SimPSOEnsemble : Hybrid PSO-SA for Exploration and Exploitation for Stock Market Prediction

Abstract—**Accurate stock price prediction is essential for informed financial decision-making and investment strategies. However, due to the non-linear nature of time series data, modeling a mathematical basis for this remains a challenging task. Although LSTM models are usually efficient for sequential data handling, they are often sensitive to the hyperparameter settings. In this scenario, metaheuristic algorithms have shown to give good results in searching for the global best settings. This paper introduces a novel approach that integrates an ensemble of Long Short-Term Memory (LSTM) networks with a hybrid Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithm for robust hyperparameter optimization. The proposed methodology leverages LSTM's proficiency in capturing temporal dependencies and optimizes the strengths of PSO and SA to search the subspace. This hybrid approach addresses the exploration-exploitation tradeoff. The experiments are conducted on the benchmark Nifty 50 Indian Stock market index (NSEI) to demonstrate that the ensemble-LSTM model significantly outperforms individual LSTM models and traditional forecasting methods in terms of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The findings highlight the effectiveness of combining sequential temporal learning networks with advanced optimization techniques to enhance stock price prediction accuracy.**

Index Terms—**Long-Short Term Memory (LSTM), Particle Swarm Optimisation (PSO), Simulated Annealing (SA), Stock Market Forecasting, Exploration-exploitation**

I. Introduction

Predicting stock market movements has been a long-standing and extensively researched problem that continues to captivate investors, mathematicians, scientists, and the general public alike. However, accurately forecasting stock prices remains inherently challenging due to the market's complex and volatile nature. Financial markets are influenced by a myriad of factors, including economic indicators, geopolitical events, and investor sentiment, all of which contribute to unpredictable and highly non-linear dynamics.

Numerous studies across various markets have attempted to address this problem using diverse methods ranging from traditional statistical models, such as ARIMA [12] and Support Vector Machines (SVM) [13] to advanced techniques, including artificial neural networks [15], Long Short-Term Memory (LSTM) networks [1] [7] [9], Convolutional Neural Networks (CNN), and their hybrid variants. Some studies have explored the integration of swarm intelligence methods, such as Particle Swarm optimisation [6] [14]. Additionally, a few studies have investigated the implementation of deep learning techniques and LSTM models for the Nifty 50 index and other Indian stock markets [11].

A fundamental challenge in accurate stock prediction lies in the highly non-linear and temporally variable nature of financial data, which complicates the detection of meaningful patterns and trends. Furthermore, the hyperparameter configuration space of complex models such as LSTMs is vast and intricate, making it difficult to identify optimal settings that can significantly improve predictive performance. Effective hyperparameter tuning is crucial, as it directly impacts the model's ability to generalize and accurately forecast future stock prices. Metaheuristic algorithms have proven to be effective tools for hyperparameter tuning [8] [10], as they are gauranteed to converge as opposed to arbitrary hyperparameter selection.

In this paper, we introduce a novel hybrid framework that employs an ensemble of Long Short-Term Memory (LSTM) networks optimized using a hybrid Particle Swarm Optimization (PSO) and Simulated Annealing (SA) algorithm. Our proposed method encompasses the following components:

- **Ensemble Learning Approach**: Utilizing multifold LSTM models to capture a diverse and complex array of temporal patterns, enhancing prediction accuracy and robustness of the model.
- **Hybrid Optimization Method**: Implementing novel Simulated Annealing(SA) and PSO pipeline to explore the hyperparameter space effeciently. Employing SA to prevent premature convergence to local optima stochastically increases the chances of a more optimal result.
- **Balanced Exploration-Exploitation**: Addressing the critical exploration-exploitation tradeoff in optimization of functions to improve model performance.
- **Performance Validation**: Conducting comprehensive experiments on the Nifty50 index, demonstrating that our optimized ensemble-LSTM model significantly outperforms traditional forecasting methods in terms of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

These findings highlight the effectiveness of combining advanced optimization techniques with sequential temporal learning networks, contributing valuable insights into enhancing stock price prediction accuracy in complex financial markets.

II. PRELIMINARIES

A. Long Short Term Memory

The Long Short Term Memory (LSTM) forms the backbone of this research. The LSTM model is one of the variants of the RNN. Its core contribution is to introduce the design of self-loop to generate the path of a gradient which could continuously flow for an extended period. The weight of the self-loop is also updated in each iteration, which solves the gradient vanishing problem that is easily generated when the RNN model updates the weights. The modelling of a time series is essentially a process of nonlinear parameter fitting. The LSTM model performs well to reveal the correlation of a nonlinear time-series in the delay state space and to realize the purpose of stock prediction. The stock or Forex trend prediction model based on LSTM obtained the corresponding data characteristics from the stock or Forex history data.

Mathematically, the LSTM cell operations are defined as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$c_t = f_t * c_{t-1} + i_t * \tilde{C}t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(c_t)$$

where σ denotes the sigmoid activation function, x_t is the input at time t, h_t is the hidden state, and C_t is the cell state and b_o is the bias vector.

B. Simulated Annealing

Simulated Annealing (SA) is a probabilistic optimization algorithm inspired by the annealing process used in metallurgy, where controlled cooling reduces the system's energy. It is used to find the global optima in optimization problems with complex non-linear search spaces. SA operates by iteratively exploring solutions, accepting both improvements as well as potentially worse solutions (based on the temperature parameter) to escape local optima. The acceptance of new candidates is determined by the Metropolis Criterion.

SA was chosen as the exploration algorithm as it is effective in efficiently escaping the problem of premature convergence to a local minima when only PSO is used and promote a more thorough search of the hyperparameter configuration subspace.

C. Particle Swarm Optimisation

Particle Swarm Optimisation is a population-based metaheuristic optimization algorithm inspired by the social behavior observed in bird flocking and fish schooling in nature. It aims to search and find the optimal solution in a multidimensional search space by simulating the pattern and movement of particles. Like other metaheuristics, PSO begins with the initialization of particles with velocity and position set arbitrarily. Each particle is a representation of a solution. The particle positions and velocities are updated iteratively based on a fitness function and a global best solution found by the swarm. PSO involves the movement of particles through a search space based on mathematical calculations. The update equations for the particle's velocity (V_i) and position (X_i) in a typical PSO algorithm are expressed as follows:

The velocity update equation is given by:

$$V_i(t+1) = w \cdot V_i(t) + c_1 \cdot r_1 \cdot (P_i(t) - X_i(t)) + c_2 \cdot r_2 \cdot (G(t) - X_i(t)) \tag{1}$$

where $V_i(t+1)$ is the updated velocity of particle i at time $t+1$, w is the inertia weight, c_1 and c_2 are the cognitive and social coefficients, respectively, r_1 and r_2 are random values between 0 and 1, $P_i(t)$ is the best-known position of particle i at time t, $G(t)$ is the best-known position of the entire swarm at time t, and $X_i(t)$ is the current position of particle i at time t. The position update equation is given by:

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

PSO is widely used for optimizing complex, multidimensional functions and has proven effective in hyperparameter tuning for machine learning models.

III. PROPOSED METHOLOGY

In this section we explicitly discuss our proposed methodology. The methodology entails optimizing the hyperparameters of an ensemble of LSTM models using a hybrid Particle Swarm Optimization and Simulated Annealing (SimPSO) algorithm. The ensemble LSTM approach enhances predictive accuracy by aggregating multiple LSTM models, each trained with different hyperparameter configurations. The SimPSO model effectively navigates the hyperparameter space, balancing exploration and exploitation to identify optimal configurations that minimize validation loss.

In our experiments, we have configured the SimPSO algorithm with the following parameters, they have been set keeping in mind our computational power:

Parameters	Values
Number of Particles	5
Maximum Iterations	3
SA Cooldown	2
SA Initial Temperature	1000
Inertia Weight (w)	0.5
Cognitive Coefficient (c1)	1
Social Coefficient (c2)	2
Perturb fraction	0.1

TABLE I: SimPSOEnsemble Configuration Parameters

A swarm of five (arbitrarily chosen number) particles is initialized, where each particle represents a unique set of hyperparameters. Each particle's position corresponds to a specific hyperparameter configuration, and its velocity dictates its movement through the hyperparameter space. The SA part of the SimPSO algorithm is governed by the following parameters:

- Inertia Weight (w) : Balances the particle's tendency to continue moving in its current direction (exploration)

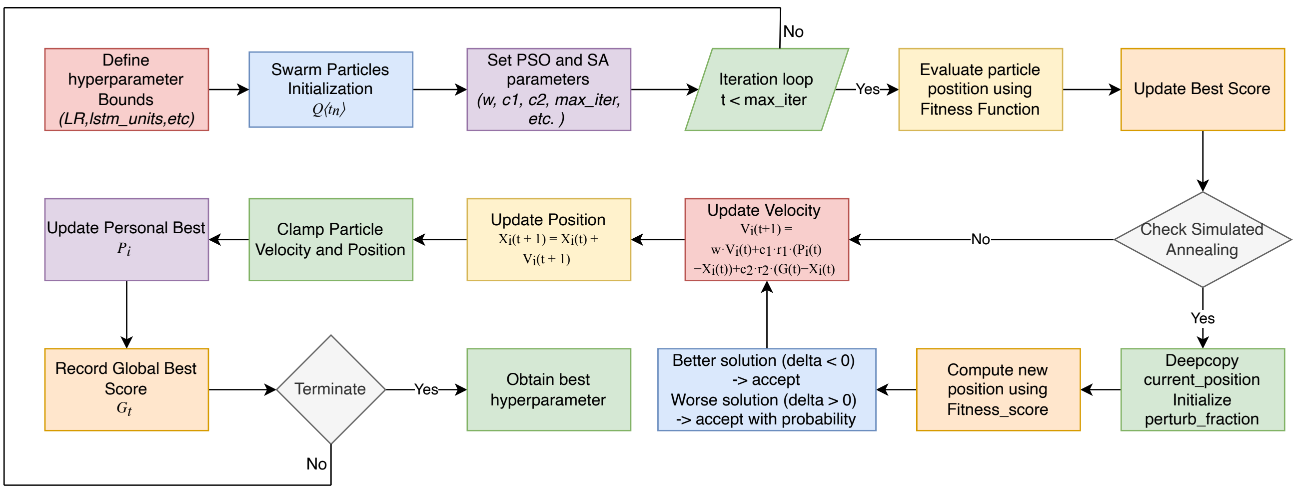


Fig. 1: SimPSOEnsemble Model Overview

against its tendency to be influenced by personal and global best positions (exploitation).

- Cognitive Coefficient (c1) : Represents the particle's tendency to return to its personal best position.
- Social Coefficient (c2) : Represents the particle's tendency to follow the swarm's global best position.
- SA Parameters : Includes `sa_cooldown`, which determines the frequency of SA application, and `sa_initial_temp`, the initial temperature for the SA process.

Each particle's fitness is evaluated using a predefined fitness function which in our case is the average loss of the LSTM models. The fitness function guides the SimPSO algorithm towards configurations that yield lower predictive errors. Within the fitness function we train an ensemble of LSTM models with the PSO chosen hyperparameters and compute the average Mean Squared Error (MSE) on the training dataset. after this, at specified intervals determined by sa_cooldown, the Simulated Annealing (SA) mechanism is applied to each particle to introduce stochastic perturbations. This integration is the crux to enhancing the exploration capabilities of the swarm, allowing particles to explore new regions of the hyperparameter space and thereby avoid stagnation in local optima.

The SA algorithm consists of the following parts which alter or rather perturb the space and path undertaken by the PSO

- Perturbation: Each hyperparameter is slightly modified within a fraction of its range (perturb_function) randomly. This controlled randomness introduces diversity and facilitates exploration.
- Acceptance Criterion: The new hyperparameter configuration is accepted based on the change in fitness and the current temperature. Better solutions are always accepted, while worse solutions are accepted with a probability that decreases as the temperature cools, following the Boltzmann probability distribution.
- Temperature Cooling: After each SA application, the temperature is reduced according to a cooling schedule (e.g., multiplicative decay). This gradual cooling diminishes the probability of accepting worse solutions over time, shifting the algorithm's focus more towards exploitation in later iterations.

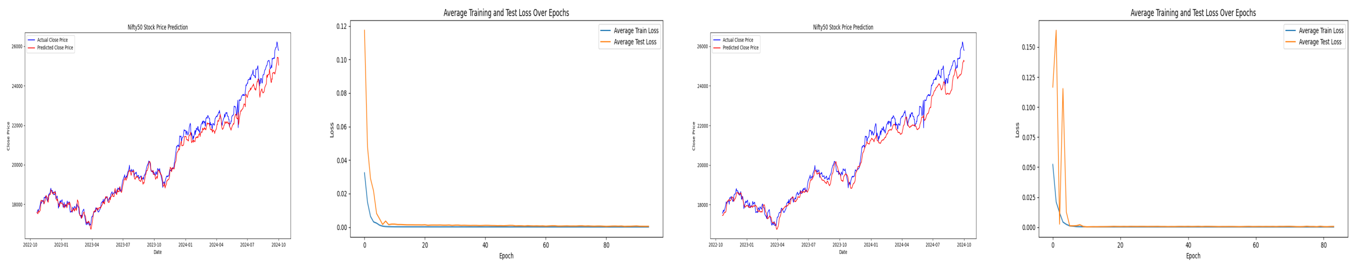
Following the annealing, each particle's velocity and position are updated based on its personal best and the global best, incorporating cognitive and social influences. The velocity update formula Equation (1) integrates random coefficients to introduce stochasticity. Through repeated iterations we converge at the most optimal hyperparameter configuration set, which is used to train the ensemble LSTM network.

As mentioned, the ensemble network comprises of multiple LSTM base models, each trained with the set of hyperparameters identified by the SimPSO algorithm. The ensemble prediction is obtained by averaging the predictions of individual models, thereby enhancing robustness and accuracy. By leveraging an ensemble approach, the methodology ensures that the predictive performance is not reliant on a single model's biases or limitations, resulting in more reliable and consistent forecasting outcomes.

IV. Experiment Details

A. Dataset Description

The study utilizes historical daily stock price data of the Nifty 50 index, a benchmark Indian stock market index that represents the weighted average of 50 of the largest Indian companies listed on the National Stock Exchange. The dataset spans a period of ten years recorded daily, starting from September 1st, 2014, to September 1st, 2024, totalling to about 2455 trading days. This data is procured from the Yahoo Finance (yfinance) API , that gives historically accurate and real-time data. The API returns a comprehensive overview of the daily open price, high price, low price and close price.



(a) SimPSO : Real vs Predicted (b) SimPSO: Train Vs Test Loss (c) PSO : Real vs Predicted (d) PSO : Train Vs Test Loss

Fig. 2: Comparison of Results Between SimPSOEnsemble and PSO

B. Dataset Preprocessing

The dataset is pre-processed by checking for missing values, and then the data points are normalized using Min-Max Normalization. Normalization ensures a homogeneous distribution of datapoints. Owing to the sequential nature of stock price, input sequences of fixed length are used to capture temporal dependencies. In our study, we use sequences of length 60 (arbitrary) consecutive trading days as input to the LSTM models, with the subsequent day's closing price serving as the target output. The resulting 2455 data points are split in the 80:20 ratio (Train: 1916, Test: 479). The splitting is done chronologically in order to preserve the temporal integrity and information of the data.

C. Experimental Setup

The experimentation was carried out using Macbook M3 chips, Intel CORE i5 processors and Kaggle cloud computational environment powered by the GPU T4 x2 accelerator cores. Tensorflow and Scikit-learn framworks and libraries such as numpy, pandas, pyswarm and matplotlib were used.

V. Results and Discussion

In this section we present the outcomes of our study and analyze the performance of our proposed architecture against different architectures that have been predominantly used. Our aim was to enhance the forecasting ability of our model, to predict the close price of the Nifty 50 market Index at the end of every trading day with increased precision. Table II shows the RMSE, MAE and the MAPE for the different models we experimented upon:

TABLE II: Comparision of RMSE, MAE, MAPE of different models

Models	RMSE	MAE	MAPE
Proposed SimPSOEnsemble	287.21	197.45	0.0118
Standalone PSO Ensemble	357.21	279.28	0.0155
LSTM	1065.97	867.54	0.1797
Artificial Neural Network	1659.75	1291.10	0.2234

The desirable results are in accordance to our hypothesis that preceeding the ensemble LSTM architecture with a hybrid optimization algorithm would greatly boost the accuracy of the model. This is because the LSTM models are highly sensitive to the initial parameters and narrowing down the LSTM hyperparameters to the globally optimal configuration improves the model considerably.

By incorporating the hybrid model of PSO exploitation and SA exploration, our research leverages the strengths of both PSO and SA to navigate complex search spaces more efficiently and effectively. This highlights that the integration of meta-heuristics is instrumental in avoiding premature convergence forimproved predictive accuracy. The hyperparameters obtained from the different models are presented in Table 3:

TABLE III: Best Hyperparameter Configurations

Parameter	SimPSO	PSO
Learning Rate	0.00157	0.00436
Ensembles	5	8
Units	153	200
Dropout	0.05	0.01
Batch Size	332	296
Epochs	84	95

Although the SA algorithm adds a computational overhead making it the slowest model (approximately 45-55 minutes per training), its performance is considerably more refined than the standalone models. Fig. 2 shows the comparison between the SimPSOEnsemble and PSO models.

VI. Conclusion and Future Work

In this paper, we presented a novel hybrid approach that combines Particle Swarm Optimization (PSO) and Simulated Annealing (SA) within an exploitation-exploration framework to optimize hyperparameters for an ensemble of Long Short-Term Memory (LSTM) models. Our method effectively balances exploration of the hyperparameter space with exploitation of promising configurations, addressing a critical challenge in stock price prediction.

The experiments conducted on the Nifty 50 index demonstrated that the proposed ensemble-LSTM model gives SOTA results when compared to the other classical machine learning model and standalone architectures.

Future research may focus on enhancing the robustness and accuracy of our models by incorporating more comprehensive training strategies, utilizing a larger number of particles and iterations, and leveraging improved computational resources.

References

[1] Mehar Vijh, Deeksha Chandola, Vinay Anand Tikkiwal, Arun Kumar. "Stock Closing Price Prediction using Machine Learning Techniques" Procedia Computer Science, Volume 167, 2020, Pages 599-606, ISSN 1877-0509. https://doi.org/10.1016/j.procs.2020.03.326.

[2] Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.

[3] Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220(4598), 671–680. https://doi.org/10.1126/science.220.4598.671.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

[5] Binkley, Kevin & Hagiwara, Masafumi. (2008). Balancing Exploitation and Exploration in Particle Swarm Optimization: Velocity-based Reinitialization. Transactions of The Japanese Society for Artificial Intelligence. 23. 27-35. 10.1527/tjsai.23.27.

[6] Behravan, Iman & Razavi, Seyed. (2020). Stock Price Prediction using Machine Learning and Swarm Intelligence. Journal of Electrical and Computer Engineering Innovations. 8. 10.22061/jecei.2020.6898.346.

[7] Gao, Ya, Wang, Rong, Zhou, Enmin, Stock Prediction Based on Optimized LSTM and GRU Models, Scientific Programming, 2021, 4055281, 8 pages, 2021. https://doi.org/10.1155/2021/4055281.

[8] S. Sai et al., "QGAPHnet : Quantum Genetic Algorithm Based Hybrid QLSTM Model for Soil Moisture Estimation," IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium, Athens, Greece, 2024, pp. 5191-5194, doi: 10.1109/IGARSS53475.2024.10641651.

[9] Hu Z, Zhao Y, Khushi M. A Survey of Forex and Stock Price Prediction Using Deep Learning. Applied System Innovation. 2021; 4(1):9. https://doi.org/10.3390/asi4010009.

[10] Stoean C, Zivkovic M, Bozovic A, Bacanin N, Strulak-Wójcikiewicz R, Antonijevic M, Stoean R. Metaheuristic-Based Hyperparameter Tuning for Recurrent Deep Learning: Application to the Prediction of Solar Energy Generation. Axioms. 2023; 12(3):266. https://doi.org/10.3390/axioms12030266.

[11] P. S. Sisodia, A. Gupta, Y. Kumar and G. K. Ameta, "Stock Market Analysis and Prediction for Nifty50 using LSTM Deep Learning Approach," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 156-161, doi: 10.1109/ICIPTM54933.2022.9754148.

[12] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

[13] Yakup Kara, Melek Acar Boyacioglu, Ömer Kaan Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange,Expert Systems with Applications, Volume 38, Issue 5, 2011, Pages 5311-5319, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2010.10.027.

[14] R. Majhi, G. Panda, G. Sahoo, A. Panda and A. Choubey, "Prediction of S&P 500 and DJIA stock indices using Particle Swarm Optimization technique," 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 2008, pp. 1276-1282, doi: 10.1109/CEC.2008.4630960.

[15] Sheta, A. F., Ahmed, S. E. M., & Faris, H. (2015). A comparison between regression, artificial neural networks and support vector machines for predicting stock market index. Soft Computing, 7(8), 2.