

Enhanced Financial Time-Series Forecasting with Hybrid PSO-Fuzzy Rough Set-GRU Model

Subash Thiyagarajan and R. Sujatha *

School of Computer Science Engineering and Information Systems (SCORE), Vellore Institute of Technology, Vellore 14, India; Subash.t2023@vitstudent.ac.in,

* Corresponding author: r.sujatha@vit.ac.in

Received date: 8 July 2025; Accepted date: 27 October 2025; Published online: 31 December 2025

Abstract: Financial markets play an important role in the economic and social structure of contemporary society, with information being a valuable commodity. Yet the sheer volume of data available presents difficulties in analyzing financial assets. In this paper, we examine the effect of different deep learning models—GRU, LSTM, and CNN-LSTM— as well as a hybrid Particle Swarm Optimization (PSO) model integrated with Fuzzy Rough Set theory (FRS) and GRU in stock price prediction. From historical prices of stocks, we preprocess data and normalize feature vectors using MinMax scaling. For each model, extensive training and testing is conducted, using the performance as judged by MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) and the R^2 (coefficient of determination). Our results indicate that the PSO-Fuzzy Rough Set-GRU model performs greatly better than the others, having the lowest Training MAE (1.56), Testing MAE (2.10), Training RMSE (2.00), and Testing RMSE (2.82), along with the highest Training R^2 (0.9995) and Testing R^2 (0.9990). By comparison, single models such as LSTM and CNN-LSTM have higher error rates and lower R^2 values.

Keywords: prediction; deep learning; gated recurrent unit; particle swarm optimization; fuzzy rough set

1. Introduction

Predictive modeling in the financial markets involves inherently complex analysis of the nature of the time-based data which is volatile, multifaceted. For instance, predicting the stock prices may be a quite. It is an extremely challenging task due to unpredictable play of numerous internal and external factors like market sentiment, macroeconomic indicators, and geopolitical events. Normally, such techniques would incorporate the fundamental and technical analysis, always being insufficient due to their assumptions being static hence unable to conform to such developments in the market. This does make the issue even more cumbersome since of volumes of data since it can only be anticipated to be distorted by external influence such as how the economy alters and political movements that may tend to cause such sudden and radical changes in trends within the markets.

With the motivation to overcome the above issues in forecasting, new methods of broad application mainly rely on paradigms coming from machine learning, which enables the exploration of complex, non-linear relationships within financial data sets. Such stocks are dynamic investments and precise models for appropriate forecasts. Derived by the combination of both LSTM and GRU, the evaluation of neural network architectures is made through MAPE, RMSPE, and RMDPE for higher accuracies observed in this paper [1]. Hybrid models that are applied to recent studies about the stock prediction developed lately in deep learning compared well with earlier approaches. One-time step and multi time steps DAX, DOW, and S&P500 indices models had performed better forecasting based on this approach by proposing novel features [2] where they had also utilized the CNN - LSTM model from the daily series data of an accessible stock period varying from the year 1991 to the year 2020 with their eight



major features related to stocks. The CNN extracts feature, and LSTM predicts prices with respect to other models such as MLP, RNN, and CNN-RNN using the objective that can achieve practical insights about financial time series analysis with high prediction accuracy in the choice [3]. The stock exchange companies also make investments based on their financial performance. This strategy dynamically predicts the performance of a company over time with online financial data, combining fuzzy C-means, DEA, and ANN [4]. Another hybrid model using multi-technical indicators and RST is targeted for predicting the trend of stock prices to avoid technical analysis complexity and conventional ARMA and neural network model limitations. From the results shown above, the proposed system seems to outperform better than RST and GAs using the TAIEX data, with an applicative time duration of six years [5]. A primary extension based on this method with prediction accuracy through applying PSO optimisation in training ANN models that utilized NASDAQ data, resulted from its potentiality, being based on both bias and the weights of the model [6]. Adding PSO in Fuzzy Rough Set models improves the optimization and generalization, thereby leading to improved performance of the model for predictive tasks.

This research, in turn, gives insight into the possibility of hybrid models in financial forecasting and propounds a comprehensive framework for sound stock price forecasting. The integration of bio-inspired algorithms with deep learning could hence be an interesting direction in which the future growth in financial analytics may be expanded. The paper compares four machine learning models, namely GRU, LSTM, CNN-LSTM, and a new model PSO-FRS-GRU. An extensively improved study allows a high accuracy with a reliable prediction by giving exact results. Such a study will show with cross-validation of the method to be used, mainly with other available data sets, which will give insight into the merits and demerits of each type of methodology for predicting future stock prices.

2. Literature Review

Predicting future stock market values is still a challenging task, and there are many studies on different approaches to enhance the accuracy of predictions. RNN and LSTM models have been used to analyze the prediction accuracy along with the epoch adjustments and proved to be more effective than traditional machine learning algorithms, especially in customer sentiment analysis and trend change detection. A single-layer LSTM model to predict the next-day closing price of the S&P 500, including nine predictors from market data, macroeconomics, and technical indicators. The results show that the single-layer LSTM outperforms multilayer models, with superior prediction accuracy based on RMSE, MAPE, and correlation coefficient [2,7]. It features the LSTM-CNN model, which merges the temporal features extracted from the stock time series and features from the candlestick charts in predicting stock prices. It is observed that the model performs better than the stand-alone CNN and LSTM models while reducing the prediction error by utilizing both feature types on SPDR S&P 500 ETF data [8]. This is a hybrid deep-learning model to predict stock prices, which predicts close and high prices for the next day. Experiments on the Shanghai Composite Index show that a single-layer RNN outperforms CNN, LSTM, CNN-RNN, and CNN-LSTM, and improves the prediction accuracy up to 2.2%, which makes investors make better decisions [9].

The passage does well to explain the basic idea behind the proposed CNN-LSTM model based on its capability to extract feature using CNN and predict stock prices using LSTM. It also quite effectively describes the dataset, specifying key features like opening price, volume, and turnover, thereby clearing up the research [3]. The Stock Sequence Array Convolutional LSTM (SACLSTM) framework, which exploits historical data and leading indicators, has also proven to enhance prediction accuracy [10]. A deep learning-based model for live stock price predictions is developed using Fast Recurrent Neural Networks (FastRNNs) and a hybrid model combining FastRNNs, CNN, and Bi-Directional LSTM. The proposed models outperform ARIMA, FBProphet, and LSTM in terms of lower RMSE and computational complexity for tested 1-minute interval stock data, making them suitable for live predictions [11]. This develops a multivariate CNN-LSTM hybrid ensemble model for time-series forecasting of stock market indices by leveraging the strengths of CNN and LSTM. The model was tested on the indices from four Asian markets during the COVID-19 pandemic and has proven superior accuracy and reliability to standalone CNN and LSTM models, proving its superiority for financial time-series prediction [12]. There are various forecasting models proposed with technical indicators, investor sentiments, and financial data. It is found that LASSO-based models outperform the PCA-based models [13]. Modified GRUs designed to handle challenges such as overcoming local minima and reducing the time complexity which have delivered accuracy in their results verified through the RMSE [1].

Hybrid models like CNN-LSTM and GRU-CNN have demonstrated tremendous improvement in terms of MSE, MAE, as well as the accuracy of multi-time-step forecasting [14]. Nature-inspired algorithms have been utilized to improve the predictability models. Some techniques, which used Bat Algorithm (BA), Flower Pollination Algorithm (FPA), Artificial Bee Colony (ABC), Modified Cuckoo

Search (MCS), and PSO to improve the performance efficiency of Least Squares Support Vector Machine (LS-SVM), have led to increased accuracy and faster convergence on S&P 500 datasets [15]. The ANN-PSOCoG model is proposed, coupling artificial neural networks and the particle swarm optimization algorithm, to boost the adequacy of stock market prediction, at better processing times with reduced errors, as compared to the current models. Tested on the S&P 500, DJIA, GOLD, NASDAQ-100, and CANUSD datasets, this model proved superior accuracy in precision even during COVID-19 with a substantially reduced MAPE, MAE, and RE values [6]. The fine-tuned version of Crow Search Algorithm (CSA) with an Extreme Learning Machine improves the level of predictions of the closing price the day after for the stock indices within the COVID-19 pandemic [16]. A hybrid model using adaptive particle swarm optimization for weights to forecast major index stocks such as Sensex, S&P 500, Nifty 50 is proposed here. This novel hybrid has provided the ability to overcome the difficulties of LSTM in its learning process. Optimization and the genetic algorithm-based hybridization overpowers both; followed by an overall LSTM was tested with a few forecasts where accuracy was superior, other varieties of PSO with Centre of Mass, Centre of Gravity yield excellent predictive precisions and accordingly, reduced error rates as opposed to the results published earlier in [17,18]. Rough Set and Fuzzy-Rough Set models have been explored for predicting Indian stock prices, yielding more accurate results compared to Support Vector Machines, Random Forest, and other models [5]. In predicting crude oil prices, incorporating detectors of various sizes and applying feature selection via fuzzy-rough set theory has improved accuracy [19]. This technique applied to CSI300 futures achieved an average return of 6.36% with better performance than benchmark methods [20]. Therefore, the proposed integration of PSO, FRS, and GRU is a breakthrough in the field of stock price forecasting. The convergence acceleration of PSO optimizes the model parameters to avoid getting stuck in the local minima.

Meanwhile, Fuzzy Rough Set theory handles data uncertainty well, improves feature selection and model robustness. GRUs are also effective in handling sequential data, catching the important temporal dependencies that become a must in cases where the precision in the forecasting is a concern. With all these strengths coming together, this method gathers superior accuracy as well as superiority in efficiency by surpassing all the existing approaches and offering innovative solutions to all the complexities arising with financial forecasting.

3. Methodology

Therefore, this is one of the advancements in stock price prediction by integrating PSO, Fuzzy Rough Set theory, and GRU. Since each method benefits from the previous one, better accuracy and performance are achieved with this approach.

3.1. Dataset Selection and Preparation

Stock market data comprises multiple attributes and types, while the stock price presents a form of temporal dependency. The features are either univariate or multivariate; hence a preliminary analysis on the dataset is required before approaching the prediction part. For this paper, a standardized open source dataset was utilized that consisted of a diverse range of attributes and behaviors to effectively depict the proposed architecture model. Table 1 shows the selected datasets. BSE OIL & GAS, HDFC BANK, RELIANCE COMMUNICATION, TATA MOTORS, and TATA STEEL are the stock market data of prominent organizations. The datasets were taken from open repositories like Yahoo Finance Repository [21] and BSE [22]. The medium to large size of these datasets contributes to obtaining meaningful and reliable results. The input dataset needs to be analyzed to verify the existence of missing values and noisy data. If the dataset has errors or missing values, proper imputation and processing steps should be carried out [23]. Moreover, normalization of the data is carried out when necessary to remove biases that could influence the results [24]. Because stock price prediction is a temporal pattern within supervised learning, the data is split into a training subset and a testing subset with a ratio of 75:25. Data preprocessing works by indexing the timestamps, then defining the closing price as the target attribute for predicting. Once the dataset is prepared, it will be used for input in data modeling and computation of the outcome of the prediction. The next phase details are discussed in the subsequent subsection.

Table 1. Datasets Description.

Datasets	Type	Instances	Attributes	Time stamps
BSE OIL & GAS	Multivariate	1131	5	Date-wise (01/01/2020 - 9/07/2024)

HDFC BANK	Multivariate	1125	7	Date-wise (01/01/2020 - 9/07/2024)
R COM	Multivariate	1125	7	Date-wise (01/01/2020 - 9/07/2024)
TATA MOTORS	Multivariate	1125	7	Date-wise (01/01/2020 - 9/07/2024)
TATA STEEL	Multivariate	1125	7	Date-wise (01/01/2020 - 9/07/2024)

3.2. Convolutional Neural Network

It uses an application of convolutional layer using a neural network during the process of feature extraction and processing the input data. There are primarily three types of layers of CNNs, typically: Fully Connected, Pooling, and Convolutional. Within this layer, features are attained through filters whereas in the Max Pooling layer, the feature dimension is down-sized. Usually, CNNs are quite effective for feature extraction and performing image processing over varied media types. This FilterNet is an elastic architecture in deep learning for time series classification that combines CNN and LSTM motifs to enhance the accuracy and efficiency in activity recognition tasks. Benchmarking on the Opportunity dataset demonstrates FilterNet to outperform existing models both in accuracy and speed, yet leaves options towards customization towards various applications [25]. Exactly, for sequential data [26], 1D-CNNs were used, therefore, we picked CNN-STM for feature extraction since stock prices are sequential data and temporal data. Recent studies prove that the coupling of LSTM models with net growth calculation algorithms increase the prediction in future stock markets using historical data. This utilizes the machine learning that identifies the patterns as well as the improved predictability without market fluctuations or intricate complexities [8].

3.3. LSTM Model

LSTM is a type of specialized RNN, particularly used for the purpose of handling long-term dependencies and consecutive data patterns. Contrary to standard RNNs, LSTMs address the problem of the vanishing gradient through unique mechanisms called the forget, input and output gates; these regulate what information to be kept or discarded and when the information flowing through the memory cell is to be updated. This capability makes LSTMs particularly useful for applications involving time-series forecasting, natural language processing, and any task where understanding sequence and context is essential.

$$\text{Forget gate } F_t = \sigma(G_f \cdot [h_{t-1}, x_t] + B_f) \quad (1)$$

$$\text{Input gate } I_t = \sigma(G_i \cdot [h_{t-1}, x_t] + B_i) \quad (2)$$

$$\text{Cell state update gate } C_t = \tanh(G_c \cdot [h_{t-1}, x_t] + B_c) \quad (3)$$

$$\text{Cell state } C_t = F_t \odot C_{t-1} + I_t \odot C_t \quad (4)$$

$$\text{Output gate } O_t = \sigma(G_o \cdot [h_{t-1}, x_t] + B_o) \quad (5)$$

$$\text{Hidden state } C_t = O_t \odot \tanh(C_t) \quad (6)$$

Here, \tanh denotes the hyperbolic tangent function, G_f , G_i , G_c , and G_o are weight matrices, σ represents the sigmoid function, while B_f , B_i , B_c , and B_o are biases. The symbol \odot indicates element-wise multiplication.

3.4. CNN-LSTM

The CNN-LSTM model integrates Convolutional Neural Networks with Long Short-Term Memory, using the strength of both architectures, Convolutional Neural Network (CNN), It is used to extract spatial features from the input data. Here, it assists in sequential data to look for patterns.

- Conv1D Layer: 1D convolution on the input sequence. It applies filters to get features.
- MaxPooling1D Layer: reduces the dimensionality of the data by selecting the maximum value in a window.
- GlobalAveragePooling1D Layer: Reduces the output from the previous layer to a fixed size by computing the average over time steps. The input for the Dense layers simplifies the network.
- LSTM (Long Short Term Memory)
 - Role: This architecture captures temporal dependency in sequential data. It captures long-term dependency without facing vanishing gradient.
- Dense Layers: Fully connected layers that take the features discovered by the CNN and produce output
 - LSTM networks are a variant of RNNs designed to handle long-term dependencies in sequence data. Here are the essential components for LSTM units: refer the equations 1 to 6.

Algorithm 1: The pseudo-code for the Hybrid PSO-Fuzzy Rough Sets-GRU
Input: Data x, PSO parameters, Fuzzy Rough Sets parameters, GRU parameters Output: Optimized GRU parameters, Predictions y 1. Initialize Parameters <ul style="list-style-type: none"> • PSO Parameters: <ul style="list-style-type: none"> – Set the number of particles for PSO. – Set cognitive coefficient C1. – Set social coefficient C2. – Set inertia weight D. • Fuzzy Rough Sets Parameters: <ul style="list-style-type: none"> – Define granulation parameters for data. – Define fuzzy membership functions. • GRU Parameters: <ul style="list-style-type: none"> – Initialize weight matrices G, U. – Initialize bias vectors B. 2. Particle Swarm Optimization (PSO) <ul style="list-style-type: none"> (a) Particle Initialization: <ul style="list-style-type: none"> • Initialize positions X_i and velocities V_i for each particle. (b) Fitness Evaluation: <ul style="list-style-type: none"> • Transform Data: <ul style="list-style-type: none"> Granulate Data: Partition the data x into granules G using fuzzy rough set analysis. – Membership Degree: Compute membership degrees $\mu_{G_i}(x)$ for each granule G_i: $\mu_{G_i}(x) = \frac{e^{-\frac{\ x-c_i\ ^2}{\sigma_i^2}}}{e^{-\frac{\ x-c_j\ ^2}{\sigma_j^2}}}$ • Train GRU Model: <ul style="list-style-type: none"> – Forward Pass: <ul style="list-style-type: none"> * Update Gate z_t: using Eq. 7 * Reset Gate r_t: using Eq. 8 * Candidate Hidden State \tilde{h}_t: using Eq. 9 * Final Hidden State h_t: using Eq. 10 – Loss Function: Compute the loss using Mean Squared Error (MSE): using Eq. 15 (c) Adjust the position and velocity of each particle: <ul style="list-style-type: none"> • Adjust the velocity and position of every particle: using Eq. 11,12 (d) Update Best Positions: <ul style="list-style-type: none"> • Update personal best and global best positions based on fitness evaluation. 3. Fuzzy Rough Sets Analysis <ul style="list-style-type: none"> • Granulate Data: Partition data into granules using fuzzy rough set analysis. • Membership Functions: Compute membership degrees for data points. • Evaluate Granulation: Assess the quality of granulation using measures such as accuracy or precision. 4. GRU Training <ul style="list-style-type: none"> (a) Prepare Data: Use transformed data from Fuzzy Rough Sets for training the GRU model. (b) Train GRU Model: Use the optimized parameters from PSO.

- Forward Pass and Loss Calculation: Perform forward pass and compute loss as described in the PSO evaluation.
- Update Parameters: Update GRU parameters using gradient descent:
 $\theta \leftarrow \theta - \eta \nabla_{\theta} Loss$
- 5. Return Results:
 - Return the optimized GRU model parameters and predictions y .

The CNN-LSTM combines all the benefits of convolutional neural networks to add to long- and short-term memory networks for predictive tasks on sequential data, which most have to do with sequences of information. The following step is applying a layer of convolution, in which this process extracts features due to a pre-defined kernel operation size and the stride applied between successive elements, and further processes each feature through some activation function: usually ReLU. The output is then downsampled using a pooling layer, usually Max Pooling, to reduce the dimensionality while preserving the important features.

The pooled feature maps will then be flattened in one-dimensional vectors and stacked to make a sequence that allows LSTM processing. LSTM network processes such a sequence over time steps, computing the input, forget, and output gates at each step in order to update the cell and hidden states, which are important for capturing temporal dependencies in the data. The final hidden state from the final time step is used for prediction. Then, a dense layer followed by an activation function, such as softmax or similar, is added to produce the final output, the predicted sequence of values. In this algorithm, spatial feature extraction is combined well with temporal modeling of sequences, making it highly successful for tasks where sequential data is in use.

3.5. GRU Model

GRUs are engineered to manage sequential data by identifying dependencies across time while,

$$\text{Update gate } z_t = \sigma(G_z \cdot [h_{t-1}, x_t] + B_z) \quad (7)$$

$$\text{Reset gate } R_t = \sigma(G_r \cdot [h_{t-1}, x_t] + B_r) \quad (8)$$

$$\text{Candidate activation } \tilde{h}_t = \tanh(G_h \cdot [R_t \odot h_{t-1}, x_t] + B_h) \quad (9)$$

$$\text{Hidden state } h_t = (1 - Z_t) \odot h_{t-1} + Z_t \odot \tilde{h}_t \quad (10)$$

3.6. Proposed Architecture

In this paper, we try to contribute to the prediction for a financial market by evaluating and comparing several deep-learning models. We focus on four models: GRU, LSTM, CNN-LSTM, and a new model that integrates PSO with Fuzzy Rough Sets and GRU. The GRU and LSTM models are known for their effectiveness in capturing time-based dependencies within time-based data. The CNN-LSTM model captures the feature mining strength of CNN and the sequential modeling capability of LSTM. The introduced PSO-Fuzzy Rough Set-GRU model will try to integrate these techniques toward solving the issue of non-linearity and uncertainties associated with financial data. Figure 1. and Figure 2. are step-by-step model development representations of the procedure, ranging from data preprocessing to feature processing by Fuzzy Rough Set and finally model training by PSO.

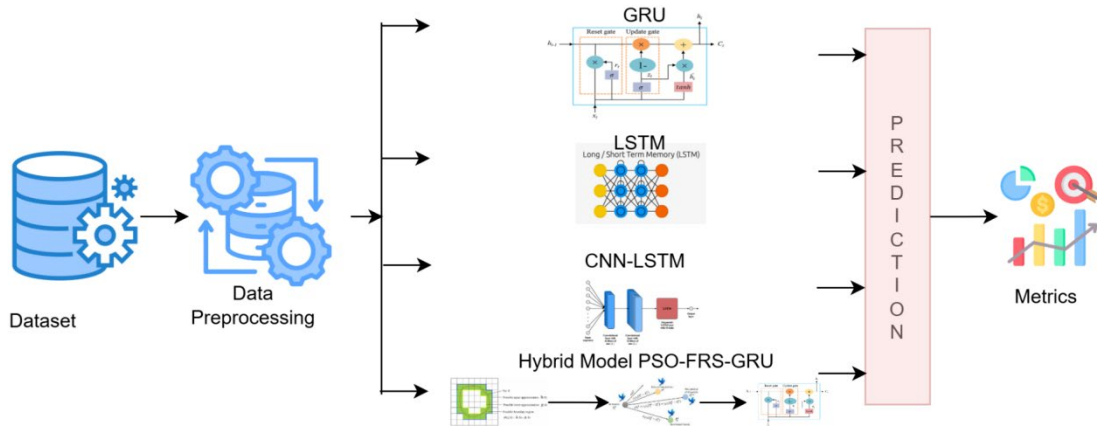


Figure 1. Block diagram of the GRU, LSTM, CNN-LSTM and proposed Financial Time-Series Forecasting risk prediction process.

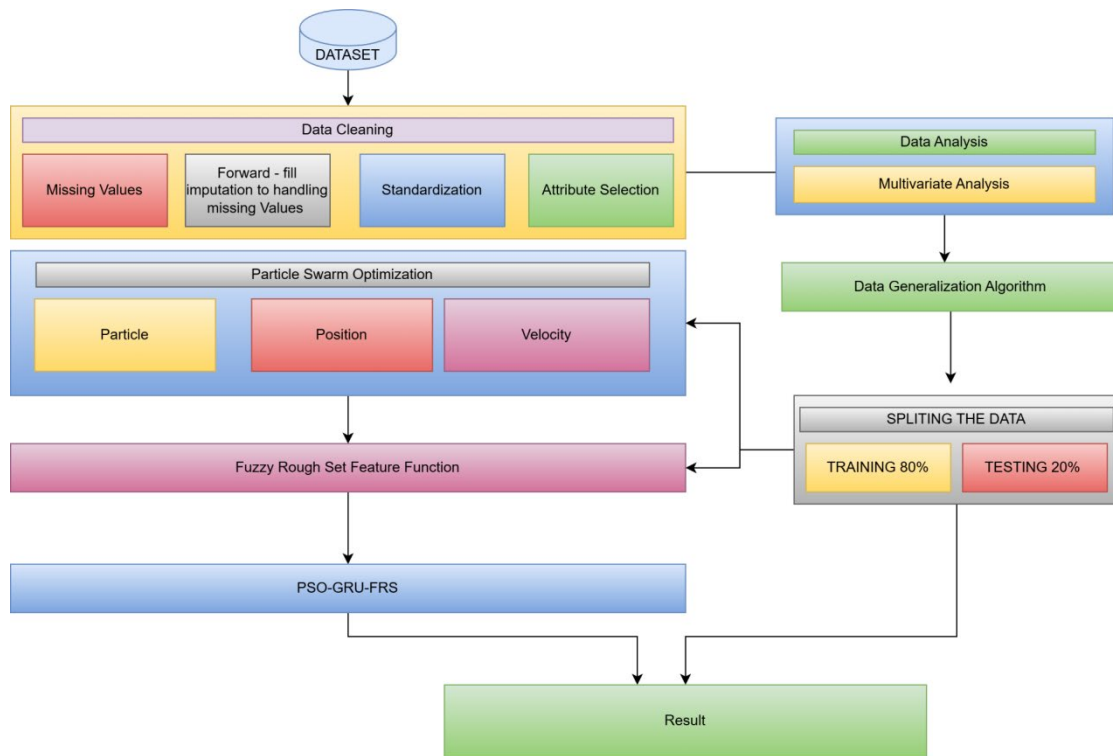


Figure 2. Schematic architectural diagram of the proposed Financial Time-Series Forecasting risk prediction framework.

3.7. Particle Swarm Optimization (PSO)

PSO is an optimization procedure derived from the behavior patterns of birds and fish, which in turn exploit a host of candidate solutions called elements traveling through the search space with a view to finding the best solution for the problem. ANN for predicting stock market indices, optimizing the ANN model with PSO for better accuracy. Results from NASDAQ index data indicate an improvement of PSO in the prediction accuracy of ANN by optimizing bias and weights [6].

The key steps are Particle Velocity Update

$$\text{Velocity Update } v_{i,s+1} = D \cdot V_{i,s} + C_1 \cdot R_1 \cdot (P_{i,best} - X_{i,s}) + C_2 \cdot R_2 \cdot (G_{best} - X_{i,s}) \quad (11)$$

where:

- $v_{i,s+1}$: Updated velocity of element i at iteration $s + 1$
- D : Inertia weightiness
- C_1, C_2 : Cognitive and communal coefficients
- R_1, R_2 : Random numbers among 0 and 1
- $P_{i,best}$: Individual best position of element i
- G_{best} : Global best position

$$\text{Particle position Update } X_{i,s+1} = X_{i,s} + v_{i,s+1} \quad (12)$$

where:

- $X_{i,s+1}$: Updated position of element i at iteration $s + 1$

3.8. Fuzzy Rough Sets

FRST (Fuzzy-Rough Set Theory) [27] generalizes Rough Sets using a non-crisp equivalence relation to delineate the limits of a Fuzzy Set in terms of a FIR (Fuzzy Indiscernibility Relation) [28]. Fuzzy Rough Sets enable to deal with the uncertainty and vagueness inherent in data. It includes, Function of Membership, Illustrates the degree to which an element is contained in a fuzzy set.

$$\mu_A(x) = \frac{1}{1 + e^{-\frac{(x-c)}{\sigma}}} \quad (13)$$

where:

- $\mu_A(x)$: Membership degree of x in fuzzy set A
- c : Center of the membership function
- σ : Spread of the membership function

Rough Set Calculation

- Lower Calculation: The collection of elements that definitely belong to the target set.
- Upper Calculation: The group of elements that potentially belong to the target set.

3.9. GRU

The GRU is a specialized RNN aimed at enhancing learning speed and reducing model complexity by modifying standard network functions. Its main features are to refer the equations 7 to 10.

3.10. Evaluation Metrics

Having fine-tuned our models, we apply the well-established evaluation criteria in the finance time-series prediction domain [29], RMSE, MAE, MSE, and R^2 . These we have computed over the predictions in the test set. The corresponding formulations are mentioned below

ions are mentioned below

$$\text{Mean Absolute Error} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (14)$$

where:

- N : Amount of observations
- y_i : True worth for observation i
- \hat{y}_i : Forecast worth for observation i

$$\text{Mean Squared Error} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (15)$$

Root Mean Squared Error

$$RMSE = \sqrt{MSE} \quad (16)$$

R-squared

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (17)$$

where:

- \bar{y} : Mean of the true values

Algorithm 1 "PSO-Fuzzy Rough Set-GRU" is designed to enhance the presentation of GRU (Gated

Recurrent Unit) models for financial time-series forecasting by incorporating Particle Swarm Optimization (PSO) and Fuzzy Rough Set Theory. The algorithm starts with the initialization of PSO parameters, such as the number of particles, cognitive and social coefficients, and inertia weight. In addition, Fuzzy Rough Set parameters, including granulation parameters and membership functions, along with GRU parameters, are initialized.

It does particle initialization, fitness evaluation in the process using Fuzzy Rough Sets for data granulation, and then training through the model GRU. Fitness is assessed based on the performance of the model by GRU so that the positions and velocities of the particles will be updated to fine-tune the GRU parameters. Updating the best positions continues to refine the model further.

After applying PSO optimization, data granulation along with the computation of membership functions using Fuzzy Rough Sets, are carried out, and as a result, all data are aptly prepared to proceed for training of GRU. The GRU training, using optimized parameters that were gained by applying PSO optimization, involves an update mechanism on GRU parameters via gradients during training. The algorithm finally outputs the optimized GRU model parameters and the predicted output to prove the success of PSO-FRS integration in enhancing the performance of GRU-based forecasting models. Data origin (Yahoo Finance Repository and BSE), Preprocessing stage (feature selection, data normalization, and fuzzy rough set transformation), the Split (80:20), the Boundaries Setting (PSO bounds, swarm size, and number of iterations), and the Implementation tool (Google Colab, Python, Keras, and Scikit-learn) were all configured and set. Figure 3. shows a graphical representation of how the algorithm works.

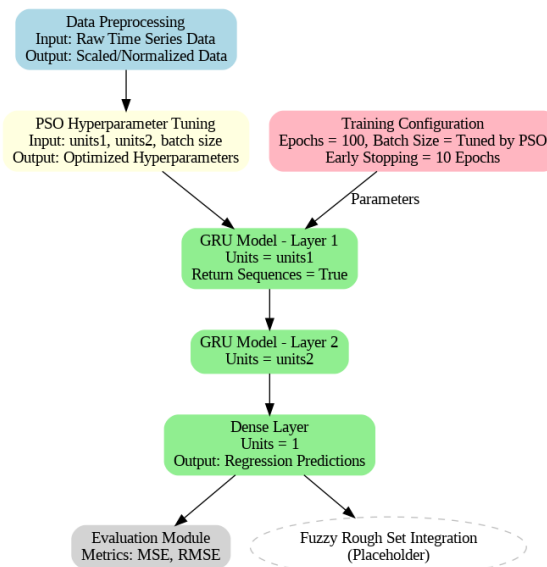


Figure 3. GRU-PSO Hybrid Framework with Fuzzy Rough Set Integration.

3.11. Model Configuration

In this paper, we use a few architectures that are specifically optimized for stock price prediction using deep learning. Table 2 gives an LSTM model with one 1D convolutional layer having 64 filters and a kernel size 3, followed by one 1D max pooling layer having pool size 2, followed by a flattened layer. A dense layer of size 50 followed by a ReLU activation is attached to feed this configuration, and the LSTM is applied post-density for handling a sequence of timesteps. It begins with the CNN- LSTM model, first starting with 1D convolutions but employing 64 filters with a size of 2. It consists of the unit MaxPooling1D and finishes it off with GlobalAveragePooling1D for the purpose of making dimensionality further reduced. It concludes with a dense neural layer where the first dense neural layer has 50 units with ReLU activation, and the output layer contains one unit for regression. The GRU model is set up with two GRU layers with 50 units in each. In this structure, the first layer returns sequences.

A dense layer follows, which contains a single unit for regression output. The framework is optimized using the Adam optimization algorithm and trained with a squared error loss function over 100 epochs with a batch size of 32, and early stopping is implemented to measure validation loss through a patience parameter set to 10, along with automatic restoration of best weights. In an advanced variant hybrid PSO-Fuzzy Rough Set-GRU, the units within GRU layers plus the batch size, are tuned via Particle Swarm Optimization.

This model, in addition also includes a single unit of regression output with the dense layer. Again, its training configuration is same as GRU model. An additional function to be suggested using Fuzzy

Rough Set has also been proposed so that even this can increase prediction accuracy.

Table 2. Hybrid Model Architecture and Parameter Details.

Model	Parameters
LSTM	1D Convolutional Layer: Filter Count: 64, ReLU activation function with Kernel size of 3. 1D Max Pooling Layer with a pool size of 2. Layer for flattening the input: Flattens output, Fully connected layer: Units: 50, Using the ReLU activation function, LSTM Layers: Added after dense layer for temporal sequences.
CNN-LSTM	1D Convolutional Layer: Number of filters: 64, Size of the kernel: 2, ReLU activation function, followed by a 1D max pooling layer with a pool size of 2. GlobalAveragePooling1D Layer: Reduces dimensionality, Dense Layers: First Dense Layer: Units: 50, ReLU activation function, Output Fully connected layer: Units: 1 (for regression output).
GRU	GRU Layers: Units: 50 (x2 layers), Return Sequences: True for the first layer, Dense Layer: Units: 1 (for regression Output layer), Optimizer: Adam, Loss Function: MSE Training Configuration: Epochs: 100, Batch Size: 32, Early Stopping: Monitor: 'val_loss', with patience of 10 epochs and restoring the best weights.
PSO-Fuzzy Rough Set-GRU	GRU Layers: Units: Tuned by PSO (units1, units2), Return Sequences: True for first layer, Dense Layer: Units: 1 (for regression outputlayer), Optimizer: Adam, Loss Function: MSE, Training Configuration: Epochs: 100, Batch Size: Tuned by PSO, Early Stopping: Monitor: 'val_loss', with a patience of 10 epochs and restoring the best weights. Fuzzy Rough Set Function: Placeholder, needs implementation.

4. Results and Discussion

4.1. Model Performance Comparison

This research work evaluates the performance of four deep learning models, namely GRU, LSTM, CNN-LSTM, and a hybrid PSO-Fuzzy Rough Set-GRU model over five different datasets: BSE OIL&GAS, HDFC BANK, RCOM, TATA MOTORS, and TATA STEEL. Key performance metrics have been used to analyze the training and testing errors. These include MAE for both training and testing, RMSE for both training and testing, and the R2 coefficient. As can be seen in Table 3, the PSO-Fuzzy Rough Set-GRU model always dominates the other models for nearly all the performance metrics for all the datasets. It produces the least error rates with the lowest values of MAE and RMSE, as well as the highest values of R2, thus better accuracy and reliability for the financial time-series forecasting task.

4.2. Detailed Dataset Analysis

Figure 4. displays a graphical comparison of Training and Validation Loss Curves for Models for Dataset BSE Oil & Gas. The PSO-Fuzzy Rough Set-GRU model obviously surpasses others. For instance, the Testing MAE achieved by the model is 79.42, and the Testing RMSE is 116.66. LSTM and CNN-LSTM models are considerably higher in error rates, having Testing RMSE values of 352.75 and 907.23, respectively. R² value for the hybrid model is 0.988, which is indicative of an extremely high level of predictive accuracy. Similarly, Figure 5. illustrates the Training and Validation Loss Curves of the

models on the HDFC Bank dataset. The Testing MAE, Testing RMSE, and R^2 of the PSO-Fuzzy Rough Set-GRU model for the HDFC Bank dataset are 2.10, 2.82, and 0.999, respectively. But CNN-LSTM model is outperformed with a Testing RMSE of 56.00 and an R^2 value of 0.575, and it proves the superiority of the hybrid model.

Figure 6. offers a visual comparison of Loss Curves for training and validating the models applied to the Reliance Communication dataset. The hybrid model leads with a Testing MAE of 0.0068 and a Testing RMSE of 0.0095, while its R^2 value of 0.997 confirms its robustness compared to the CNN-LSTM model, which has a Testing RMSE value of 0.111 and a relatively low R^2 of 0.590. Likewise, Figure 7. shows the Training and Validation Loss Curves for models based on the TATA Motors dataset. The PSO-Fuzzy Rough Set-GRU model has a Testing MAE of 3.53 and a Testing RMSE of 4.18, with an R^2 value of 0.982, reflecting high predictive power, while the CNN-LSTM model has a negative R^2 value, reflecting poor fit. Finally, Figure 8. illustrates the Training and Validation Loss Curves for the TATA Steel dataset, where the hybrid model maintains its dominance with a Testing MAE of 1.36 and a Testing RMSE of 2.10, while achieving an R^2 value of 0.892. In contrast, the LSTM and CNN-LSTM models result in higher error rates and lower R^2 values, further proving the superiority of the PSO-Fuzzy Rough Set-GRU model.

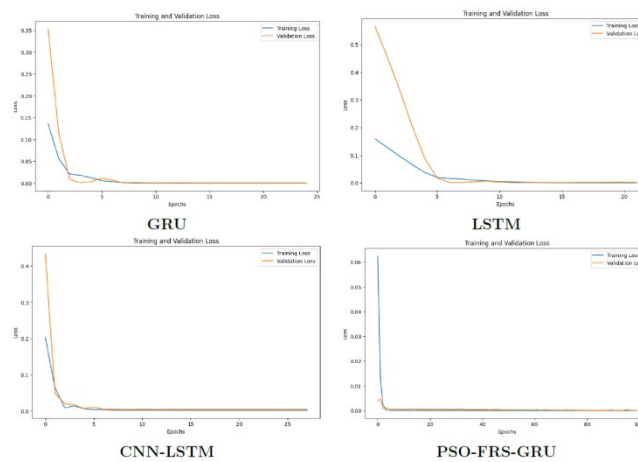


Figure 4. BSE Oil&Gas Dataset Training and Validation Loss Curves for Models

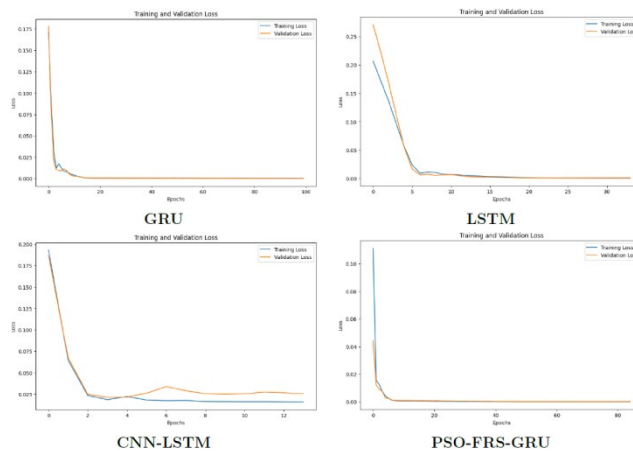


Figure 5. HDFC Dataset Training and Validation Loss Curves for Models

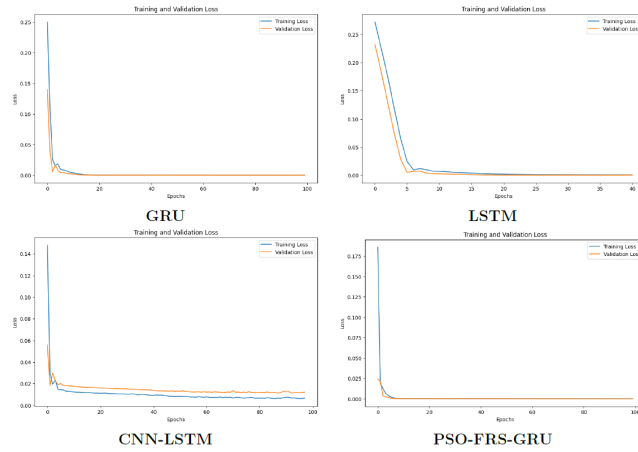


Figure 6. Reliance Communication Dataset Training and Validation Loss Curves for Models

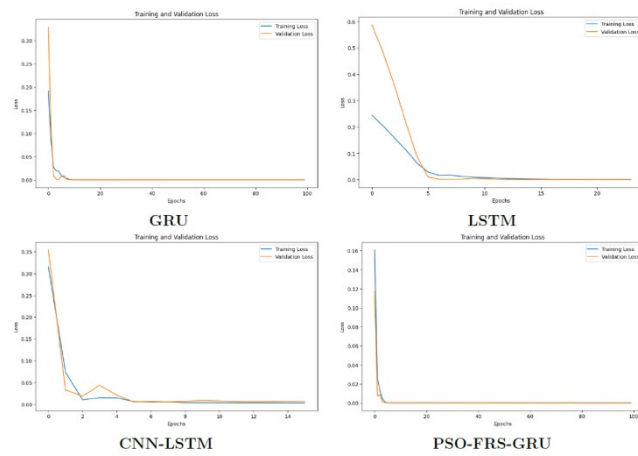


Figure 7. TATA Motors Dataset Training and Validation Loss Curves for Models

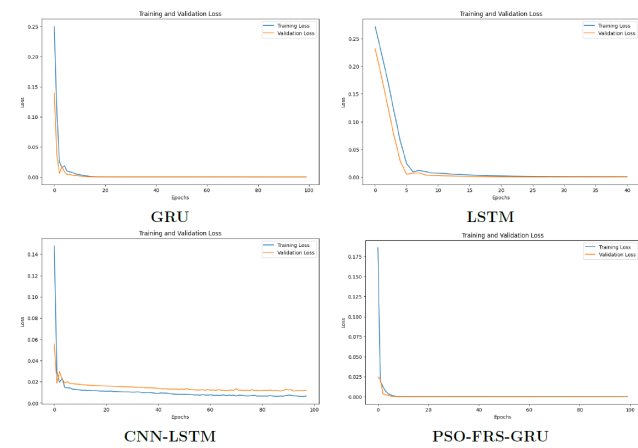


Figure 8. TATA Steel Training and Validation Loss Curves for Models

4.3. Discussion

The results clearly indicate that the incorporation of Particle Swarm Optimization with Fuzzy Rough Set Theory to the GRU model enhances its performance. The hybrid approach optimizes the parameters of the GRU model much better, giving it more accurate generalization capabilities and the accuracy of prediction over different datasets. The hybrid model always shows the lowest error rates and highest R2 values, ensuring reliability and robustness in tasks related to financial forecasting.

Figure 9. depicts the Monte Carlo Simulation of Testing MAE for Different Models Across Datasets. The equation used in Monte Carlo simulations is

$$simulation = \frac{1}{N} \sum_{i=1}^N x_i \quad (18)$$

where x_i represents the resampled MAE values and N denotes the number of simulations.

Figure 9. Monte Carlo Simulation of Training and Testing MAE for Different Models Across Various Datasets

Figure 10. Simulated Visualization of Stochastic Differential Equations (SDE) to Analyze the Variance in Training and Testing Mean Absolute Error (MAE) by Different Models for Different Datasets. Using Euler-Maruyama, it will generate paths of MAE. It visualizes the mean with confidence intervals in terms of the time steps taken. The equation of SDE used is:

$$dX_t = \mu X_t dt + \sigma X_t dw_t \quad (19)$$

where μ is the drift term, σ is the volatility term, and dw_t is the Wiener process increment. Figure 11. Figure 12. and Figure 13. are shows Comparative Performance Analysis of Various Models Across Different Datasets Using MAE, RMSE, and R^2 . Figure 14. is Sample images for BSE Oil & Gas Dataset, actual vs predicted Training and Validation Data for Model PSO-Fuzzy Rough Set-GRU.

Table 3. Performance Comparison of Different Models on Various Datasets.

DATASET	Model	Training MAE	Testing MAE	Training RMSE	Testing RMSE	Trainin g R^2	Testing R^2
BSE OIL	GRU	121.7827	199.1759	175.0512	356.0498	0.9981	0.8889
	LSTM	549.3547	214.3051	653.4935	352.7503	0.9730	0.8909
	CNN-LSTM	455.2087	742.5183	637.8823	907.2320	0.9746	0.3035
	PSO-FRS-GRU	45.0826	79.4217	64.4995	116.6556	0.9997	0.9881
HDFCBANK	GRU	3.0288	4.2271	4.1673	5.6329	0.9977	0.9960
	LSTM	9.1070	9.9279	12.1034	12.7678	0.9805	0.9794
	CNN-LSTM	42.9749	48.2101	57.0345	56.0012	0.5363	0.5758
	PSO-FRS-GRU	1.5572	2.0981	2.0041	2.8194	0.9995	0.9990
RCOM	GRU	0.0112	0.0103	0.0158	0.0155	0.9963	0.9925
	LSTM	0.0249	0.0147	0.0346	0.0183	0.9824	0.9895
	CNN-LSTM	0.0636	0.0829	0.0840	0.1119	0.8781	0.5910
	PSO-FRS-GRU	0.0079	0.0068	0.0105	0.0096	0.9984	0.9971
TATA MOTORS	GRU	3.6932	5.4453	4.7871	6.5571	0.9990	0.9564
	LSTM	17.9638	5.6730	22.9752	7.5897	0.9771	0.9416
	CNN-LSTM	23.6300	27.6417	29.5245	32.0411	0.9631	-0.2134
	PSO-FRS-GRU	2.4624	3.5343	3.2043	4.1824	0.9996	0.9823
TATA STEEL	GRU	0.6527	0.7576	0.8621	1.1023	0.9966	0.9705
	LSTM	2.2019	1.2447	2.7567	1.6264	0.9648	0.9359
	CNN-LSTM	3.6545	5.4986	4.5175	6.6356	0.9063	-0.1521
	PSO-FRS-GRU	1.2030	1.3636	1.5489	2.1050	0.9889	0.8926

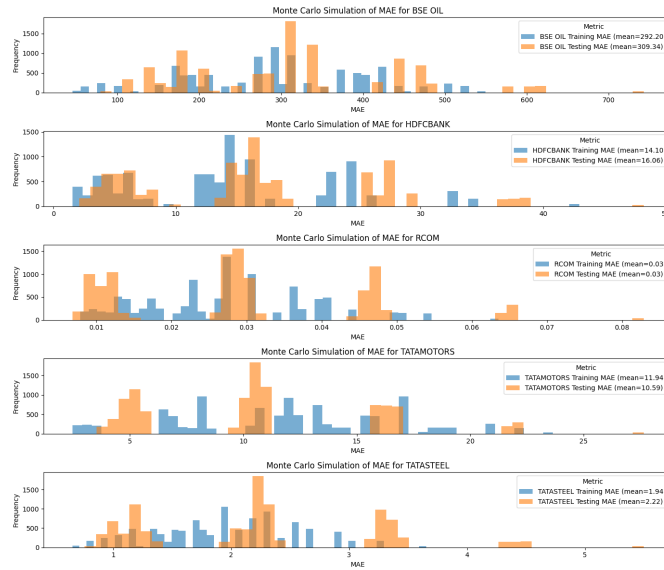


Figure 9. Monte Carlo Simulation of Training and Testing MAE for Different Models Across Various Datasets

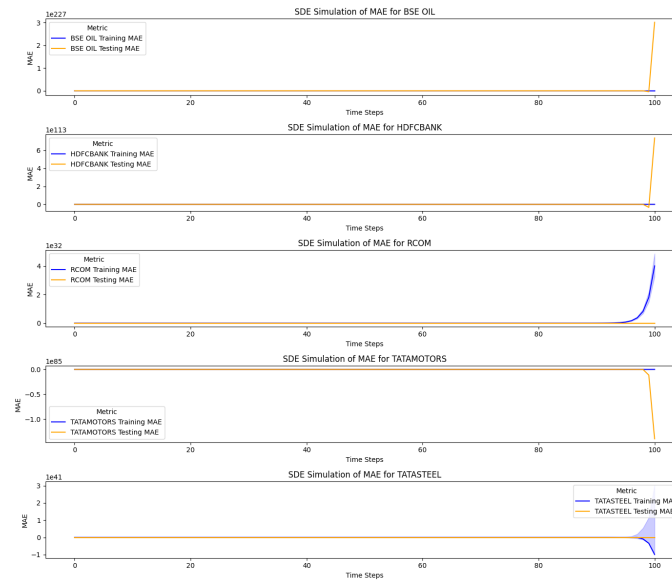


Figure 10. Stochastic Differential Equation (SDE) Simulation of Training and Testing MAE for Different Models Across Various Datasets

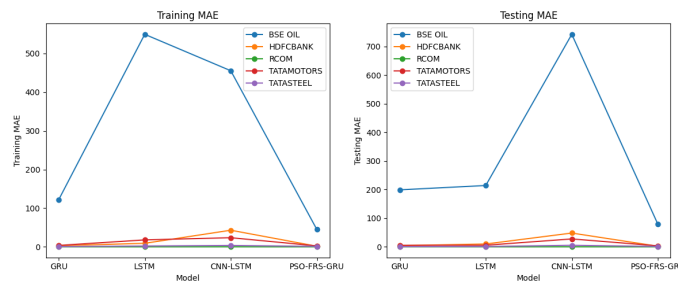


Figure 11. Comparative Performance Analysis of Various Models Across Different Datasets Using MAE Metrics.

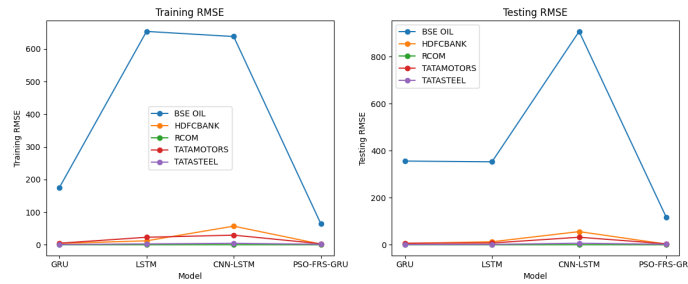


Figure 12. Performance Comparison of Different Models Across Various Datasets Using RMSE Metrics.

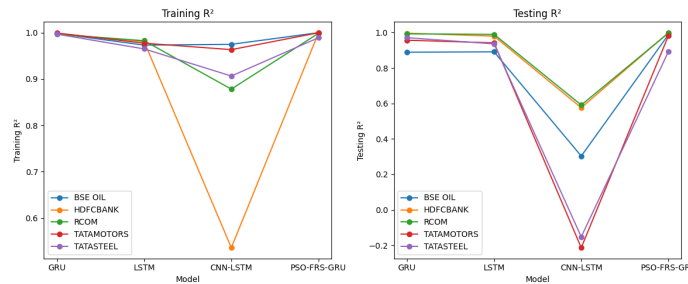


Figure 13. Comparison of Model Performance Across Various Datasets Using R² Metrics.

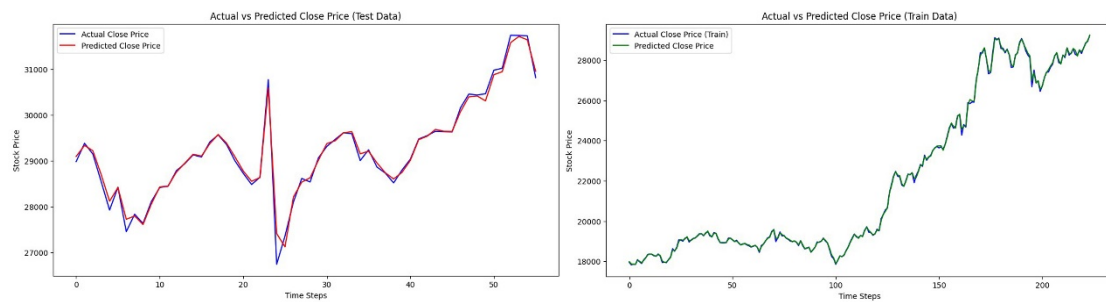


Figure 14. Sample images for BSE Oil & Gas Dataset, actual vs predicted Training and Validation Data for Model PSO-Fuzzy Rough Set-GRU.

5. Conclusion

This research demonstrated that the hybrid PSO-Fuzzy Rough Set-GRU model is better than the traditional deep learning models, such as GRU, LSTM, and CNN-LSTM, in the financial time-series forecasting across multiple datasets. Indeed, it holds the lowest error rates of MAE with RMSE values and the highest R² values, which are more indicative of higher predictive accuracy and reliability. This performance improvement is due to the integration of PSO and Fuzzy Rough Set theory that optimizes the model GRU and enhances its generalization capability. Hybrid approaches can potentially give more accurate and dependable financial predictions, making the PSO-Fuzzy Rough Set-GRU model very useful for accurate stock price forecasting. Future research may include the improvement of the PSO-Fuzzy Rough Set-GRU model by using alternative optimization algorithms, such as Genetic Algorithms or Differential Evolution, and advanced feature engineering techniques which include macroeconomic indicators and sentiment analysis.

Further research in the model could perhaps confirm its stability in other forms of applications outside financial markets, like establishing cryptocurrencies and global stock exchanges. Fundamental research into the model can also be conducted through real-time data environments, such as high-frequency trading, to check if it can predict longer trends and volatility in a market. This model, with comparative analyses with even newer deep learning models and efforts to make the model more explainable in conjunction with practical implementation in real-world financial scenarios, would be great for development and potential industry adoption. Stock prices should be segmented into more stationary elements using hybrid decomposition techniques, such as CEEMDAN or wavelets. Within the GRU layers, attention mechanisms are implemented to understand the sequential aspects more effectively. Variance is reduced and model robustness is increased through ensemble learning with GRU, LSTM,

and CNN-LSTM predictions.

Author Contributions

Subash Thiyagarajan wrote the main manuscript text, prepared figures, and tables. R. Sujatha reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding

The authors received no specific funding for this study.

Conflict of Interest Statement

There is no conflict of interest.

Data availability statement

The following information was supplied regarding the codes are publicly available at GitHub: <https://github.com/subhashtiyagarajan-sudo/Subash-123/blob/main/stock.ipynb>.

Acknowledgment

This work was carried out independently, without any assistance from supporting agencies. we express our sincere gratitude to Vellore Institute of Technology, Vellore, Tamil Nadu, India. Support and provision of resources throughout this research.

References

1. Rahman, M. O., Hossain, M. S., Junaid, T. S., Forhad, M. S. A., & Hossen, M. K. (2019). "Predicting prices of stock market using gated recurrent units (GRUs) neural networks". *Int. J. Comput. Sci. Netw. Secur*, 19(1), 213-222.
2. Moghar, A., & Hamiche, M. (2020). "Stock market prediction using LSTM recurrent neural network". *Procedia computer science*, 170, 1168-1173.
3. Lu, W., Li, J., Li, Y., Sun, A., & Wang, J. (2020). "A CNN-LSTM-based model to forecast stock prices". *Complexity*, 2020(1), 6622927.
4. Sugumar, R., Rengarajan, A., & Jayakumar, C. (2014). "A technique to stock market prediction using fuzzy clustering and artificial neural networks". *Computing and Informatics*, 33(5), 992-1024.
5. Roy, A. S., & Chatterjee, N. (2022). "Forecasting of Indian stock market using rough set and fuzzy-rough set based models". *IETE Technical Review*, 39(5), 1105-1113.
6. Ghashami, F., Kamyar, K., & Riazi, S. A. (2021). "Prediction of stock market index using a hybrid technique of artificial neural networks and particle swarm optimization". *Applied Economics and Finance*, 8(1), 10-11114
7. Pawar, K., Jalem, R. S., & Tiwari, V. (2019). "Stock market price prediction using LSTM RNN. In *Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS2018* (pp. 493-503). Springer Singapore.
8. Ghosh, A., Bose, S., Maji, G., Debnath, N., & Sen, S. (2019, September). Stock price prediction using LSTM on Indian share market. In *Proceedings of 32nd international conference on (Vol. 63, pp. 101-110)*.
9. Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017, September). "Stock price prediction using LSTM, RNN and CNN-sliding window model". In *2017 international conference on advances in computing, communications and informatics (icacci)* (pp. 1643-1647). IEEE.
10. Wu, J. M. T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C. W. (2023). "A graph-based CNNLSTMstock price prediction algorithm with leading indicators. *Multimedia Systems*", 29(3), 1751-1770.
11. Mehtab, S., & Sen, J. (2020, November). "Stock price prediction using CNN and LSTMbased deep learning models". In *2020 International Conference on Decision Aid Sciences and Application (DASA)* (pp. 447-453). IEEE.
12. Eapen, J., Bein, D., & Verma, A. (2019, January). "Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction". In *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)* (pp. 0264-0270). IEEE.
13. Gao, Y., Wang, R., & Zhou, E. (2021). "Stock prediction based on optimized LSTM and GRU models. *Scientific Programming*", 2021(1), 4055281.
14. Song, H., & Choi, H. (2023). "Forecasting stock market indices using the recurrent neural network based hybrid models: CNN-LSTM, GRU-CNN, and ensemble models". *Applied Sciences*, 13(7), 4644.
15. Hegazy, O., Soliman, O. S., & Salam, M. A. (2015). "Comparative study between FPA, BA, MCS, ABC, and PSO algorithms in training and optimizing of LS-SVM for stock market prediction". *International Journal of Advanced Computer Research*, 5(18), 35-45.

16. Das, S., Sahu, T. P., Janghel, R. R., & Sahu, B. K. (2022). "Effective forecasting of stock market price by using extreme learning machine optimized by PSO-based group oriented crow search algorithm". *Neural Computing and Applications*, 34(1), 555-591.
17. Seidy, E. E. (2016). "A new particle swarm optimization based stock market prediction technique". *International Journal of Advanced Computer Science and Applications*, 7(4).
18. Jamous, R., ALRahhal, H., & El-Darieby, M. (2021). "A New ANN-Particle Swarm Optimization with Center of Gravity (ANN-PSOCoG) Prediction Model for the Stock Market under the Effect of COVID-19". *Scientific Programming*, 2021(1), 6656150.
19. Lasisi, A., Tairan, N., Ghazali, R., Mashwani, W. K., Qasem, S. N., GR, H. K., & Arora, A. (2019). "Predicting crude oil price using fuzzy rough set and bio-inspired negative selection algorithm". *International Journal of Swarm Intelligence Research (IJSIR)*, 10(4), 25-37.
20. Deng, S., Zhu, Y., Liu, R., & Xu, W. (2022). "Financial futures prediction using fuzzy rough set and synthetic minority oversampling technique". *Advances in Mathematical Physics*, 2022(1), 7622906.
21. <https://finance.yahoo.com>
22. <https://www.bseindia.com>
23. Nayak, S. C., Misra, B. B., & Behera, H. S. (2014). "Impact of data normalization on stock index forecasting". *International Journal of Computer Information Systems and Industrial Management Applications*, 6, 13-13.
24. Jaiswal, R., & Singh, B. (2021, December). "A Comparative Study of Loss Functions for Deep Neural Networks in Time Series Analysis". In *International Conference on Big Data, Machine Learning, and Applications* (pp. 147-163). Singapore: Springer Nature Singapore.
25. Guo, Z., Wang, H., Liu, Q., & Yang, J. (2014). "A feature fusion based forecasting model for financial time series". *PloS one*, 9(6), e101113.
26. Tang, W., Long, G., Liu, L., Zhou, T., Blumenstein, M., & Jiang, J. (2020). "Omni-scale cnns: a simple and effective kernel size configuration for time series classification". *arXiv preprint arXiv:2002.10061*.
27. Jensen, R., & Shen, Q. (2008). "New approaches to fuzzy-rough feature selection". *IEEE Transactions on fuzzy systems*, 17(4), 824-838.
28. Chatterjee, N., & Yadav, N. (2019). "Fuzzy rough set-based sentence similarity measure and its application to text summarization". *IETE Technical Review*, 36(5), 517-525.
29. Bisong, E. (2019). "Building machine learning and deep learning models on Google cloud platform" (pp. 59-64). Berkeley, CA: Apress.

Author Biographies



First Author Mr. T. Subash is an experienced academician with over 15 years of teaching expertise. He holds a B.Sc. in Mathematics from C. Abdul Hakeem College (Madras University), an MCA from Adhiparasakthi College of Science (Madras University), and an M.Phil. from Thiruvalluvar University. From 2008 to 2023, he served as an Assistant Professor at Dr. MGR Chockalingam Arts and Science College, affiliated with Thiruvalluvar University. During his period, he held multiple leadership roles, including Head of the Department (HOD), Placement Officer, AISHE (All India Survey on Higher Education) Coordinator from 2012 to 2023, and NPTEL Single Point of Contact (SPOC), showcasing his dedication to academic excellence and student development. Currently, Mr. T. Subash is pursuing his Ph.D. as a research scholar at Vellore Institute of Technology (VIT), further advancing his expertise and contributing to his field of study finance using Deep Learning.



Second Author Prof. Dr. R. Sujatha completed her Ph.D. degree at Vellore Institute of Technology, Vellore, India, in the area of data mining. She received her M.E. degree in Computer Science from Anna University, India, with ninth rank, and pursued a Master of Financial Management from Pondicherry University, India. She received her B.E. degree in Computer Science from Madras University, India. She has 20 years of teaching experience and has been serving as a professor in the School of Computer Science Engineering and Information Systems at Vellore Institute of Technology, Vellore. Also served as Assistant Director (Students' Welfare). She has organized and attended several workshops and faculty development programs. She actively involves herself in the growth of the institute by contributing to various committees at both academic and administrative levels. She used to guide projects for undergraduate and postgraduate students and currently guides doctoral students. She gives technical talks in colleges for symposiums and various sessions. She acts as an advisory, editorial, and technical committee member in conferences conducted in other educational institutions and in-house. She has published a book titled *Software Project Management for College Students*. She has published research articles in reputed high-impact journals and can find 90 articles in Google Scholar, 23 book chapters, 3 edited books, and also authored books. The Institution of Green Engineers awarded her IGEN Women Achiever 2021 award in the Future Computing category. Outstanding Researcher Awardee for the year 2022 was awarded by VIT. She is interested in exploring and visiting different places to learn about the culture and people of various areas. She is interested in learning upcoming things and gets acquainted with the student's level. Her areas of research interest include Information Management and Data Analytics, Artificial Intelligence, Soft Computing, Medical AI, Big Data, Deep Learning, and Blockchain Software Engineering.