Intelligent Stock Trading Strategy Based on Aroon Indicator

Arjun Singh Saud [1]
Subarna Shakya [2]
Bindu Neupane [3]

Abstract

The study of stock trading signal forecasting has piqued the interest of machine learning and technical analysis specialists. One of the popular tools for anticipating buy and sell signals is the Aroon indicator, but it is not utilized by machine learning researchers to predict stock trading signals. This study proposed an intelligent stock trading strategy based on the association between Aroon indicators. The performance of the proposed stock trading strategy was compared to that of a classical Aroon indicator based trading strategy in terms of annual rate of return (ARR), Sharpe ratio (SR), and percentage of gain/loss trades. In terms of all three measures, it was discovered that the intelligent trading strategy outperformed the classical trading method. The intelligent trading method generated 3.91% to 40.07% greater ARR than the classical strategy, and it did so with a positive SR for all 10 stocks studied. In addition, the intelligent approach executed a higher percentage of profitable transactions than the traditional strategy. Thus, it was established that the proposed intelligent trading method is a better and safer trading technique than the classical strategy.

Keywords : **Intelligent stock trading, Aroon indicator, trading signals, algorithmic stock trading**

Paper Submission Date : October 21, 2021 ; Paper sent back for Revision : February 9, 2022 ; Paper Acceptance Date : February 25, 2022

Stock investors want to buy and sell stocks at the proper time. When it comes to purchasing and selling stocks, investors rely majorly on their expertise and experience. Making this type of prediction, however, is difficult since stock prices are a result of numerous known and unknown variables (Ding & Qin, 2020). The known variables include company fundamental characteristics such as earnings, growth, dividend history, and dividend capacity while the unknown variables include political developments, rumors, changing regulatory policies, etc. For anticipating the direction of stock prices, financial experts use technical analysis (Nada & Igor, 2016; Nti et al., 2020), wherein they tend to examine a large volume of historical stock trading data. However, analyzing past trading data of each stock using various technical indicators and drawing a conclusion from it is a time-consuming task. Therefore, machine learning approaches have become popular for predicting stock price trends (Sarangi et al., 2019; Shah et al., 2019). Due to the incapacity of machine learning algorithms to recall the context, recurrent neural networks (RNNs) were widely used by researchers for predicting market trends. Furthermore, due to vanishing gradient issues with recurrent neural networks, gated recurrent unit (GRU),

[1] *Assistant Professor (Corresponding Author),* Central Department of Computer Science and IT, Tribhuvan University, Kathmandu, Bagmati, Nepal - 44 613. (Email : arjunsaud@cdcsit.edu.np) ;
ORCID iD : https://orcid.org/0000-0002-5235-7322

[2] *Professor,* Department of Electronics and Computer Engineering, IOE, Tribhuvan University, Lalitpur, Bagmati, Nepal - 44 700.(Email : drss@ioe.edu.np) ; ORCID iD : https://orcid.org/0000-0003-2268-0097

[3] *Lecturer,* Nagarjuna College of IT, Tribhuvan University, Lalitpur, Bagmati, Nepal, 447 00. (Email : bneupanesaud@gmail.com) ; ORCID iD : https://orcid.org/0000-0003-1855-6253

DOI : https://doi.org/10.17010/ijrcm/2022/v9i1/170401

and long short-term memory (LSTM) networks emerged as state-of-the-art models for stock price trend forecasting (Lien Minh et al., 2018; Moghar & Hamiche, 2020; Shen et al., 2018).

Many machine learning algorithms for automated stock trading have been presented (Gudelek et al., 2017; Stoean et al., 2019; Tilakaratne et al., 2009; Yang et al., 2020). For predicting stock trading signals of the Australian All Ordinary Index, Tilakaratne et al. (2020) suggested a new neural network approach with a modified error function. The authors observed that the proposed approach was able to produce superior forecasts by combining close prices of stocks from several stock exchanges as input to the algorithm. Gudelek et al. (2017) proposed a 2D-CNN-based stock trading model. Several technical indicators were used as input features to the proposed model, including the simple moving average (SMA), moving average convergence divergence oscillator (MACD), relative strength index (RSI), Williams percent range (Williams %R), ultimate oscillator, stochastic oscillator, and relative strength index (RSI). From the outcomes of experimental results, researchers discovered that the proposed model outperformed the buy-hold strategy. Stoean et al. (2019) proposed a deep learning model for anticipating buy and sell signals based on the difference between the current trading day's close price and the next day's forecasted close price. The proposed model was able to achieve enhanced annualized return and Sharpe ratio based on the experimental results of 25 stocks listed on the Romanian stock exchange. Yang et al. (2020) developed an ensemble strategy based on deep reinforcement learning. Historical stock prices, current stock holdings, and numerous technical indicators were input into the trading agent. According to the findings of the experiments, the ensemble technique outperformed individual algorithms.

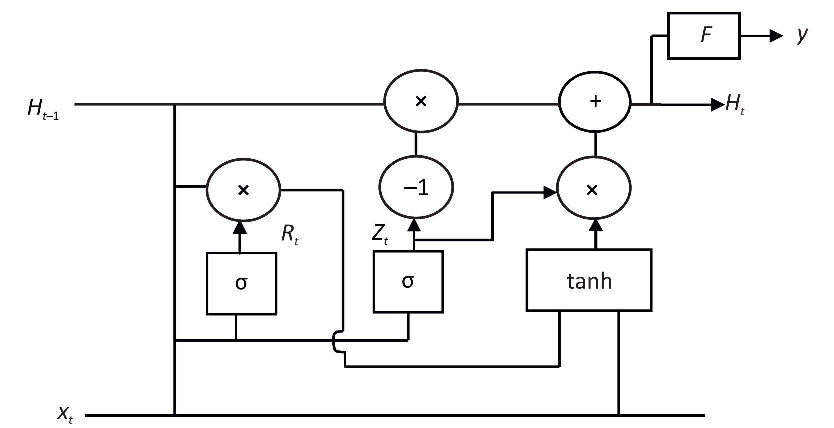
Momentum indicators, trend indicators, volatility indicators, volume indicators, and miscellaneous indicators are the five categories of technical indicators (TA Library in Python, n.d.). Many studies used these technical indicators as an input to machine learning model to predict stock prices or stock trading signals (Atsalakis & Valavanis, 2009 ; Chen & Hao, 2017; Chiang et al., 2016; Ding & Qin, 2020 ; Gao & Chai, 2018; Gudelek et al., 2017; Jabbarzadeh et al., 2016 ; Kim & Kim, 2019 ; Long et al., 2019 ; Patel et al., 2015; Rodríguez-González et al., 2011; Shynkevich et al., 2017). Although the relationship between technical indicators is frequently studied and used in the technical analysts' community for stock buying and selling, no machine learning technique based on this notion has been developed yet. Among technical analysts, the Aroon indicator is a prominent trend and momentum indicator. But to utter dismay, machine learning experts have overlooked its potential for forecasting stock trading signals. Based on this discovery, this research work developed an intelligent stock trading method for anticipating stock trading signals based on the relationship between Aroon indicators (Aroon-ML) and compared its performance with the classical Aroon indicator-based trading strategy (Classical-Aroon). The proposed Aroon-ML technique was implemented in this study using the GRU network because of its generalization capabilities with a moderate volume of data.

Gated Recurrent Unit (GRU) Network

GRU is a recurrent neural network (RNN) variant meant to solve the vanishing and exploding gradient descent problems belonging to RNNs. Another RNN variant, long short-term memory (LSTM), is also used to solve the gradient descent difficulties. However, it employs more gates and parameters than GRU and requires more data to generalize (Cho et al., 2014; Weiss et al., 2018). GRU network's architecture and mathematical formulation are provided in Figure 1 and Equation 1 (Cho et al., 2014), respectively.

$$\begin{aligned}
R_t &= \sigma\,(W_r x_t + U_r H_{t-1}) \\
Z_t &= \sigma\,(W_z x_t + U_z H_{t-1}) \qquad (1) \\
H'_t &= \tanh\,(W_h x_t + (R_t \times H_{t-1})U_h) \\
H_t &= (Z_t \times H'_t) + ((1 - Z_t) \times H_{t-1})
\end{aligned}$$

Figure 1. *GRU Architecture*



where, x is an input vector, W and U are weight matrices, H_t and $\dot{H}_t$ are hidden and candidate hidden states, and Z and R are update and reset gates, respectively.

Aroon Indicator

The Aroon Indicator is a trend indicator that technical analysts use to identify stock trend shifts and the strength of the existing trend. Aroon Up and Aroon Down are two indicators that make up this indicator. The Aroon Up indicator counts the days since the most recent high in the N-days period, whereas the Aroon Down indicator counts the days since the most recent low (*Trading Strategy Guides*, n.d.). When calculating the values of indicators, N = 25 is commonly utilized. The formula for calculating up and down indicators is given in Equation 2.

$$AroonUP = \frac{(N - d_1)}{N} \times 100$$

$$AroonDown = \frac{(N - d_2)}{N} \times 100 \qquad (2)$$

where, d_1 and d_2 are the numbers of days since the most recent high and low, respectively, in the N-days' time period.

Both indicators have a range of 0 to 100. Aroon Up evaluates the strength of a stock's bullish trend, whereas Aroon Down measures its bearish trend. Buying stocks when Aroon Up crosses above Aroon Down and selling the stocks when Aroon Down crosses above Aroon Up is the general trading strategy. In this paper, this strategy is referred to as the classical-Aroon strategy. Technical analysts also utilize the Aroon oscillator in addition to the Aroon indicators. It is calculated by subtracting the value of Aroon Up from Aroon Down. This oscillator represents the current trend's strength.

Proposed Strategy

This section covers the target feature generation model for producing Buy, Hold, and Sell signals, the Aroon-ML trading strategy for predicting stock trading signals, and the trading simulation model for performing automated stock trading using signals predicted by the Aroon-ML strategy or the classical-Aroon strategy.

Target Feature Generation

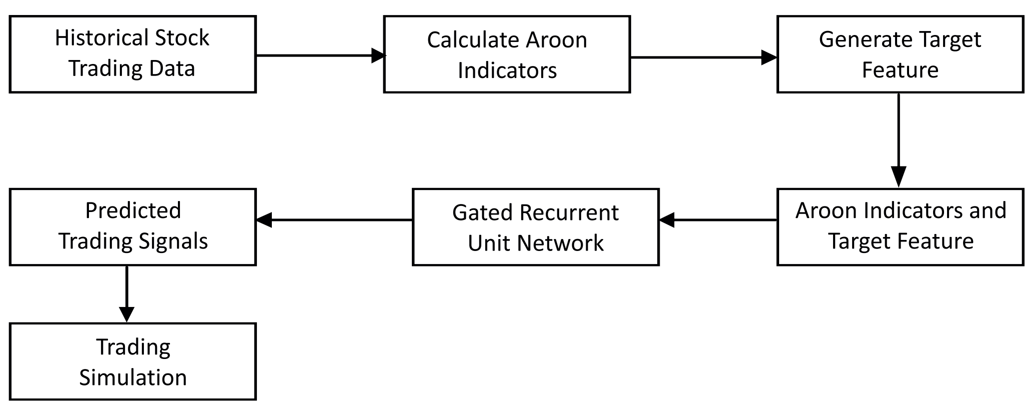
Based on the relationship between the Aroon Up and Aroon Down indicators, a model was devised for generating the target feature "Signal". Equation 3 is the mathematical formulation of the model. This model alternately generates trading signals for purchasing and selling stocks. The model pads three Buy or Sell signals in the target feature when there is a crossover between the Aroon Up and Aroon Down indicators. The purpose of this padding is to aid the machine learning model in identifying trading signals. Thus, a series of Buy, Hold, and Sell signals are generated by the target generation model.

$$\begin{array}{ll} \text{if } AroonUp > AroonDown & singal\,[k{:}k+3] = \text{'Buy'} \\ \text{if } AroonDown > AroonUp & singal\,[k{:}k+3] = \text{'Sell'} \\ else & singal\,[k] = \text{'Hold'} \end{array} \qquad (3)$$

Aroon-ML Strategy

Figure 2 depicts a conceptual framework of the proposed Aroon-ML strategy to forecast stock trading signals. The components of the proposed strategy are elaborated in the following sections :

Figure 2. ***Proposed Aroon-ML Trading Strategy***



Stock Trading Simulation

To perform automated stock trading using signals anticipated by the Aroon-ML strategy or the classical-Aroon strategy, this study constructed a trading simulator. The simulator conducts stock transactions assuming that stock investors have no stocks to begin with. Hence, Buy should be the simulator's first trading operation. Thereafter, the trading simulator alternates between Buy and Sell actions. Thus, a Buy and Sell sequence is executed by the simulator. The sequence should have the same number of both signals if the last trading activity is Sell. However, if Buy is the last trading activity, the sequence will have one more Buy action than the Sell action. In such a case, the simulator ignores the last Buy signal and calculates gross profit/loss using automatic trading data. The close price was used as the Buy/Sell price in the simulator. Equation 4 is the formula for calculating gross profit/loss from trading. The calculations of transaction costs and income gain tax were not included in this equation.

$$gpl = \sum_{i=0}^{len(s)} (s_i - b_i) \tag{4}$$

where, s and b are the vectors representing buy and sell prices of the trading sequence.

Following the profit/loss calculation, the simulator used Equations 5 to calculate the average amount invested in stocks and used Equation 6 to calculate the profit/loss percentage yielded from the Aroon-ML strategy and the classical-Aroon strategy.

$$a_i = \frac{1}{len(b)} \sum_{i=0}^{len(b)} b_i \tag{5}$$

$$plp = \frac{gpl}{a_i} \times 100 \tag{6}$$

Methodology

This study adopted an experimental research design. The historical stock trading data, data preprocessing and preparation approach, and configuration of the GRU network used to implement the proposed Aroon-ML strategy are described in this part.

Stock Data

This study experimented with historical trading data of stocks listed on the Bombay Stock Exchange (BSE) and the Nepal Stock Exchange (NEPSE). Five stocks were picked at random from both exchanges. The daily data, from 01/01/2000 to 14/11/2020, of BSE stocks were collected from BSE India (S&P BSE SENSEX Stock Prices, n.d.). Axis Bank (AXIS), Bajaj Auto Limited (BAJAJ), Nestle India Limited (NESTEL), JSW Steels Ltd. (JSW), and Sun Pharmaceutical Industries Limited (SUNP) were among the five BSE stocks experimented. Similarly, the data of NEPSE stocks were collected from Nepal Stock Exchange (*Nepal Stock Exchange Ltd,* n.d.). The data were also daily data from 15/04/2010 to 14/11/2020. Laxmi Bank Limited (LBL), Citizens Bank International Limited (CZBIL), Siddhartha Bank Limited (SBL), Arun Valley Hydropower Company (AHPC), and Butwal Power Company Limited (BPCL) were among the five NEPSE stocks experimented.

Data Pre - Processing and Preparation

Initially, the datasets were organized in chronological order. Thereafter, the Aroon indicators were calculated and all other features from the dataset were dropped. Following that, using the target generation model that has been formulated, the target feature 'Signal' was generated. Finally, one-hot encoding strategy was used to encode the target feature and a standard scalar was used to normalize the input features.

Further, the datasets were split into train, validation, and test sets in 8:1:1 ratio. The objective of this study was to predict stock trading signals for the day $t + 1$ utilizing input features from the $(t–N+1)^{th}$ day to the t^{th} day, where t represents the current trading day and N represents the window size. Thus, $t \in [1, L \times 0.8]$ for the training set, $t \in [L \times 0.8 + 1, L \times 0.9]$ for the validation set, and $t \in [L \times 0.9 + 1, L]$ for the test set, where L is the length of the dataset. Window size 5 was chosen in this study.

Configuration of GRU Network

The GRU network used in this study was configured as 3×100×100×3. A dropout layer was placed after each layer of the network. The GRU network also included an Adam optimizer as recommended by Saud and Shakya (2019). The network's hidden layers employed the ReLU activation function and the output layer used the

Softmax activation function. This GRU configuration was not fine-tuned. However, some random experiments with different configurations were undertaken, and this configuration was eventually adopted.

Tools Used

In Google Colaboratory, the trading strategies were implemented using the Python programming language. Technical analysis (TA), Sklearn, Keras, and Statistics libraries were also used in the implementation. The Technical Analysis library assists us in generating features from financial time series datasets. Many efficient machine learning and statistical modeling tools are included in the Sklearn library. Keras is an open-source application programming interface (API) for artificial neural networks. The Statistics library contains functions for calculating mathematical statistics from numerical data.

Analysis and Results

Performance of the proposed Aroon-ML approach was assessed using three metrics: (1) annual rate of return (ARR), (2) sharp ratio (SR), and (3) percentage of gain/loss transactions executed by the strategy. In addition, the proposed approach's performance was compared to that of a classical-Aroon strategy. The classification accuracy of the Aroon-ML approach was not assessed. This is because the target feature ‘Signal’ was simply an approximation of trade signals on the basis of crossover between Aroon Up and Aroon Down indicators. The Aroon-ML technique was devised to filter misleading trade signals generated by the Aroon indicator. Thus, accuracy is not a relevant metric for the proposed intelligent stock trading method.

Figure 3. ***Buy/Sell Actions for AXIS***

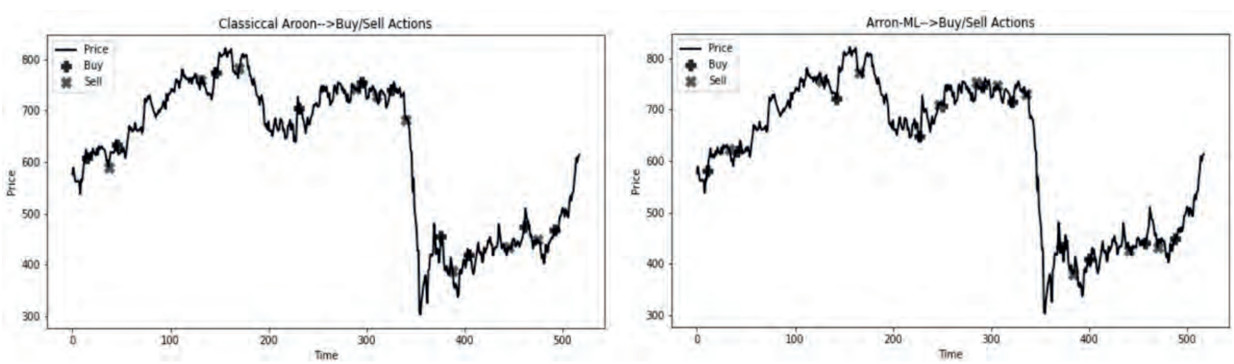
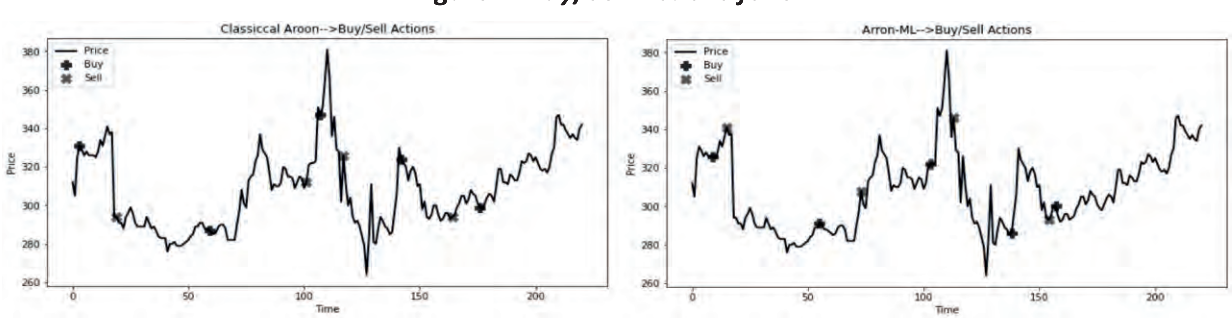


Figure 4. ***Buy/Sell Actions for SBL***



Figures 3 and 4 show the buy/sell actions generated by the proposed Aroon-ML technique and the classical-Aroon strategy for the stocks AXIS and SBL, respectively. The figure's x-axis and y-axis reflect the time (number of trading days) and price dimensions, respectively.

Analysis of Trading Returns

For the test data, a trading simulation was conducted utilizing trade signals generated by the Aroon-ML technique and the classical-Aroon strategy. The annual rate of return (ARR) and sharp ratio (SR) for each strategy were then determined using Equations 7 and 8, and the results are presented in Figures 5 and 6 correspondingly.

$$ARR = (1 + Return)^{1/n} - 1 \qquad (7)$$

Where, n is the number of years

$$SR = \frac{R_t - R_f}{\sigma} \qquad (8)$$

where, R_t is the return yielded from stock trading, R_f is the risk-free return, and σ is the standard deviation of R_t.

As can be observed in Figures 5 and 6, the performance of the Aroon-ML trading strategy is superior to that of the classical-Aroon strategy. For all the stocks, the Aroon-ML strategy yielded the highest annual return. The approach generated an ARR of 11.1% to 141.91 %, with a positive SR. The classical-Aroon strategy, on the other hand, consistently delivered relatively lower ARR. The ARR for the approach ranged from –18.09% to 128.29%. Only three stocks in the strategy attained the above-mentioned ARR with a positive SR. It has a negative SR for the remainder of the seven stocks. In summary, the Aroon-ML technique generated a 35.12% average ARR with a 1.89 average SR value, while the classical-Aroon strategy generated 11.98% average ARR with a –0.66 average SR value. This demonstrates that the Aroon-ML strategy is more profitable and safer than the classical-Aroon strategy.

Figure 5. ***Annual Rate of Return (ARR)***

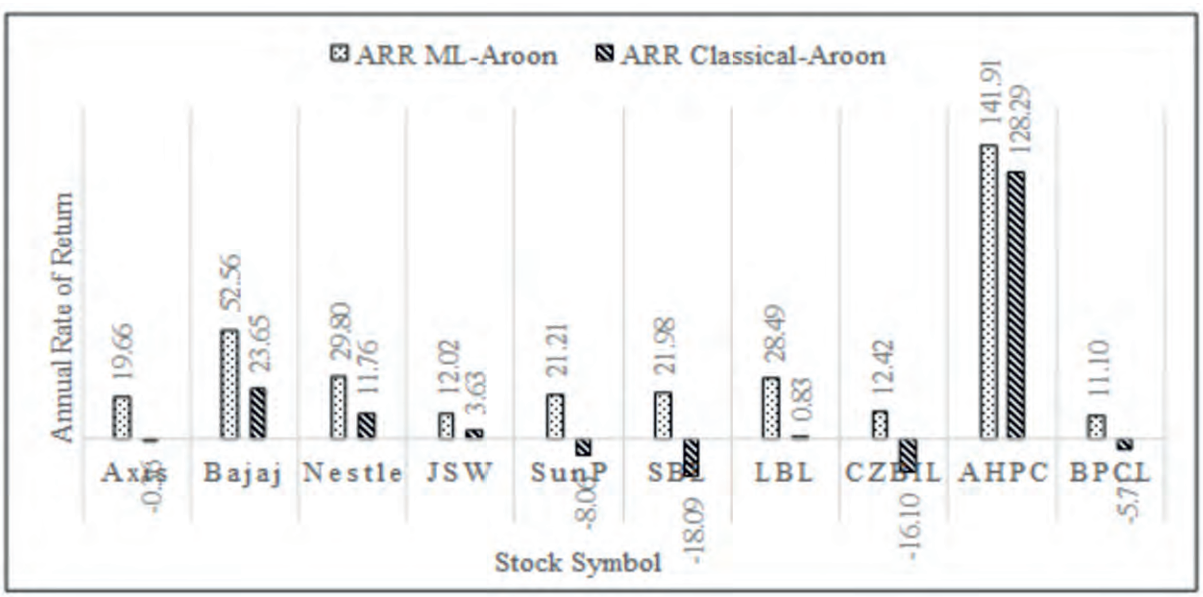
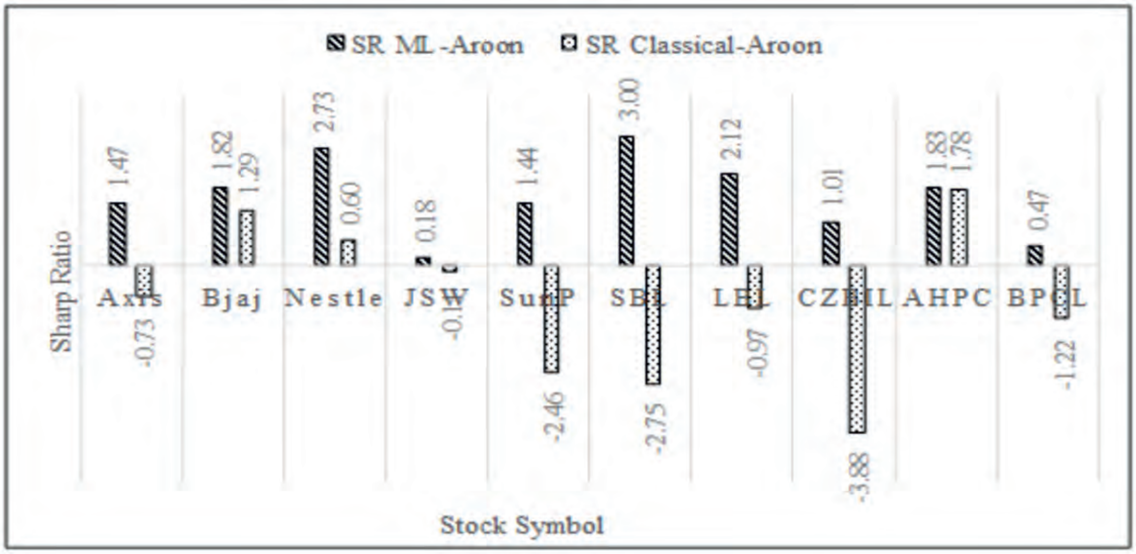


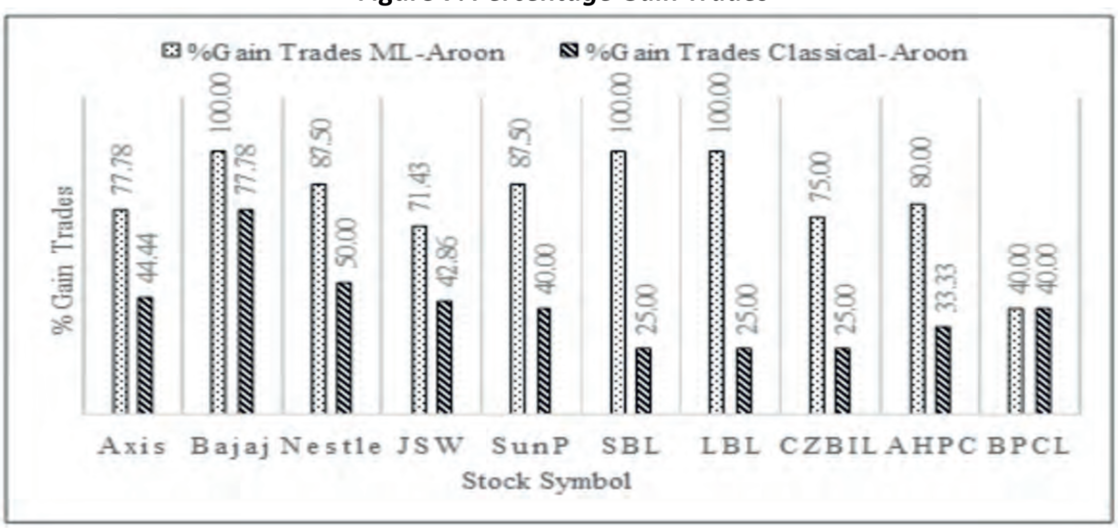
Figure 6. *Sharpe Ratio (SR)*



Analysis of Profitable Trades

One of the most essential metrics for any stock trading strategy is the percentage of profitable trades. Therefore, the automated trading simulation module was used to keep track of this metric. Figure 7 shows a comparison of the proportion of profitable trades executed by the Aroon-ML technique vs the classical-Aroon approach. One notable observation from the figure is that the Aroon-ML method executed a high percentage (i.e. 40% to 100%) of gain trades. However, relatively a smaller percentage (i.e. 25% to 77.78%) of trades performed using the classical-Aroon approach was profitable. On average, the Aroon-ML strategy executed 81.92% profitable trades whereas the classical strategy only executed 40.34% profitable trades. This observation reveals that the Aroon indicator generated a substantial number of incorrect trade signals. This leads to another fact: the Aroon-ML

Figure 7. *Percentage Gain Trades*



trading method filtered out large percentages of erroneous trading signals and learned true Buy/Sell signal patterns from the dataset.

Conclusion, Implications of the Study, and Scope for Future Research

This study offered an intelligent stock trading approach to forecast stock trading signals based on the Aroon indicator (Aroon-ML). The proposed Aroon-ML trading method was compared to the classical Aroon indicator based trading strategy (classical-Aroon) in terms of performance. The annual rate of return (ARR), the Sharpe ratio (SR), and the percentage of gain/loss trades conducted by the trading methods were used to evaluate the trading techniques.

The results of the experiment provided evidence that the Aroon-ML method outperformed the classical-Aroon strategy in all three measures. The Aroon-ML technique yielded an ARR of 8.39% to 40.07% higher than the classical-Aroon method. Moreover, it achieved the specified ARR with positive SR for all stocks, whereas the classical-Aroon approach only achieved positive SR for four stocks. Additionally, the Aroon-ML strategy executed 40 % to 100% profitable trades, whereas the classical trading strategy executed only 25% to 77.78 % profitable trades. Based on these findings, it can be concluded that the proposed Aroon-ML approach is a better and less risky trading strategy than the classical-Aroon strategy. The main implications of this research work are listed below :

- Devised an intelligent stock trading strategy on the basis of the relationship between trading strategies.
- Demonstrated that machine learning empowered the Aroon indicator based trading strategy is superior to classical Aroon indicator based trading strategy.
- The finding indicated that an intelligent trading strategy could be developed based on the relationships between technical indicators commonly used by technical analysts.

This research can be expanded in a variety of ways. One of the simplest approaches is to replicate the research work by experimenting with stocks from other stock exchanges and comparing the results. Another method is to devise intelligent stock trading strategies using other technical indicators. The third way to expand on the research is to use a reinforcement learning strategy to train the machine using trading signals generated by Aroon indicators.

Limitations of the Study

In this study, the proposed strategy experimented only with stocks from developing countries such as Nepal and India. It is suggested that the proposed strategy be experimented with the stocks from developed countries. Another limitation of this study is that the proposed strategy was only implemented using the GRU network. It is recommended that the proposed strategy be implemented using both GRU and LSTM networks and the results be compared.

Authors' Contribution

All authors actively participated in the entire research process. Mr. Arjun Sing Saud was the driving force behind this research. He devised the proposed intelligent trading strategy, put it into action, collected data, analyzed the results, and wrote the research paper. Mrs. Bindu Neupane assisted with implementation, data analysis, and manuscript preparation. She had played a critical role in locating and summarizing relevant research papers. Prof. Subarna Shakya served as the research project's advisor. His suggestions and comments were invaluable throughout the research process.

Conflict of Interest

The authors certify that they are not affiliated with or involved in any organization or entity that has a financial or non-financial interest in the subject matter or materials discussed in this manuscript.

Funding Acknowledgement

The authors received no financial support for their research, authorship, or publication of this article.

References

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods. *Expert Systems with Applications, 36*(3), 5932–5941. https://doi.org/10.1016/j.eswa.2008.07.006

Bombay Stock Exchange. (n.d.). *S&P BSE SENSEX stock prices*. https://www.bseindia.com/markets/equity/EQReports/StockPrcHistori.aspx

Chen, Y., & Hao, Y. (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications, 80*, 340–355. https://doi.org/10.1016/j.eswa.2017.02.044

Chiang, W.-C., Enke, D., Wu, T., & Wang, R. (2016). An adaptive stock index trading decision support system. *Expert Systems with Applications, 59*, 195–207. https://doi.org/10.1016/j.eswa.2016.04.025

Cho, K., Merriënboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics, 11*(6), 1307–1317. https://doi.org/10.1007/s13042-019-01041-1

Gao, T., & Chai, Y. (2018). Improving stock closing price prediction using recurrent neural network and technical indicators. *Neural Computation, 30*(10), 2833–2854. https://doi.org/10.1162/neco_a_01124

Gudelek, M. U., Boluk, S. A., & Ozbayoglu, A. M. (2017). A deep learning based stock trading model with 2-D CNN trend detection (Accession No. 17560625). In, *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–8. https://doi.org/10.1109/SSCI.2017.8285188

Jabbarzadeh, A., Shavvalpour, S., Khanjarpanah, H., & Dourvash, D. (2016). A multiple-criteria approach for forecasting stock price direction: Nonlinear probability models with application in S&P 500 index. *International Journal of Applied Engineering Research, 11*(6), 3870–3878.

Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLOS One, 14*(2), Article 0212320. https://doi.org/10.1371/journal.pone.0212320

Lien Minh, D., Sadeghi-Niaraki, A., Huy, H. D., Min, K., & Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. *IEEE Access, 6*, 55392–55404. https://doi.org/10.1109/ACCESS.2018.2868970

Long, W., Lu, Z., & Cui, L. (2019). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems, 164*, 163–173. https://doi.org/10.1016/j.knosys.2018.10.034

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science, 170*, 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

Nada, P., & Igor, J. (2016). Comparative analysis between the fundamental and technical analysis of stocks. *Journal of Process Management and New Technologies, 4*(2), 26–31. https://doi.org/10.5937/JPMNT1602026P

Nepal Stock Exchange Ltd. (n.d.). *Stock prices*. http://nepalstock.com.np/stockWisePrices

Nti, I. K., Adekoya, A. F., & Weyori, B. A. (2020). A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review, 53* (4), 3007–3057. https://doi.org/10.1007/s10462-019-09754-z

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

Rodríguez-González, A., García-Crespo, Á., Colomo-Palacios, R., Guldrís Iglesias, F., & Gómez-Berbís, J. M. (2011). CAST: Using neural networks to improve trading systems based on technical analysis by means of the RSI financial indicator. *Expert Systems with Applications, 38*(9), 11489–11500. https://doi.org/10.1016/j.eswa.2011.03.023

Sarangi, P., Sinha, D., Sinha, S., & Dubey, M. (2019). Financial modeling using ANN technologies: Result analysis with different network architectures and parameters. *Indian Journal of Research in Capital Markets, 6*(1), 21 – 33. https://doi.org/10.17010/ ijrcm/2019/v6/i1/144039

Saud, A. S., & Shakya, S. (2019). Analysis of gradient descent optimization techniques with gated recurrent unit for stock price prediction: A case study on banking sector of nepal stock exchange. *Journal of Institute of Science and Technology, 24*(2), 17–21. https://doi.org/10.3126/jist.v24i2.27247

Shah, D., Isah, H., & Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies, 7*(2), 26. https://doi.org/10.3390/ijfs7020026

Shen, G., Tan, Q., Zhang, H., Zeng, P., & Xu, J. (2018). Deep learning with gated recurrent unit networks for financial sequence predictions. *Procedia Computer Science, 131*, 895–903. https://doi.org/10.1016/j.procs.2018.04.298

Shynkevich, Y., McGinnity, T. M., Coleman, S. A., Belatreche, A., & Li, Y. (2017). Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing, 264*, 71–88. https://doi.org/10.1016/j.neucom.2016.11.095

Stoean, C., Paja, W., Stoean, R., & Sandita, A. (2019). Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations. *PLoS ONE, 14*(10), Article 0223593. https://doi.org/10.1371/journal.pone.0223593

TA Library in Python. (n.d.). *Documentation*. https://technical-analysis-library-in-python.readthedocs.io/en/latest/ta.html

Tilakaratne, C. D., Mammadov, M. A., & Morris, S. A. (2009). Modified neural network algorithms for predicting trading signals of stock market indices. *Advances in Decision Sciences,* Article id 125308. https://doi.org/10.1155/2009/125308

Trading Strategy Guides. (n.d.). *Aroon indicator trading strategy*. https://tradingstrategyguides.com/aroon-indicator/

Weiss, G., Goldberg, Y., & Yahav, E. (2018). On the practical computational power of finite precision RNNs for language recognition. In, *Proceedings of 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), 2*, 740–745. https://doi.org/10.18653/v1/p18-2117

Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. (2020). Deep reinforcement learning for automated stock trading: An ensemble strategy. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3690996

About the Authors

Arjun Singh Saud received a M.Sc. degree in Computer Science and Information Technology in 2006 from Tribhuvan University, Kathmandu, Nepal. He is a Ph.D. candidate and an Assistant Professor in the same department. Artificial intelligence, machine learning, deep learning, and data science are among his research interests.

Subarna Shakya recieved an M.Sc. in Computer Engineering and a PhD in Computer Engineering from Lviv Polytechnic National University, Ukraine in 1996 and 2000, respectively. He is currently a Professor at the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University. His research interests include electronic government, distributed and cloud computing, software engineering and information systems, deep learning, and data science.

Bindu Neupane recieved a Master's Degree in Computer Application (MCA) from Purbanchal University, Biratnagar, Nepal, in 2020. She is currently a faculty member at Tribhuvan University's Nagarjuna College of Information Technology. Her research interests include data mining and bioinformatics.