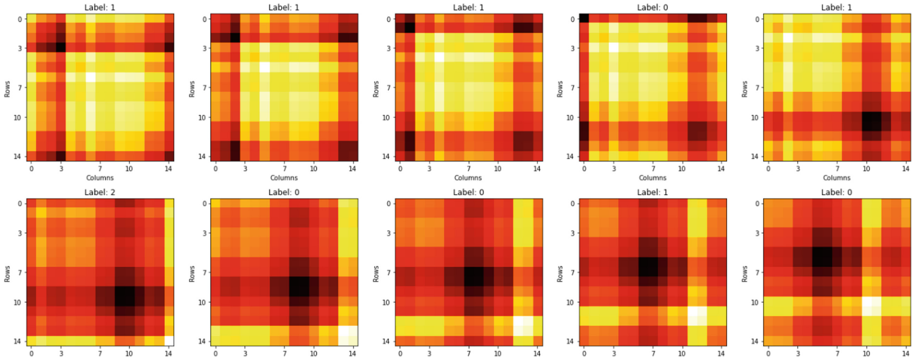

Fig. 3. Stock GAF Images

3.4.2 Markov Transition Field(MTF)

The Markov Transition Field (MTF) is another innovative method for transforming time-series data into an image for use in machine learning analysis. The MTF method specifically captures the temporal dependency in time-series data. The steps involved in the MTF method are as follows:

1. **Discretization:** The first step in this MTF method is to discretize the normalized time series data $X = [x_1, x_2, \ldots, x_P]$. This is basically done using quantile bins, where the bin edges are determined by the quantiles of the data distribution. As a result, each x_i in the series is replaced by its corresponding quantile bin.
2. **Transition Matrix:** After discretization, we calculate the transition matrix T, which is a $P \times P$ matrix where each entry $T_{i,j}$ represents the probability of transitioning from state i to state j in the time series. This transition matrix captures the dynamics of the time series.
3. **Markov Transition Field:** Then, we compute the Markov Transition Field (MTF), which is also a $P \times P$ matrix. Each entry $MTF_{i,j}$ in the MTF is calculated as the probability of transitioning from the state at time i to the state at time j, which is given by T_{x_i,x_j}.
4. **Image Representation:** The MTF matrix can then be visualized as an image, where each pixel's intensity corresponds to the value of the MTF matrix. The resulting image retains the dynamic transitions of the original time series.

The images generated through the MTF transformation can be used as inputs for image-based machine learning models, such as convolutional neural networks (CNNs). A key advantage of MTF over other time-series-to-image transformation methods is its capability to maintain and emphasize the Markov property of the time series, which could potentially improve the performance of time-series prediction or classification tasks (Fig. 4).

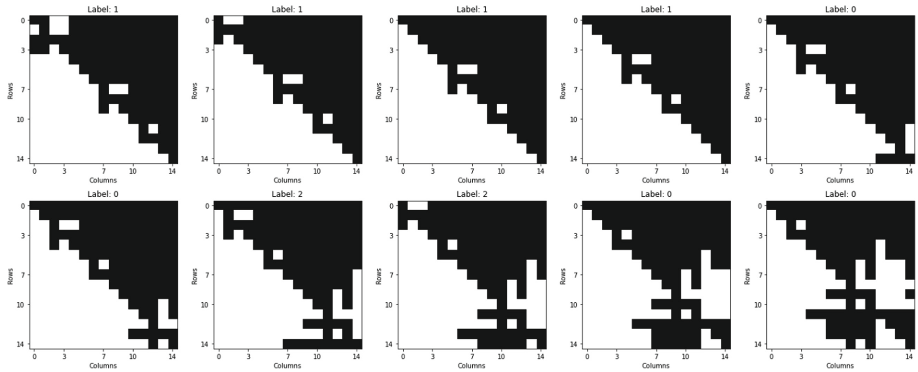

Fig. 4. Stock MTF Images

4 Results

4.1 Dataset and Evaluation Criterion

The dataset used in this study consists of the daily stock prices for the following Indian companies over the past five years: COAL INDIA, CIPLA, BRITTANIA, BPCL, BHARTI AIRTEL, BAJAJ AUTO, AXISBANK, ASIAN PAINTS, ADANI PORTS, ONGC, MARUTI, INFY, ICICIBANK, HINDALCO, HEROMOTOR, HDFCBANK, HCLTECH, GAIL, EICHERMOTOR, DRREDDY. This data has been retrieved from Yahoo Finance, which is a reliable source of historical stock price information.

Each data record in our dataset includes the date, opening price, closing price, highest price of the day, lowest price of the day, the volume of shares traded that day, and the adjusted closing price. The information of the close price is used to generate the image-like structures through the Gramian Angular Fields (GAF) and Markov Transition Fields (MTF) transformations. The labels for the data, which represent the trends we are trying to predict, are generated by a window-based labeling technique shown in Fig. 2. For each window of 15 d, the trend is classified as "Buy" if the mid-window price is the minimum within the window, "Sell" if the mid-window price is the maximum within the window, and "Hold" otherwise.

As for the evaluation criteria, the performance of the CNN models (VGG16, ResNet50, InceptionV3, MobileNetV2, and Xception) on the stock market trend prediction task will be measured using accuracy, precision, recall, and F1 score:

1. Accuracy: This is the proportion of total predictions that are completely correct. It's calculated as

$$(\text{True Positives} + \text{True Negatives})/\text{Total Predictions}$$

.

2. Precision: Also called Positive Predictive Value, this is the proportion of positive predictions that are actually correct. It's calculated as Precision

$$\text{True Positives}/(\text{True Positives} + \text{False Positives})$$

3. Recall: Also known as Sensitivity, Hit Rate, or True Positive Rate, this is the proportion of actual positives that are correctly identified. It's calculated as

$$\text{True Positives}/(\text{True Positives} + \text{False Negatives})$$

4. F1 Score: The F1 Score is the harmonic mean of Precision and Recall, and it tries to balance the two in a single number. It's calculated as

$$2 \ast (\text{ Precision } * \text{ Recall})/(\text{Precision} + \text{Recall })$$

.

4.2 Experimentation with Different Pretrained Model

The overall performance comparison of different methods across a diverse variety of stocks is an essential exercise to decide the best model for stock market trend prediction. For each version, we file the accuracy, precision, recall, and F1 rating across 20 stocks, together with COAL INDIA, CIPLA, BRITTANIA, BPCL, BHARTI AIRTEL, BAJAJ AUTO, AXISBANK, ASIAN PAINTS, ADANI PORTS, ONGC, MARUTI, INFY, ICICIBANK, HINDALCO, HEROMOTOR, HDFCBANK, HCLTECH, GAIL, EICHERMOTOR, DRREDDY.

4.2.1 GAF Images Results with Different Pretrained Model

Table 1. Results from AsianPaint Stocks Images through GAF

Image Creation	Pretrained Model	Label	Precision	Recall	F1-Score	Accuracy
GAF	VGG16	Hold	0.44	0.42	0.43	0.55
		Buy	0.6	0.85	0.71	–
		Sell	0.47	0.14	0.21	–
	ResNet50	Hold	0.28	0.28	0.28	0.47
		Buy	0.59	0.61	0.6	–
		Sell	0.43	0.41	0.42	–
	InceptionV3	Hold	0.35	0.48	0.4	0.51
		Buy	0.61	0.73	0.66	–
		Sell	0.6	0.14	0.22	–
	MobileNetV2	Hold	0.41	0.22	0.3	0.54
		Buy	0.55	0.97	0.7	–
		Sell	0.6	0.09	0.16	–
	Xception	Hold	0.31	0.07	0.12	0.54
		Buy	0.57	0.93	0.7	–
		Sell	0.48	0.3	0.37	–

Table 2. Results from Maruti Stocks Images through GAF

Image Creation	Pretrained Model	Label	Precision	Recall	F1-Score	Accuracy
GAF	VGG16	Hold	0.36	0.56	0.45	0.47
		Buy	0.6	0.5	0.55	–
		Sell	0.44	0.33	0.38	–
	ResNet50	Hold	0.5	0.21	0.29	0.52
		Buy	0.54	0.88	0.67	–
		Sell	0.43	0.19	0.26	–
	InceptionV3	Hold	0.52	0.23	0.32	0.55
		Buy	0.56	0.95	0.71	–
		Sell	0.46	0.12	0.2	–
	MobileNetV2	Hold	0.36	0.33	0.35	0.52
		Buy	0.57	0.83	0.68	–
		Sell	0.46	0.12	0.2	–
	Xception	Hold	0.36	0.42	0.38	0.5
		Buy	0.64	0.58	0.61	–
		Sell	0.44	0.44	0.44	–

Table 3. Results from Average of 20 Stocks Images through GAF

Image Creation	Pretrained Model	Label	Precision	Recall	F1-Score	Accuracy
GAF	VGG16	Hold	0.45	0.33	0.36	0.54
		Buy	0.61	0.78	0.67	–
		Sell	0.51	0.3	0.33	–
	ResNet50	Hold	0.44	0.31	0.3	0.49
		Buy	0.58	0.69	0.61	–
		Sell	0.35	0.3	0.29	–
	InceptionV3	Hold	0.43	0.81	0.3	0.5
		Buy	0.57	0.75	0.63	–
		Sell	0.47	0.3	0.32	–
	MobileNetV2	Hold	0.47	0.24	0.28	0.52
		Buy	0.58	0.81	0.66	–
		Sell	0.48	0.29	0.33	–
	Xception	Hold	0.43	0.32	0.34	0.52
		Buy	0.58	0.76	0.65	–
		Sell	0.48	0.29	0.34	–

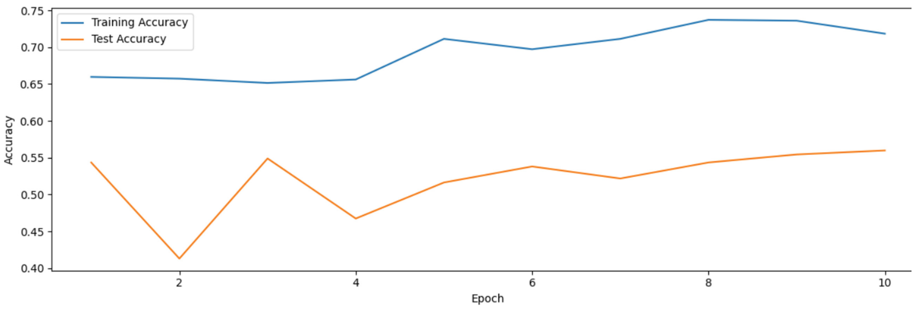

Fig. 5. GAF Maruti Accuracy vs Epoch

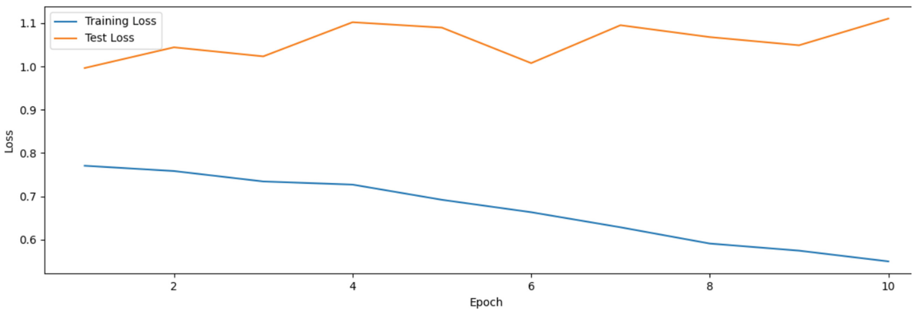

Fig. 6. GAF Maruti Loss vs Epoch

In our study, we investigated the performance of pre-trained AsianPaint stock images processed through GAF. The analysis, as shown in Table 1, reveals remarkable variability in precision, recall, F1-score, and accuracy on different characters. In particular, the VGG16 model had an F1-Score of 0.71 for the Buy line but only 0.21 for the Sell line. In comparison, the MobileNetV2 model achieved the best balance with F1 scores of 0.7 and 0.16 for the Buy and Sell lists, respectively. The data also show significant differences in performance between the models, with Xception showing a surprisingly low F1-Score for Hold labels at 0.12. Through systematic analysis, it is clear that obtaining a well-balanced display of Hold, Buy, and Sell labels is a complex task, and it is important to choose an appropriate pre-trained model In another study, we investigated the application of pre-trained images on Maruti Stocks Images, using GAF for visualization. As described in Table 2, there are some interesting patterns to note. For example, the InceptionV3 model exhibited an impressive F1-Score of 0.71 for Buy labels, while the VGG16 model balanced all three classes ResNet50, although it showed promising performance with a 0.67 F1-Score for Buy labels, its inaccuracies and remember Hold and Sell lines are. Furthermore, Figs. 3 and 4 show the accuracy and loss and epoch graphs for GAF Maruti and MobileNetV2, showing the complex correlation between training achievement and final performance. Finally, we performed a detailed analysis of 20 stock figures using the average GAF and previously trained models. Table 3 summarizes these findings, showing consistency in some areas, and variation in others. For example, the VGG16 instance was able to produce an F1-Score of 0.67 for the Buy label, while Xception had a balanced performance with F1-Scores of 0.65 and 0.34 for the Buy and Sell labels, respectively MobileNetV2 regular performance with almost equal results for all Buy and Sell labels, while F1-Scores were 0.66 and Interestingly, InceptionV3 had an impressive recall of 0.81 for the Hold label but a low F1-Score of 0.3. This detailed analysis highlights the complexity of model selection, emphasizing that optimizing performance requires a fine-grained and thorough understanding of specific dataset characteristics In summary, this study contributes important insights into how the models perform on stock datasets using GAF. They emphasize the importance of careful model selection, tuning, and possibly, fusion in order to obtain the best results across different metrics. Furthermore, these findings can lay the foundation for future studies of robust sampling, transferability, and real-world applications to the complexity of banking research (Fig. 5, 6 and Table 4).

4.2.2 MTF Images Results with Different Pretrained Model

Table 4. Results from Asian Paint Stocks Images through MTF

Image Creation	Pretrained Model	Label	Precision	Recall	F1-Score	Accuracy
MTF	VGG16	Hold	0.72	0.36	0.48	0.66
		Buy	0.67	0.86	0.76	–
		Sell	0.6	0.6	0.6	–
	ResNet50	Hold	0.79	0.22	0.34	0.58
		Buy	0.54	1	0.7	–
		Sell	1	0.16	0.27	–
	InceptionV3	Hold	0.6	0.62	0.61	0.63
		Buy	0.66	0.89	0.75	–
		Sell	0.5	0.13	0.21	–
	MobileNetV2	Hold	0.58	0.62	0.6	0.66
		Buy	0.73	0.74	0.73	–
		Sell	0.59	0.53	0.56	–
	Xception	Hold	0.49	0.34	0.4	0.57
		Buy	0.59	0.8	0.68	–
		Sell	0.6	0.4	0.48	–

Table 5. Results from Maruti Images through MTF

Image Creation	Pretrained Model	Label	Precision	Recall	F1-Score	Accuracy
MTF	VGG16	Hold	0.44	0.58	0.5	0.59
		Buy	0.68	0.69	0.69	–
		Sell	0.63	0.4	0.49	–
	ResNet50	Hold	0.53	0.48	0.51	0.55
		Buy	0.56	0.83	0.67	–
		Sell	0.5	0.1	0.17	–
	InceptionV3	Hold	0.61	0.46	0.52	0.57
		Buy	0.61	0.62	0.62	–
		Sell	0.48	0.58	0.53	–
	MobileNetV2	Hold	0.51	0.44	0.47	0.57
		Buy	0.62	0.68	0.65	–
		Sell	0.49	0.48	0.48	–
	Xception	Hold	0.77	0.5	0.61	0.63
		Buy	0.58	0.91	0.7	–
		Sell	0.86	0.25	0.39	–

Table 6. Results from Average of 20 Stocks Images through MTF

Image Creation	Pretrained Model	Label	Precision	Recall	F1-Score	Accuracy
MTF	VGG16	Hold	0.57	0.47	0.49	0.61
		Buy	0.64	0.73	0.68	–
		Sell	0.62	0.52	0.55	–
	ResNet50	Hold	0.64	0.36	0.42	0.59
		Buy	0.6	0.8	0.68	–
		Sell	0.63	0.42	0.47	–
	InceptionV3	Hold	0.55	0.49	0.5	0.55
		Buy	0.61	0.66	0.62	–
		Sell	0.5	0.43	0.44	–
	MobileNetV2	Hold	0.56	0.41	0.46	0.58
		Buy	0.65	0.76	0.66	–
		Sell	0.63	0.41	0.46	–
	Xception	Hold	0.56	0.42	0.46	0.57
		Buy	0.61	0.73	0.66	–
		Sell	0.57	0.44	0.47	–

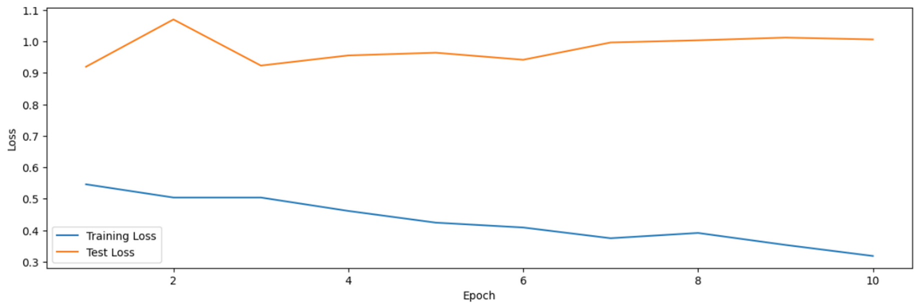

Fig. 7. MTF Maruti Accuracy vs Epoch

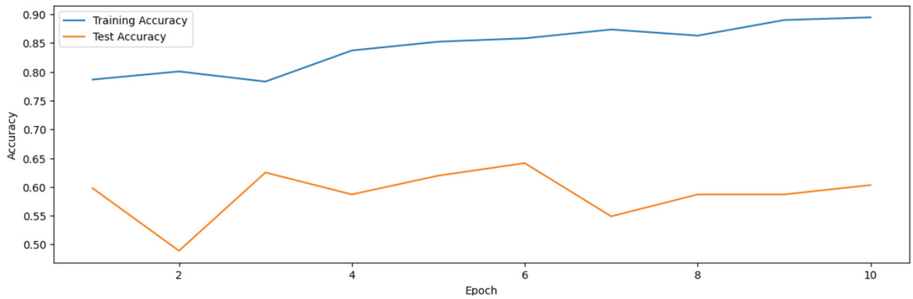

Fig. 8. MTF Maruti Loss vs Epoch

While testing Asian Paint Stocks Images using MTF, we found a significant variation in model performance among the pre-trained models, as shown in Table 5. Using the VGG16 model, we found 0.72, 0.36, and Precision, Recall, and F1-Score 0.48 for the Hold label, indicating a balanced model. On the other hand, using ResNet50, we found a reduced F1-Score of 0.34 for Hold, while obtaining a perfect Recall score for Buy labels. Such a diversity of pre-trained models highlights the sensitivity and robustness of the dataset. Notably, Xception showed the lowest performance with an F1-Score of 0.4 for the Hold label but showed promising results for the Buy and Sell labels. The improvement in performance due to the large amount of heterogeneity in the model emphasizes the importance of careful selection and optimization for specific financial image analysis tasks In analyzing the Maruti coefficients by MTF, as shown in Table 6, we found a large spread of results among the pre-trained models. VGG16 was used to note a slightly more balanced performance, with an F1-Score of 0.5 for Hold and 0.69 for Buy labels. However, with ResNet50, the F1-Score for the Sell label significantly decreased to 0.17, indicating difficulty in classifying this particular label. Using MobileNetV2, the Buy and Sell scores showed nearly identical patterns, indicating how similar the model can be. Xception's performance stood out with a high F1-Score of 0.7 for the Buy label and 0.86 Precision for the Sell label, although the challenges were indicated by a low Recall of 0.25 for the Sell label (Fig. 7 and Fig. 8). The average analysis of the 20 stock images by MTF, as shown in Table 6, presents a relatively consistent pattern between the previously trained models. VGG16 showed a moderate performance with an F1-Score of 0.49 for the Hold label, while ResNet50 showed a slightly better Precision for Hold but a lower F1-Score of 0.42. The InceptionV3 sample showed a balanced view with nearly identical F1 scores for the Hold, Buy, and Sell labels. Both MobileNetV2 and Xception followed a similar trend, with a trend toward better results for Buy labels. The consistency in these results may reflect the nature of the dataset, with an average of 20 stock images. This highlights the importance of a standardized approach that takes into account the specific characteristics of the stock under analysis in order to achieve optimal performance By analyzing three tables detailing the performance of the pre-trained models on stock image datasets we have found unique insights into the behavior of these models Differences in Accuracy, Recall, F1-Score, Accuracy between forms and constructions that need a standardized approach and f emphasize careful consideration of the specific characteristics of the dataset Understanding these nuances financial analysts and data scientists can properly select and refine models and has provided accurate and meaningful predictions in stock image analysis using MTF Corresponding statistics of accuracy vs. accuracy. and loss vs. loss. era further supports the findings, providing visual insight into the models' training dynamics. On average, 20 stock images were analyzed using both GAF and MTF methods. GAF analyses using VGG16, Xception, MobileNetV2, and InceptionV3 models revealed a complex terrain of model selection, with performance variability in different lines MTF method used for Asian Paint Stocks Image and Maruti Image, is explored in detail in the context of the pre-trained model While the GAF analysis emphasized the generic nature of the data set and the importance of standardized design, the MTF analysis emphasized the importance greater if rigorous analysis and targeted changes for optimal performance in stock classes are emphasized. Both approaches emphasize the importance of understanding and adapting to specific dataset characteristics to achieve optimal results.

5 Conclusion

The study's evaluation of the use of pre-trained convolutional neural networks (CNNs) to predict stock market trends shows significant improvements in financial forecasting. Using other data transformation techniques such as Gramian angular fields (GAF) and Markov transition fields (MTF) provides a new perspective on how to capture temporal dependence in stock market data. The efficiency of the MTF transform method and different results in different pre-trained models such as VGG16, ResNet50, etc. Emphasize the importance of the transform method and model selection. Future directions, such as model refinement, exploration of new data transformation techniques, and clustering techniques, mean that we can scale up this approach as we move towards a more data-intensive economy. Integrating these approaches could pave the way for more sophisticated and reliable tools in economic forecasting, and give the region the ability to navigate the ever-changing and complex world of capital markets to the sky.

Declaration of Competing Interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abhyankar, A., Copeland, L., Wong, W.: Non-linear dynamics in financial markets: evidence and implications. Econ. J. **107**, 864–880 (1997)
2. Hartman, D., Hlinka, J.: Nonlinear dependencies in international stock market returns: are they predictable? J. Int. Money Financ. **81**, 116–135 (2018)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proceedings of ICLR (2014)
4. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of CVPR (2015)
5. Szegedy, C., et al.: Inception-v3: rethinking the inception architecture for computer vision. In: Proceedings of CVPR (2016)
6. Wang, Z., Oates, T.: Imaging time-series to improve classification and imputation. In: 2015 International Conference on Image Processing (ICIP), pp. 2796–2800 (2015). IEEE
7. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, ??? (1970)
8. Nelson, D.B.: Conditional heteroskedasticity in asset returns: A new approach. Econometrica (1991)
9. Taylor, S.J.: Modelling financial time series (1986)
10. Brock, W., Lakonishok, J., LeBaron, B.: Simple technical trading rules and the stochastic properties of stock returns. J. Finance **47**(5), 1731–1764 (1992)
11. Wang, Z., Oates, T.: Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In: AAAI Workshops (2015)
12. Guo, Q., Song, Y., Li, X.: An innovative method for daily traffic flow forecasting using VLNN and GRNN. IEEE Access (2016)
13. Xie, J., Xu, X., Wang, S.: A comparison of denoising methods for chaotic time series with application to short-term traffic flow forecasting. Expert Systems with Applications (2016)
14. Lahmiri, S., Boukadoum, M.: Stock market forecasting using empirical mode decomposition coupled with neural network models. Neural Computing and Applications (2019)
15. Sezer, O.B., Ozbayoglu, M.: Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing (2018)
16. Zhang, W., Zhou, X., Yang, H., Wang, J.: Financial market prediction with a hybrid approach. Information Sciences (2019)
17. Patel, J.S., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock market index using fusion of machine learning techniques. Expert Systems with Applications (2015)
18. Li, L., Li, Q., Li, D.: The ARIMA+GARCH model application in the forecasting of stock index. Mathematical Problems in Engineering (2020)

19. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A.: Forecasting stock prices from the limit order book using convolutional neural networks. In: Business Informatics (CBI), 2017 IEEE 19th Conference on (2017)
20. Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one (2017)
21. Hoseinzade, E., Haratizadeh, S.: Deep learning in prediction of stock market indices: A case study of tehran stock exchange. Financial Innovation (2019)
22. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: Statistical and machine learning forecasting methods: Concerns and ways forward. PloS one (2018)
23. Gu, S., Kelly, B., Xiu, D.: Empirical asset pricing via machine learning. The Review of Financial Studies (2020)