International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-4, November 2019

Timeseries Forecasting using Long Short-Term Memory Optimized by Multi Heuristics Algorithm

Hendri, Rina Novita Sari, Antoni Wibowo

Abstract*: Forecasting future price of financial instruments (such as equity, bonds and mutual funds) has become an ongoing effort of financial and capital market industry members. The most current technology is usually applied by high economic scale companies to solve the ambitious and complicated problem. This paper presents optimization solution for a deep learning model in forecasting selected Indonesian mutual funds' Net Asset Value (NAV). There is a well-known issue in determining a deep learning parameters in LSTM network like window timestep and number of neurons to be used in getting the optimal learning from the historical data. This research tries to provide solution by utilizing multi-heuristics optimization approach consists of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) to determine the best LSTM's network parameters, namely window timesteps and number of neurons. The result shows that from the nine selected mutual funds, PSO outperforms GA in optimizing the LSTM model by giving a lower Root Square Mean Error (RMSE) by 460.84% compared to GA's. However, PSO took a longer execution time by 1.78 times of GA's. This paper also confirms that based on RMSE for both training and evaluation dataset, equity mutual fund's forecasted NAV has the highest RMSE followed by fixed income mutual fund's forecasted NAV and money market mutual fund forecasted NAV.

Keywords: long short-term memory; recurrent neural network; genetic algorithm; particle swarm optimization; financial instruments prediction; mutual funds

I. INTRODUCTION

Financial instruments forecasting has been a fascinating subject ever since its establishment, while the prediction itself is both ambitious and complicated issue, the growing collected data nowadays could be a potential fuel to machine learning approach. It is the act of trying to foretell the future value of financial instruments such as mutual funds traded on a market. The efficient-market hypothesis (EMH) suggests that stock prices as one of the financial instrument reflect all currently available information and any changing price are not based on recent revealed information, so it is unpredictable [1]. A famous random walk down wall street also claimed that stock prices could not be accurately predicted by looking at price history [2], Malkiel argued, stock price are best described by a statistical process called a "random walk" meaning deviations from the central value are random and unpredictable. Malkiel concluded that paying a professional services to predict market is senseless rather than help. This is supported by the fact that most cases the portfolios managed by professional rarely outperform the market average return after deducted by the professional fees. Others disagree and those with these faiths backed them up with various methods and technologies which as appears enable them to predict future price.

A prediction methodology for most financial instruments like stock falls into three common categories, they are fundamental analysis, technical analysis and technological methods. Fundamental analysis is derived on the belief that company requires capital to make further progress and if the company operates well, it should be rewarded with additional capital and increased demand of the company stock thus increase the stock's value. Technical analysis is more concerned on the trends of the price history which form a time-series analysis. There are various techniques are used such as exponential moving average (EMA), head and shoulders, candle stick patterns and many more [3]. With the recently developed method utilizing the technological advance, machine learning are heavily utilized. An Artificial Neural Network (ANN) has been used and demonstrating its capability of addressing complex problems on several areas. ANN approach may be promising to improve investor's forecasting ability. Multivariate analytical and techniques using both quantitative and qualitative variables have been repeatedly used [4] to assist the basis of investor stock price expectations, as well as influence investment decision making. Stock price prediction itself is considered as a timeseries data. There are several approaches that normally implemented on timeseries data, one of them is Recurrent Neural Network (RNN), specifically Long Short-Term Memory (LSTM) networks, this network by most is claimed more appropriate for stock prediction as it is a time-series data. According to [5] Recurrent Neural Networks (RNNs), particularly those using Long Short-Term Memory (LSTM) hidden units, are powerful and increasingly popular models for learning from sequence data.

Manuscript published on November 30, 2019.
* Correspondence Author
Hendri, Master's degree, computer science, Nusantara University
Antoni Wibowo*, Associate Professor, Department of Decision Sciences, School of Quantitative Sciences , Universiti Utara Malaysia (UUM).
Rina Novita Sari , Bachelor of Engineering, Civil Engineering, University of Indonesia

The experiment done by Siyuan Liu [6] produce accuracy rate of the single layer LSTM by 0.66 and consequently add another layer, by three-layer LSTM model up to 0.72 for the short period of data. Thus, the more stack layers of LSTM model, the higher accuracy of prediction results, and it was believed as its necessary for LSTM network to be combined with existing clustering techniques to gain significant speed ups in training and testing at minimum loss in performance.In recent years, there have been increasing attempts to apply deep learning techniques to stock market prediction. Deep learning is a generic term for an ANN with multiple hidden layers between the input and output layers. They have been attracting significant attention for their excellent predictability in image classification and natural language processing (NLP). Deep belief network (DBN), convolution neural network (CNN), and recurrent neural network (RNN) are representative methodologies of deep learning. In particular, RNN is mainly used for time series analysis, because it has feedback connections inside the network that allow past information to persist, and time series and nonlinear prediction capabilities. Conventional ANNs do not take the "temporal effects" of past significant events into account. The temporal representation capabilities of RNN have advantages in tasks that process sequential data, such as financial predictions, natural language processing, and speech recognition. Traditional neural networks cannot handle this type of data effectively, which is one of their major weaknesses. This study intends to overcome this limitation by applying RNN to stock market predictions. We adopt long short-term memory (LSTM) units for sequence learning of financial time series data. LSTM is a state-of-the-art unit of RNN, and RNN composed of LSTM units is generally referred to as "LSTM networks". They are one of the most advanced deep learning algorithms, but less commonly applied to the area of financial prediction, yet inherently appropriate for this domain.

Given the use of LSTM network as a powerful tool in time series and pattern recognition problems, the use of an LSTM network has several drawbacks. Firstly, neural network models, like LSTM networks, suffer from a lack of capacity to clarify the final decision taken by models. Models of the neural network have a highly complex computational method that can achieve a popular solution for solving the target problem. We can not, however, offer precise explanations for the outcomes of their prediction. To prevent this issue, Kim Proposed a hybrid approach to genetic algorithm (GA) and ANN integration and rules derived from the prediction model of bankruptcy [7].

Furthermore, like other neural network models, the LSTM network has many parameters that the researcher requires to be change, such as the number of layers, neurons per layer and the number of time steps. Nevertheless, constraints in time and computation make it impossible to comb through a space parameter and find the optimum set of parameters. The determination of these control parameters in previous research has been heavily dependent on the researchers ' experience. Despite its significance, the analysis of optimal parameters for LSTM networks is limited. Accordingly, A hybrid approach integrating long short-term memory (LSTM) network and genetic algorithm (GA) [8] to estimate the time window size and architectural factors of LSTM network which comprehensively handle these aspects of LSTM models that significantly affect the performance of stock market prediction models. This study proposes two hybrid models that integrates LSTM network with GA and PSO to search for a suitable model for predicting of the financial instruments such as mutual fund's net asset value (NAV) of the three well-known types of mutual funds, they are: stock, fixed income and money market. We emphasis using GA and PSO to evaluate the architectural factors associated with the detection of a given dataset's temporal patterns, such as the time window size and number of LSTM units in hidden layers. In general, when designing the LSTM network, detecting the appropriate size for the time window that can contain the dataset background is a critical task. If the time window is too small, important signals may be missed, while unfitting data can behave as noise if the time window is too large. Regarding the investigation of the time window of RNN, many studies have suggested general approaches based on statistical methods or trial and error, along with various heuristics. We apply and compare using both GA and PSO to obtain the least root mean square error (RMSE) by using parameters of window size and number of units with the best optimum solution derived from GA and PSO. We tested our method on the selected mutual funds and finds that PSO able to outperform GA in minimizing the RMSE by 460.84% but took longer in the training execution by 177.67%. We also find that the predictability of the mutual funds derived by the product's composition confirmed by each RMSE, they are in order by the highest RMSE: 1. Stock based, 2. Fixed-income, and the least RMSE that provide more predictable return is 3. Money-market based.The remainder of this paper is organized as follows: the second section is related work, the third is theoretical literature, then follows by the methodologies that are used in this study and section 4 presents the experimental results and fifth summarizes the findings and provides suggestions for further research.

II. RELATED WORKS

A. LSTM Forecasting on Timeseries Data

A study compared three neural network models, time delay, recurrent, and probabilistic neural networks, and employed training methods of conjugate gradient and multi-stream extended Kalman filter for time delay neural network (TDNN) and RNN for stock trend prediction. RNN showed the best performance among other models [9]. LSTM with adaptive moment estimation (ADAM) optimizer achieve RMSE to 0.00859, 0.62 accuracy for Standard and Poor (S&P) 500 index and 0.83 for foreign exchange (FOREX) [10]. An addition of layer to the LSTM network improves accuracy by 18% to predict closing stock price [11], A comparison shown that LSTM model beat Back Propagation (BP) neural network on the accuracy rate on stock price forecasting by 60-65% [12].

Most of literature study above shows that LSTM network is suitable model for time-series data, because it makes use of a memory in the network. Having a memory in the network is useful because when dealing with sequenced data.

The initial version of LSTM block includes cell or possibly multiple cells, input and output gates, but no forget gate and no peephole connections. The output gate, unit biases, or input activation function were omitted for certain experiments. Training was done using a mixture of Real Time Recurrent Learning (RTRL). Only the gradient of the cell was propagated back through time, and the gradient for the other recurrent connections was truncated. Thus, that study did not use the exact gradient for training. The very first paper to suggest a modification of the LSTM architecture and to introduce the forget gate [13], enabling the LSTM to reset its own state. And like other DL architecture, there is no guidance or best practice on setting up the parameters like window size and number of units on hidden layers, therefore some researchers combine hybrid approach as an approach to tackle the parameters issue.

B. LSTM Forecasting with GA Approach

A research solves various configurations to construct forecasting models for short to medium term aggregate load forecasting by training several linear and non-linear machines learning algorithms and picking the best as baseline, choosing best features using wrapper and embedded feature selection methods and finally using genetic algorithm (GA) to find optimal time lags and number of layers for LSTM model predictive performance optimization [14], The accuracy measured by RMSE is 0.61% for the short term and an average of 0.56% for the medium term. Another hybrid approach of LSTM and meta-heuristic GA [8] suggesting a systematic approach to determine the time window size and the network topology shows better prediction performance and statistical significance as against their benchmark model, measured by mean absolute percentage error (MAPE), mean absolute error (MAE) and MSE. LSTM and GA approach is also being applied electricity load forecasting and performed better than the standard LSTM by 5.38% to 53.33% minimizing the MAPE of the training data [15], though in the future work discussion emphasis the usage RMSE instead of MAPE is more suitable measurements to be adopted.

C. LSTM Forecasting with PSO Approach

PSO with gradient descent (GD) were used to perform the trend following of electromagnetic radiation intensity data sampled from a coal mine and the atmospheric particulate PM2.5 matter data from the US Embassy in Beijing for safety forecast [16]. The improved PSO-LSTM model is compared with the conventional LSTM and the integrated moving average autoregressive model (ARIMA). The results show that compared with the standard PSO, the improved PSO has a faster convergence rate and can improve the prediction accuracy of the LSTM model effectively [17].

Table- I: Summary of Related Works

No	Author	Approach	Input	Objective
1	[9]	LSTM on time series data	Stock trend prediction	Comparative study of stock trend prediction
2	[10]	LSTM with ADAM	S&P 500	Improves accuracy
3	[11]	Adding more layers on LSTM	Closing stock price	Improves accuracy
4	[12]	Comparison of LSTM and BP Neural Net	Stock price forecasting	Comparison of accuracy
5	[13]	Introduce forget gate on LSTM	Embedded Rebber Grammar (ERG) benchmark problem	Solving the continual problem
6	[14]	LSTM and GA to find optimal time lags and number of layers	Electric Load Forecasting	Improves accuracy for both short term and medium term
7	[8]	LSTM and GA to determine the time window size and the number of units	daily Korea Stock Price Index (KOSPI) data	Improve performance, measured by MSE, MAE and MAPE
8	[15]	LSTM and GA	electricity load	minimizing the MAPE of the training data
9	[16]	LSTM and PSO	PM2.5 matter data from the US Embassy in Beijing	perform the trend following of electromagnetic radiation for safety forecast
10	[17]	LSTM and PSO	Nickel metal's closing prices in London Metal Exchange	improved a faster convergence rate and improve the prediction accuracy

III. THEORY AND METHODS

A. LSTM

LSTM network is a type of LSTM deep RNN system. RNN is a network of deep learning with internal neuronal feedback. Such internal feedbacks require the memorization and integration of knowledge of significant past events [18]. In contrast to a traditional fully connected feedforward network, RNN shares parameters across all parts of a model so that it can be generalized to sequence lengths. Fig 1 presents an example of RNN architecture that produces an output at every time step and has recurrent connections among hidden neurons.

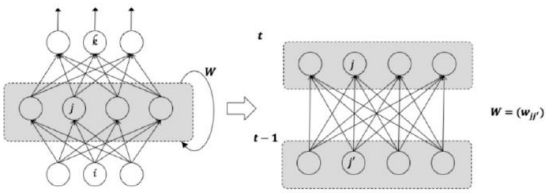

Fig 1. A simple recurrent neural network (RNN)

The RNN has weight matrices u that link the matrix of input-to-hidden weight W, which connects hidden-to-hidden, and a matrix of weight V that connects hidden-to-output. Proceeds with forward propagation by specifying the hidden unit *j0*'s initial state. Then we apply the following update for each time stage from *j0*.

The input value of hidden neuron *h* at time *t* is given as

$$h_j^t = \sum_i u_{ji} x_i^t + \sum_{j'} w_{jj'} z_{j'}^{t-1}$$

where u_{ji} is the weight between input neuron *i* and hidden neuron *j*, and x_i^t is input value at time *t*. *wjj'* denotes the weight between hidden neuron *j* and *j'*, and $z_{j'}^{t-1}$ is output value of hidden neuron *j'* at time *t* – *1*.

The transfer function of hidden neuron is named *f*, and the output of hidden neuron is expressed as

$$z_j^t = f(h_j^t)$$

Finally, the output value of the hidden layer *z* is fed into output neuron *k*, and the output value of output layer is given as

$$S_k^t = \sum_j v_j z_{j'}^t$$

Where v_j is the weight between hidden and output neurons

However, RNN has difficulty in learning long time-dependencies that are more than a few time steps in length. As the number of time steps to consider increases, information from the past events exponentially disappears [19]. LSTM is proposed LSTM is proposed to overcome the long-term dependency problem. LSTM networks can contain past information of more than 1000-time steps [18].

LSTM can scale to much longer sequences than simple RNN, overcoming the intrinsic drawbacks of simple RNN, i.e., vanishing and exploding gradients. Today, LSTM is widely used in many sequential modeling tasks [20], including speech recognition [21], motion detection [22], and natural language processing [23]. The LSTM block contains memory cell and three multiplicative gating units; an input, an output, and a forget gate [13]. There are recurrent connections between the cells, and each gate provides continuous operations for the cells. The cell is responsible for conveying "state" values over arbitrary time intervals, and each gate conducts write, read, and reset operations for the cells.

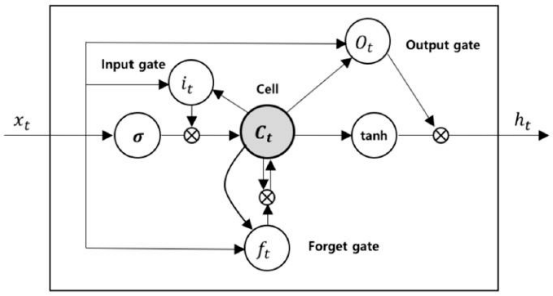

Fig 2. Long short-term memory (LSTM) cell with gating units

The computation process within an LSTM block is as follows. The input value can only be
preserved in the state of the cell if the input gate permits it. The input value of i_t and the candidate value of the memory cells, $\tilde{C}_t$, at time step, *t*, is calculated as follows:

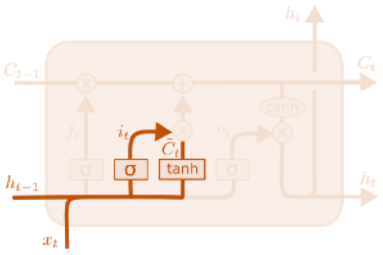

Fig 5. Input gate

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C.[h_{t-1}, x_t] + b_C$$

where *W*, $[h_{t-1}, x_t]$, *b* represents the weight matrices and bias, respectively.

The weight of the state unit is managed by the forget gate and the value of forget gate is computed as

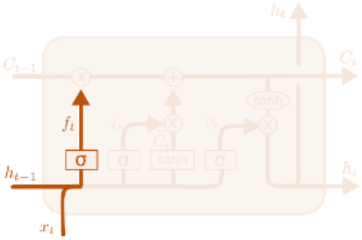

Fig 6. Forget gate

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f$$

Through this process, the new state of memory cell is updated as

$$C_t = i_t.\tilde{C}_t + f_t.\tilde{C}_{t-1}$$

With the new state of memory cell, the output value of the gate is calculated as follows:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

The final output value of cell is defined as

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * tanh(C_t)$$

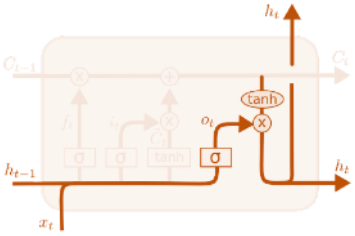

Fig 7. Output gate

B. Genetic Algorithm

Since LSTM network uses past information during the learning process, a suitably chosen time window plays an important role in the promising performance. If the window is too small, the model will neglect important information, while, if the window is too large, the model will be overfitted on the training data. In this study, we propose hybrid approaches of LSTM network with both GA and PSO to find the optimum time window and number of LSTM units for mutual funds NAV prediction.

GA is metaheuristic and stochastic optimization algorithm inspired by the process of natural evolution [24]. They are widely used to find near-optimal solutions to optimization problems with large search spaces. An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocates reproductive opportunities in such away that those chromosomes which represent a better solution to the target problem are given more chances to "reproduce" than those chromosomes which are poorer solutions. The "goodness" of a solution is typically defined with respect to the current population [25]. The process of GA includes operators that imitate natural genetic and evolutionary principles, such as crossover and mutation. The major feature of GA is the "chromosomes" group. Each chromosome acts as a potential solution to a problem with the target and is usually expressed as binary strings. These chromosomes are randomly generated and the one that provides the best solution gets more chance of reproduction. It is possible to divide the GA process into six stages: initialization, fitness calculation, termination condition check, selection, crossover, and mutation, as shown in Fig 8. In the initialization stage, a chromosome in the search space is arbitrarily selected, and then the fitness of each selected chromosome is calculated in accordance with the predefined fitness function. The fitness function is a concept used to numerically encode a chromosome's performance.

The concept of a fitness function in optimization algorithms, such as GA, is a crucial factor influencing results. Also, solutions with excellent performance are retained for further processes of replication through the process of measuring the fitness for the fitness function. Through the selection process, some chromosomes are selected many times, and chromosomes that disappear without selection are produced because they are stochastically selected according to fitness function adaptability. That is, prominent chromosomes are more likely to be inherited by the next generation. Selected superior chromosomes produce offspring by changing the respective parts of the string and the gene combinations. The crossover method results in the creation of new ideas from existing ones. One of the chromosomes is selected in the mutation process to change a randomly selected bit. The aim of this method is to add variety and innovation by random switching or turning off solution bits into the solution pool. The crossover method has the drawback that it is not possible to generate completely new data. Nonetheless, by modifying the corresponding bits to completely new values, these limitations can be solved by the mutation process.

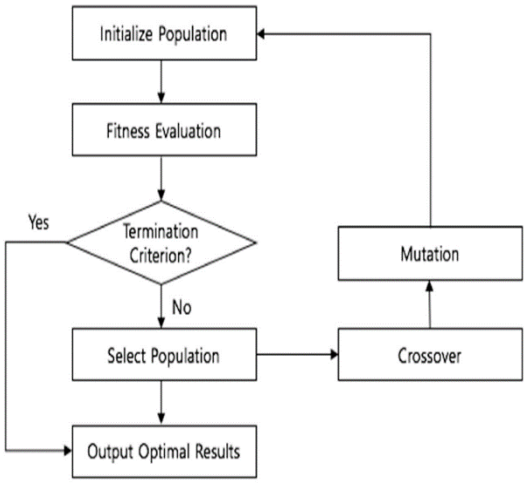

Fig 8. Process of genetic algorithm (GA)

By selection, crossover, and mutation processes, the newly generated chromosome calculates the model's fitness and verifies the termination criteria. When the termination conditions have been met, GA's standard procedure is over. If some termination conditions are not met, the processes of selection, crossover, and mutation are repeated to produce a higher-performing superior chromosome. In this study, chromosomes are represented as binary arrays and the RMSE of the prediction model is acting as the fitness value.

C. *Particle Swarm Optimization*

Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming [26].

Particle swarm optimization (PSO) shares many similarities with evolutionary computation approaches such as Genetic Algorithms (GA). The framework is initialized by updating generations with a population of random solutions and searches for optimum. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization [27], artificial neural network training [28], fuzzy system control [29], and other areas where GA can be applied. Following is the process of particle swarm optimization.

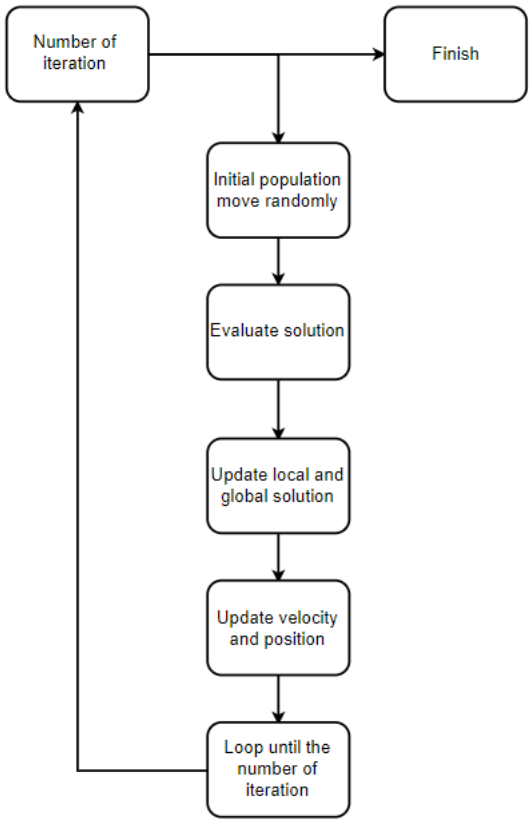

Fig 9. Process of particle swarm optimization (PSO)

A collection of random particles (solutions) initializes PSO and then looks for optima by updating generations. -particle is modified in each iteration by following two "best" values. The first is the best (fitness) approach that has been found so far. This value is called *pbest*.

Another "best" value the particle swarm optimizer tracks is the best value obtained so far by any particle in the population [30]. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*. After finding the two best values, the particle updates its velocity and positions with following equations.

Equation (a):

*v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[])*

Equation (b):

present[] = present[] + v[]

where *v[]* is the particle velocity, *present[]* is the current particle (solution). *pbest[]* and *gbest[]* are defined as stated before. *rand()* is a random number between (0,1). *c1*, *c2* are learning factors. usually *c1 = c2 = 2*.

The pseudo code of the procedure is as follows:

```
For each particle
    Initialize particle
End

Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the best fitness value (pBest) in history
            set current value as the new pBest
    End
    Choose the particle with the best fitness value of all the particles as the gBest
    For each particle
        Calculate particle velocity according equation (a)
        Update particle position according equation (b)
    End
While maximum iterations or minimum error criteria is not attained
```

Particles' velocities on each dimension are clamped to a maximum velocity *Vmax*. If the sum of accelerations would cause the velocity on that dimension to exceed *Vmax*, which is a parameter specified by the user. Then the velocity on that dimension is limited to *Vmax*.

The process for sharing information in PSO is significantly different compared to genetic algorithms (GAs). Chromosomes share information among themselves in GAs. So, the entire population is heading towards an optimum region like a single group. In PSO, the information is only given to others by *gBest* (or *lBest*). It's a one-way mechanism for sharing information. Evolution searches only for the best solution. Compared to GA, all particles appear to converge rapidly even in the local version in most cases to the best solution [31].

IV. PROPOSED METHODS

This study methodological workflow is presented in Fig 2. It starts from data collection, then followed by data preprocessing, GA feature selection, LSTM architecture, GA parameters optimization, Evaluation and Reporting. The similar procedure is used for PSO optimization.

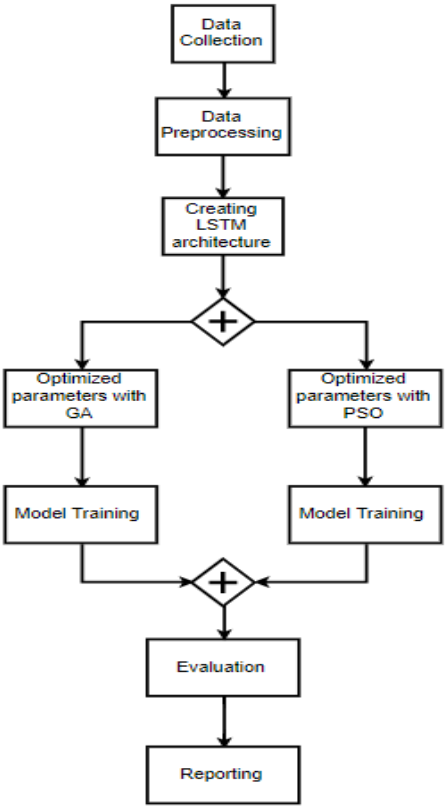

Fig 10. Methodological workflow

A. *Data collection*

The financial asset chosen for this study are Indonesian based equity, fixed income and money market mutual funds.

We collect the mutual funds' historical Net asset value (NAV) for the period more than 10 years. NAV represents a fund market value per investor's unit share. NAV is calculated by subtracting the total value of all the cash and securities in a fund's portfolio by any liabilities then dividing the result by the total number of outstanding shares

Equity mutual funds are considered the riskiest among the three because their portfolio consist of more than 80% allocation in equity. Fixed mutual funds are considered the medium risk mutual funds because more than 80% of their portfolio consists of bond while money market mutual funds are considered the least risky mutual fund because more than 80% of their portfolio consists of time deposit and less than one-year mature bond.

B. *Data Preprocessing*

To avoid bias we make sure to use only the best performed mutual funds in term of return and risk in this research. We use algorithm to filter the mutual funds using returns, performance ratios (Sortino ratio, Sharpe ratio and Information ratio) and asset under management (AUM) as tools. The stability of long term returns, most current return and performance ratios and the size of AUM are scored with different weights. The performance ratios used are sharpe ratio, sortino ratio and information ratio.
Sortino ratio's:

$$sortino\ ratio = \frac{R_p - rf}{\sigma d}$$

Where:
R_p = actual or expected portfolio return
rf = risk- free rate
σd = standard deviation of the downside

Sharpe ratio's :

$$sharpe\ ratio = \frac{R_p - rf}{\sigma p}$$

Where:
R_p = return of portfolio
rf = risk- free rate
σd = standard deviation of the portfolio's excess return

and information ratio (IR):

$$IR = \frac{portfolio\ return - benchmark\ return}{tracking\ error}$$

Where tracking error is the standard deviation of difference between portfolio and benchmark returns.
We use Jakarta Composite Index (JCI) as the benchmark for equity mutual fund, Indonesia Bond Pricing Agency's Composite Bond Index and Bank of Indonesia (BI) daily interest rate as the benchmark for fixed income mutual fund and one-month time deposit rate as the benchmark for money market mutual fund.

C. *LSTM Architecture*

We use the following model and LSTM's parameters such as window size and the number of units to be determined by GA and PSO. The experiments implements KERAS, the python deep learning library [32] as a framework.

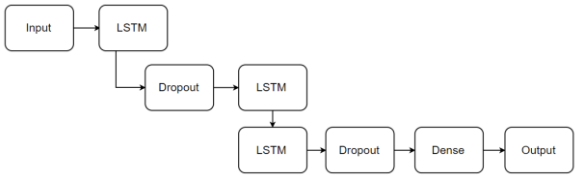

Fig 11. LSTM Architecture

*D. Optimized Parameters with GA*The GA configuration are 5 as the population size and the number of generations, and 10 as the length of gene. A solution to our problem is a 10-bit integer where the first six bits represent the window size and the next four bits represent the number of units of LSTM.

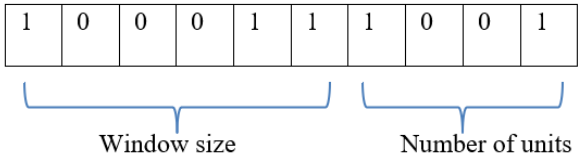

Fig 12. Gene representation

The complete architecture optimized by GA is shown in Fig 13. DEAP is being used for this research as it over comprehensive framework to implement evolutionary algorithm [33].

E. Optimized Parameters with PSO

The PSO configuration we use is both 5 as swarm size and maximum iteration, and we set the both lower and the lower bond is 1 to 31. The complete architecture optimized by PSO is shown in fig 14. Pyswarm is being used for this research as to implement PSO algorithm.

F. Hardware specification

The hardware specification is as following:
Processor: AMD Ryzen Threadripper 2920X 12-Core Processor, 3500 Mhz, 12 Cores, 24 Logical Processors
Installed Physical Memory (RAM): 32 GB
Total Virtual Memory: 70.6 GB
Page File Space: 38.8 GB
OS: Microsoft Windows 10 Pro
Platform Role: Desktop

V. RESULTS AND DISCUSSION

In summary, LSTM optimized with PSO outperforms GA by 460.84%, but took longer time to execute by 177.67 times of GA.

Table- II: Summary of Evaluation results

	RMSE	Execution time (s)
LSTM + GA	0.06378703	515.96594299
LSTM + PSO	0.01384161	916.72544617

And the following RMSE for the mutual funds based on type of product shown that the risk for these products in-line with the predictability of its NAV.

Table- III: Summary of Evaluation results

No.	Mutual funds	RMSE	
		GA	PSO
1.	Equity	0.09044404	0.02493967
2.	Fixed-income	0.04091847	0.00915334
3.	Money market	0.05999860	0.00743181

From the result, PSO may achieve better results depending on the number of maximum iterations to be set, while GA has no definitive number of iterations. This research has not implemented the parallel processing provided by DEAP and we may gain significant amount of processing time by implementing it, as well as to replace the Pyswarm framework with DEAP supported with multiswarm PSO.

VI. CONCLUSION

The accuracy of price prediction affects financial and capital market companies' profit and loss in a significant value. Time series is one the most frequent techniques used for price prediction. Therefore, improving time series model predictability power has been the focus of several studies. Deep learning model has been studied and tested actively in recent years because of its extraordinary predicting power. This study suggests useful consequences from developing a proper LSTM network architecture whose network factors have been optimized. The optimized architecture has more predicting power in temporal pattern detection.

Time window plays a very significant role in analyzing the temporal properties of a given dataset. Thus, defining the suitable time window for each problem function is very critical. Unfortunately, for each LSTM network process, time window isn't automatically fine-tuned for solving the problem function. The consequence is the system may not recognize the most important pattern in the dataset. Most of the existing literatures, which use LSTM network in time series problems, usually uses trial and error-based subjective approaches rather than systematic approaches to find the optimal time window length. While the hybrid multi-heuristics model proposed has a prominent predictive performance, it still has some insufficiencies. There is room for improvements in future research that comes from setting the control parameters for both GA and PSO optimization such as crossover rate, mutation rate, swarm size and maximum iteration. More optimized parameters used may improve the predictive power of this multi-heuristics optimized LSTM architecture. Alfahq, a company who develops an AI based robo-advisor application, has applied and made further improvements on this study for the back-engine of an Indonesian based robo-advisor named Kina.

REFERENCES

1. E. F. Fama and K. R. French, “Efficient capital markets: a review of theory and empirical work,” J. Finance, vol. 47, pp. 383–417, 1992.
2. B. Malkiel, A random walk down Wall street. 2013.
3. R. D. Edwards, J. Magee, and W. H. C. Bassetti, Technical Analysis of Stock Trends. 2012.
4. Y. Yoon and G. Swales, “Predicting stock price performance: A neural network approach,” in Proceedings of the Annual Hawaii International Conference on System Sciences, 1991.
5. Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, “Learning to Diagnose with LSTM Recurrent Neural Networks,” no. November, 2015.
6. S. Liu, G. Liao, and Y. Ding, “Stock transaction prediction modeling and analysis based on LSTM,” Proc. 13th IEEE Conf. Ind. Electron. Appl. ICIEA 2018, pp. 2787–2790, 2018.
7. J. K. Kim, K. C. Lee, and J. K. Lee, “Hybrid of neural network and decision knowledge approach to generating influence diagrams,” Expert Syst. Appl., 2002.
8. H. Chung and K. S. Shin, “Genetic algorithm-optimized long short-term memory network for stock market prediction,” Sustain., vol. 10, no. 10, 2018.
9. E. W. Saad, D. V. Prokhorov, and D. C. Wunsch, “Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks,” IEEE Transactions on Neural Networks, vol. 9, no. 6. pp. 1456–1470, 1998.
10. S. Minami, “Predicting Equity Price with Corporate Action Events Using LSTM-RNN,” J. Math. Financ., 2018.
11. T. Gao, Y. Chai, and Y. Liu, “Applying long short term momory neural networks for predicting stock closing price,” Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS, vol. 2017-Novem, pp. 575–578, 2018.
12. Y. Wang, Y. Liu, M. Wang, and R. Liu, “LSTM Model Optimization on Stock Price Forecasting,” Proc. - 2018 17th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. DCABES 2018, pp. 173–177, 2018.
13. F. A. Gers, J. Schmidhuber, and F. Cummins, “Continual Prediction using LSTM with Forget Gates,” 1999.
14. S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, “Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches,” Energies, 2018.
15. A. S. Santra and J. L. Lin, “Integrating long short-term memory and genetic algorithm for short-term load forecasting,” Energies, 2019.
16. Y. Hu, X. Sun, X. Nie, Y. Li, and L. Liu, “An Enhanced LSTM for Trend Following of Time Series,” IEEE Access, vol. 7, no. c, pp. 34020–34030, 2019.
17. B. Shao, M. Li, Y. Zhao, and G. Bian, “Nickel Price Forecast Based on the LSTM Neural Network Optimized by the Improved PSO Algorithm,” Math. Probl. Eng., 2019.
18. S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.
19. R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in 30th International Conference on Machine Learning, ICML 2013, 2013.
20. Y. Yao and Z. Huang, “Bi-directional LSTM recurrent neural network for chinese word segmentation,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9950 LNCS, pp. 345–353, 2016.
21. H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, 2014.
22. A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social LSTM: Human trajectory prediction in crowded spaces,” in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.
23. Melamud, J. Goldberger, and I. Dagan, “context2vec: Learning Generic Context Embedding with Bidirectional LSTM,” 2016.
24. J. H. Holland, Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. 1975.
25. D. Whitley, “A genetic algorithm tutorial,” Stat. Comput., 1994.
26. T. Ma, C. Wang, J. Wang, J. Cheng, and X. Chen, “Particle-swarm optimization of ensemble neural networks with negative correlation learning for forecasting short-term wind speed of wind farms in western China,” Inf. Sci. (Ny)., 2019.
27. S. Mirjalili and S. Z. M. Hashim, “A new hybrid PSOGSA algorithm for function optimization,” in Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application, 2010.
28. V. G. Gudise and G. K. Venayagamoorthy, “Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks,” in 2003 IEEE Swarm Intelligence Symposium, SIS 2003 - Proceedings, 2013.
29. Y. Zhang, S. Wang, and G. Ji, “A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications,” Mathematical Problems in Engineering. 2015.
30. D. P. Rini and S. M. Shamsuddin, “Particle Swarm Optimization: Technique, System and Challenges,” Int. J. Appl. Inf. Syst., 2011.
31. R. C. Eberhart and Y. Shi, “Comparison between genetic algorithms and particle swarm optimization,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1998.
32. F. Chollet, “Keras: The Python Deep Learning library,” Keras.Io, 2015.
33. F. M. De Rainville, F. A. Fortin, M. A. Gardner, M. Parizeau, and C. Gagné, “DEAP: A Python framework for Evolutionary Algorithms,” in GECCO’12 - Proceedings of the 14th International Conference on Genetic and Evolutionary Computation Companion, 2012.

AUTHORS PROFILE

Hendri received his bachelor's degree of Computer science in 2005 from Bina Nusantara University and currently pursuing a master's degree in computer science from Bina Nusantara University since 2018. He has been very fascinated in Artificial Intelligence and Data mining topics as it has been a tremendously beneficial to his position as Director at PT. Mitra Konsultansi Indonesia (MKINDO), a registered Independent Software Vendor (ISV) partner of Microsoft providing cutting edge web and mobile and AI solutions to their clients. With more than 15 years of software development services, he has been repeatedly recognized for developing innovative solutions for large-scale industry such as Oil and Gas, Construction and Heavy Industry. He has several professional IT certifications such as Microsoft Certified Professional Developer (MCPD), Red Hat Certified Engineer (RHCE), Professional Scrum Master -1 (PSM-1), and Certified Information Technology Infrastructure Library (ITIL) Practitioner in IT Service Management.

Rina Novita Sari is a capital market professional and a co-founder of Alfahq. She received a Bachelor of Engineering in Civil Engineering from University of Indonesia in 2002 and a Master Degree in Management from Gadjah Mada University in 2004. In 2003, she was awarded a scholarship for an Exchange Student program from Association of International Exchange Japan (AIEJ) to deepen her capital market and financial education in International University of Japan. In 2019, she finished Data Science Academy in Algoritma Data Science School, Jakarta where she studied Data Science and Machine Learning. She finished the academy by creating a machine learned-based apps that predicts Indonesia stocks' price and performs a back testing profit simulation model. She has 15 years' experience as a fund manager and holds a license as Investment Manager Representative from Indonesia Capital Market Supervisory Agency (BAPEPAM) with a letter numbered KEP - 111/PM/WMI/2005 dated December 5th, 2005 that has been extended by Indonesia Financial Service Authority (OJK) with a letter numbered KEP-416/PM.211/PJ-WMI/2018 dated November 28th, 2018. She has passed level 1 Chartered Financial Analyst exam in 2010 and level 1 Chartered Market Technician exam in 2013.

Antoni Wibowo has received the first degree of Applied Mathematics in 1995 and a master degree of Computer Science in 2000. In 2003, He awarded a Japanese Government Scholarship (Monbukagakusho) to attend Master and PhD programs at Systems and Information Engineering in University of Tsukuba-Japan. He completed the second master degree in 2006 and PhD degree in 2009, respectively. His PhD research focused on machine learning, operations research, multivariate statistical analysis and mathematical programming, especially in developing nonlinear robust regressions using statistical learning theory. He has worked from 1997 to 2010 as a researcher in the Agency for the Assessment and Application of Technology – Indonesia. From April 2010 – September 2014, he worked as a senior lecturer in the Department of Computer Science - Faculty of Computing, and a researcher in the Operation Business Intelligence (OBI) Research Group, Universiti Teknologi Malaysia (UTM) – His PhD research focused on machine learning, operations research, multivariate statistical analysis and mathematical programming, especially in developing nonlinear robust regressions using statistical learning theory. He has worked from 1997 to 2010 as a researcher in the Agency for the Assessment and Application of Technology – Indonesia. From April 2010 – September 2014, he worked as a senior lecturer in the Department of Computer Science - Faculty of Computing, and a researcher in the Operation Business Intelligence (OBI) Research Group, Universiti Teknologi Malaysia (UTM) – Malaysia. From October 2014 – October 2016, he was an Associate Professor at Department of Decision Sciences, School of Quantitative Sciences in Universiti Utara Malaysia (UUM). Dr. Eng. Wibowo is currently working at Binus Graduate Program (Master in Computer Science) in Bina Nusantara University-Indonesia as a Specialist Lecturer and continues his research activities in machine learning, optimization, operations research, multivariate data analysis, data mining, computational intelligence and artificial intelligence.

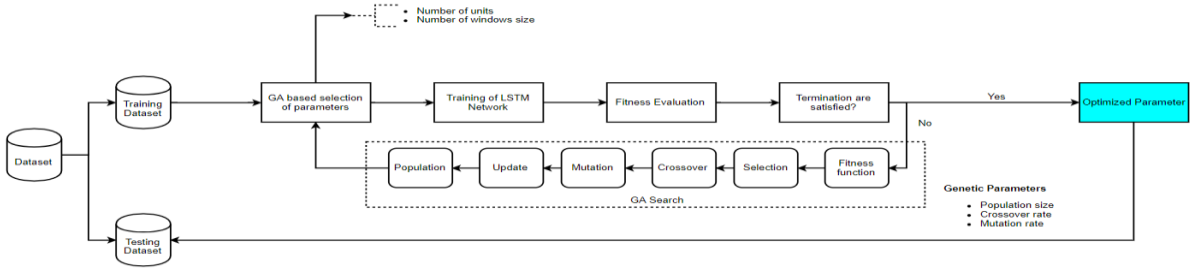

Fig 13. LSTM optimized with GA

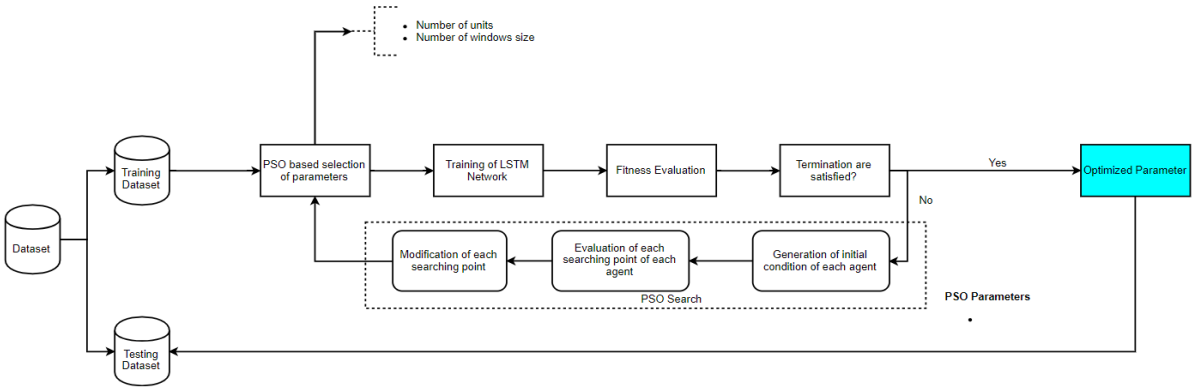

Fig 14. LSTM optimized with GA