Adaptive Machine Learning-Based Stock Prediction Using Financial Time Series Technical Indicators

Ahmed K. Taha(✉), Mohamed H. Kholief, and Walid AbdelMoez

College of Computing and Information Technology, Arab Academy for Science, Technology and Maritime Transport, Alexandria, Egypt
ahmedengu@student.aast.edu, {kholief, walid.abdelmoez}@aast.edu

Abstract. Stock market prediction is a hard task even with the help of advanced machine learning algorithms and computational power. Although much research has been conducted in the field, the results often are not reproducible. That is the reason why the proposed workflow is publicly available on GitHub [1] as a continuous effort to help improve the research in the field. This study explores in detail the importance of financial time series technical indicators. Exploring new approaches and technical indicators, targets, feature selection techniques, and machine learning algorithms. Using data from multiple assets and periods, the proposed model adapts to market patterns to predict the future and using multiple supervised learning algorithms to ensure the adoption of different markets. The lack of research focusing on feature importance and the premise that technical indicators can improve prediction accuracy directed this research. The proposed approach highest accuracy reaches 75% with an area under the curve (AUC) of 0.82, using historical data up to 2019 to ensure the applicability for today's market, with more than a hundred experiments on a diverse set of assets publicly available.

Keywords: Stock price prediction · Technical indicators feature importance · Adaptive stock prediction · Machine learning · Feature selection

1 Introduction

Despite the abundance and the quality of stock market data, it suffers from a high noise to signal ratio that makes the task of predicting the stock market direction an extremely challenging task. It has been established by the random walk hypothesis [2] and the efficient market hypothesis [3] that no one can beat the markets, however, the extraordinary results of quant funds may indicate otherwise. All of that is due to the uniqueness of the stock market unlike any other problem; the financial markets require careful handling of the data to prevent look-ahead bias and overfitting that results in false discoveries from experienced and inexperienced data scientists alike.

Following a strict research process is the key to achieve high accuracy market forecasts, as suggested by López de Prado [4]. De Prado also places emphasis on the importance of avoiding overfitting at all costs by ensuring test and training data

H. Yin et al. (Eds.): IDEAL 2019, LNCS 11872, pp. 191–199, 2019.
https://doi.org/10.1007/978-3-030-33617-2_21

separation; reviewing data preprocessing results to make sure there is no data leakage; and always start with a hypothesis to avoid fitting to the noise, which can happen due to the prolonged exposure to the same data.

Japanese Candlestick Charting Techniques is considered to be a good indication of the timeframe it is covering, as it resamples the trades into Open, High, Low, Close, and Volume as a snapshot of a timeframe. However, a bar chart over a long time can be considered to be very complicated to use alone to understand the market. That is why technical analyst [5] ordinarily makes use of technical indicators [5], which are mathematical formulas that try to reduce the market noise to help identify opportunities.

Technical analysis [5] is a method that uses statistical analysis to evaluate market assets and involves looking at asset price movements over the years. Trying to identify recurring patterns and price action for entering and exiting the market, this can be conducted manually by a technical analyst who stares at the market charts for hours constructing support and resistance lines to identify the market trend, which is not scalable.

The proposed work aims at designing an adaptive machine learning workflow, with the following contributions: (1) Inclusion of one hundred fifty-four technical indicators plus seventeen time-based features, (2) Employing multiple machine learning algorithms and feature selection techniques to generate a scoring leaderboard, which isused to achieve consistent performance over multiple timeframes and assets, (3) Exploring multiple unrelated markets, timeframes, and targets to illustrate the benefits of the proposed approach; with publicly accessible codebase on GitHub [1] to allow results validation, improvements, and future modifications, (4) Exploring rarely researched machine learning algorithms and feature selection techniques such as generalized linear model and feature importance extraction from deep learning models using the Gedeon method, alongside a diverse set of feature selection and machine learning algorithms, (5) Introducing the ability to improve prediction using alternative targets such as the High price which is explored below, using the High price as a label allows capturing better signals from the market that can then get used within a trading strategy.

This paper is structured as follows: Sect. 2 presents the field literature. Section 3 describes the proposed model layers in detail from the dataset, data preprocessing to model training. Section 4 highlights and analyzes the results of the experiments, and finally, Sect. 5 concludes the paper.

2 Related Work

Ding et al. [6], introduced a long term and short term stock price prediction based on a convolutional neural network with input processed using an event-embedding layer. Ding proposed approach improved the accuracy by 6% compared to reviewed papers, thus achieving individual stock prediction accuracy of 65.48%. The paper's event embedding layer is a CNN that trained using input datasets of events from Reuter's financial news and Bloomberg financial news using Open IE. The prediction model training makes use of the previous layer output divided into long-term, medium-term, and short-term events.

Si et al. [7] introduced a topic-based twitter sentiment technique. Si proposes one of the first attempts to make use of a non-parametric continuous Dirichlet process model (cDPM) based on twitter sentiments for stock prediction in an autoregressive framework experimented on real-life Standard & Poor's 100 (S&P 100) market index. The author experiments involved three phases. First, the author makes use of cDPM to identify the daily topic set, which helps solve the broad topic diversity of Twitter tweets. Next, the proposed technique makes use of an opinion lexicon to determine every tweet sentiment to build a sentiment time series. Finally, Si presents a vector autoregressive layer that uses sentiment time series and stock index to predict the stock index. The best prediction the paper was able to achieve was with using a window size of (21, 22) days of data with an average accuracy of 68.0%.

In [8], Jung and Aggarwal introduce a feature generation layer that makes use of 24 technical indicators and generate a total of 24 numerical features and 126 binary features as the input dataset to the prediction algorithms. Jung makes use of a Bayesian Naive Classifier and a Support Vector Machine (SVM) as prediction algorithms. The author used six years of S&P 500 daily stock prices as the raw dataset, and the experiment revealed a performance accuracy of 73.9% for Naive Bayes and 74.3% for SVM. Moreover, the Naive Bayes was able to finish within 10 s while the SVM training time was 600 s.

In Zhang et al. [9], a comparative study has been conducted based on 13 years' worth of daily data from the Chinese stock market, to observe the features effect on the model used. The author proposed a causal feature selection (CFS) algorithm to select the most relevant set of features to improve the model predictability. Moreover, Zhang compares it with principal component analysis (PCA) [10], decision trees (DT) [11], and the least absolute shrinkage and selection operator [12]. The author chooses to work with 50 fundamental and technical features from various categories, such as Valuation factors, Profitability factors, and Growth factors. Zhang trains the model on a sliding window of the dataset one year at a time to prevent overfitting; the CFS approach on average selected the highest number of features achieving higher prediction accuracy over PCA, DT, and LASSO, with a reported accuracy of 54.70%.

3 Proposed Approach

The proposed model workflow in Fig. 1 constructs an adaptive stock prediction model that it is usable for any market or timeframe. The model consists of four different layers. The first layer is the data-gathering layer that is configurable to collect data from multiple sources such as Yahoo Finance, Dukascopy, and Kaggle. The second layer is the data-preprocessing layer that generates the targets and 154 technical indicators from the OHLCV data, alongside the Exploratory Data Analysis (EDA) of the input dataset and the generated features to help understand and evaluate the results, then finally scaling and splitting. The third layer is where the model training, hyperparameter optimization, and feature elimination takes place; Model training happens first on all the features to define a performance benchmark, then once more after the feature elimination process as a two-way process with adaptive feedback. The fourth layer

evaluates and scores the training performance then sorts based on AUC as it considers the randomness and imbalance of the testing dataset.

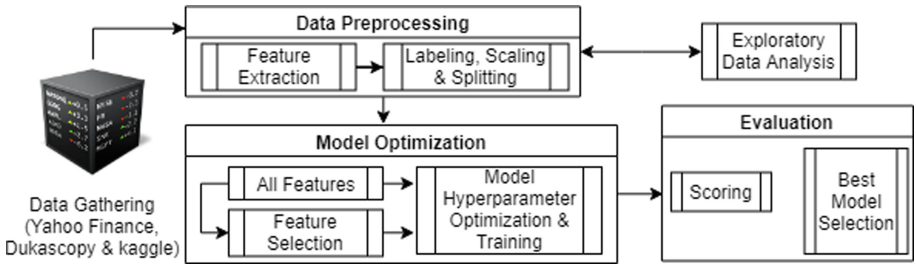

Fig. 1. Illustration of the proposed machine learning workflow for adaptive stock prediction using technical indicators.

The published workflow on GitHub [1] can be configured and reused to test any other asset available on Yahoo Finance or Dukascopy, with no code modifications nor local environment, as the code can be executed from the browser using Google Colab.

3.1 Datasets

The dataset consists of time series based Open, High, Low, Close, and Volume bars of the asset price to represent a snapshot of the trades that happened in the bar period. The reported results derived from multiple datasets and timeframes, as shown in Table 1. To show how the model performs under different markets and timeframes.

Table 1. A list of the used tickers and data sources.

Asset ticker	Asset name	Daily		Hourly	
		Range	Source	Range	Source
AAPL	Apple US	2009–2019	Yahoo Finance	2018–2019	Dukascopy
EURUSD	EUR/USD	2009–2019	Yahoo Finance	2018–2019	Dukascopy
BTCUSD	Bitcoin USD	2012–2019	Bitstamp	2018–2019	Bitstamp

The conducted experiment uses multiple data sources as shown in Table 1; that is due to Dukascopy one year constraint on historical data; thus Yahoo Finance data is used as the source of the ten-year daily timeframe and Dukascopy for the hourly data.

3.2 Data Preprocessing

The target used for this supervised ML problem is a binary label. The label equals (1) when the next bar is higher than the current bar and equals (0) when it is lower. The label calculated for the Close price and the High price within two separate experiments to compare the information gain for each of them.

An extensive list of technical indicators hasbeen used, with all of the datasets in Table 1; at the data-preprocessing layer before training the model. One hundred fifty-four technical indicators [13] was used for this research to allow the evaluation of all technical indicators to capture weak market signals and improving accuracy. Not only that, those indicators have been incorporated with multiple parameters and time-based feature extraction that generates seventeen new features from the timestamp to capture time-related market behavior such as holidays, which increases the number of columns to one thousand twenty-four features.

Then min-max scaling gets applied per row on the entire dataset to ensure no look forward bias and to help the model train. Finally, before training the dataset to get divided into training, validation, and testing sets with the following ratios 70%, 20%, 10% respectively.

3.3 Feature Selection Algorithms

Multiple feature selection techniques tested within the research process, such as principal component analysis, high correlation filters, and low variance filters. However, based on the results of the experiments, this list contains the best performing algorithms: (1)All of the generated features (ALL), (2) Deep Learning Feature (DLF) importance, (3) XGB Feature (XGBF) importance, (4)Recursive Feature Elimination (REF), (5) Family-Wise Error (FWE) rate.

Where ALL represents all of the technical indicators and time-based features generated, the DLF feature's importance is extracted from the trained deep learning model using the Gedeon method that uses the first two-layer of the neural network to assign weights to the input features. XGBF calculates the variable importance from the trained XGBoost model to represent the new input set for the training of all the models. RFE makes use of an extremely randomized tree classifier as the RFE estimator that calculates the feature importance for every step. FWE is a univariate selection technique that filters the features ANOVA f-value based on the p-value set for the experiment.

3.4 Machine Learning Algorithms

This research goal is building a versatile predictive workflow, with many machine-learning algorithms implemented. This experiment used only the three best-performing algorithms listed below: (1) Multi-layer Feedforward Neural Network Deep Learning (DL) [14], (2) eXtreme Gradient Boosting (XGBoost) [15], (3) Generalized Linear Model (GLM) [16].

All of the machine learning algorithms listed above go through the hyperparameter optimization layer, which trains up to 10 models until it reaches the optimal set of parameters that achieve the highest accuracy.

Model training initially took place on the original training dataset as a performance benchmark, then trained with the features and data calculated from the dimensionality reduction algorithms stated before.

4 Experiment Results and Analysis

Table 2 shows the results of the conducted experiments on APPL, Bitcoin, and EURUSD. Moreover, an emphasis on the advantage of the proposed workflow, despite the weakness of some models alone, the entire workflow reports better final results; it can adapt to different markets, timeframes, targets, and regimes. For more than 50 additional experiments, detailed results, exploratory data analysis, and different metrics review the most recent implementation on GitHub [1] and run the experiments in the browser using Google Colab for free.

Table 2. The results of the various experiments in this study, reported in the accuracy metric, with the best accuracy in bold for each asset, target pairs.

Timeframe		Daily						Hourly					
Target		Close			High			Close			High		
Model		DL	GLM	XGB	DL	GLM	XGB	DL	GLM	XGB	DL	GLM	XGB
Asset	Features												
APPL	ALL	0.51	0.51	0.563	0.494	0.494	0.721	0.541	0.553	0.553	0.659	0.688	0.671
	DLF	0.51	0.51	0.538	0.494	0.494	0.725	0.547	0.571	0.582	0.647	0.671	0.694
	FWE	0.51	0.51	0.563	0.692	0.725	**0.729**	0.571	0.553	0.553	0.676	0.659	0.671
	RFE	0.534	**0.563**	0.547	0.7	0.725	0.721	0.559	0.553	0.547	0.641	0.653	0.641
	XGBF	0.51	0.51	0.538	0.494	0.494	0.725	0.571	0.571	**0.582**	0.659	0.671	**0.694**
Bitcoin	ALL	0.546	0.538	0.55	0.602	0.594	0.697	0.587	0.576	0.576	0.651	0.665	**0.692**
	DLF	0.538	0.53	0.55	0.622	0.61	0.709	0.576	0.572	0.579	0.668	0.662	0.69
	FWE	0.582	0.562	0.57	0.622	0.598	**0.717**	0.582	0.584	0.58	0.656	0.673	0.688
	RFE	0.57	0.538	**0.59**	0.653	0.602	0.709	**0.588**	0.583	0.586	0.665	0.68	0.683
	XGBF	0.558	0.53	0.55	0.602	0.61	0.709	0.576	0.572	0.579	0.66	0.662	0.69
EURUSD	ALL	0.719	0.738	**0.754**	0.57	0.551	**0.648**	0.556	0.543	0.562	0.724	0.72	0.733
	DLF	0.742	0.738	0.738	0.555	0.547	0.613	0.562	0.536	0.551	0.726	0.733	**0.734**
	FWE	0.711	0.711	0.703	0.57	0.559	0.57	0.548	0.543	0.562	0.712	0.726	0.721
	RFE	0.746	0.73	0.742	0.586	0.582	0.621	0.543	0.553	**0.562**	0.728	0.715	0.721
	XGBF	0.746	0.738	0.738	0.613	0.547	0.613	0.546	0.536	0.551	0.721	0.733	0.734

In Table 2, Highlights the best results in bold; however, in the case of multiple feature selection approaches achieving the same best accuracy, the approach with the highest average accuracy is highlighted.

Apple achieved 56% on the Daily-Close with GLM-RFE model. However, it reported AUC of 0.53, which may indicate a slight data imbalance; however, with the XGB-ALL approach, it reports the same accuracy with a higher AUC of 0.54 which emphasizes the advantage of using AUC for scoring and sorting, instead of accuracy.

EURUSD reported AUC of 0.82 for the Daily-Close price, 0.66 for the Daily-High price, 0.55 for the Hourly-Close price, and 80% for the Hourly-High price.

Despite Bitcoin volatility, the model achieved 59% accuracy predicting Daily-Close and 71% for the Daily-High, with AUC of 0.56 and 0.74 respectively.

From Table 2 results, it appears that the Close-price sometime has a weaker signal in comparison with the High, Low, and Mid-price targets, that is why it is essential to use an ensemble of multiple algorithms and strategies in production, not just one

machine learning model with an emphasis on sentiment-based models that can act as a circuit breaker. In the case of a black swan event, the system can shut down the system immediately due to instability.

On every run, the proposed architecture captures a different set of essential features that help maximize the predictability of the market direction from technical factors, as shown in Fig. 2. Technical indicators based features such as Balance of Power, Stochastic Fast, and True Range, and time-based features such as Day of the Week are some of the top features that increased helped increase the accuracy of predicting EURUSD direction.

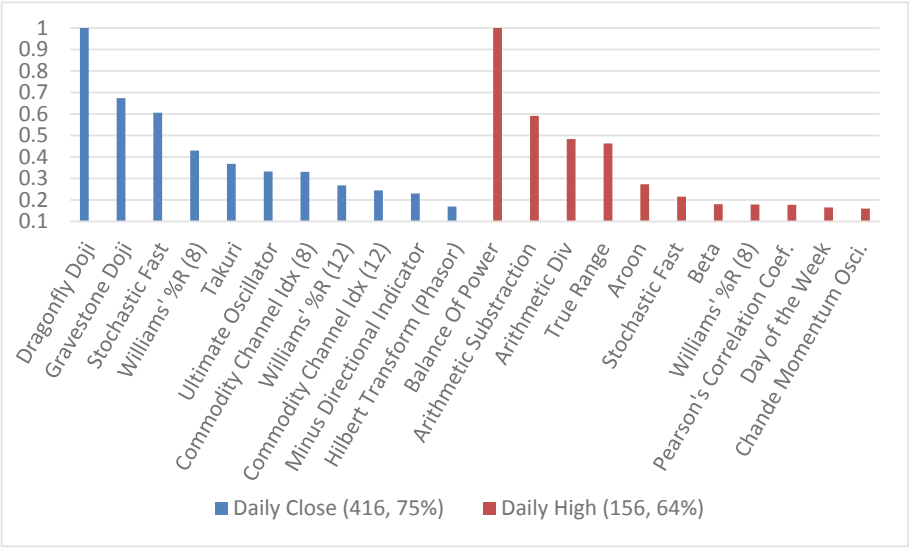

Fig. 2. A comparison of EURUSD top 10 selected features.

EURUSD Daily-Close reports an accuracy of 71% with all the features as an input using the deep learning model, however, after using XGBF the accuracy increases up to 74% with the best accuracy of 75% with the XGB model, and the number of features drops to 416 as in Fig. 2. The drop in the number of features helps improve the model accuracy, as it captures a specific market characteristic of the data used, which leads to adaptive accuracy and less computation requirement for the extra-unrelated features to get predictions on time during back-testing and production.

Figure 3 visualizes the model's prediction, which reports good accuracy. However, using it directly on the market can result in red profit and loss (PnL) despite having high accuracy. It is best to use the model within an algorithmic trading strategy or combined with another model to minimize the downside and protect against black swans.

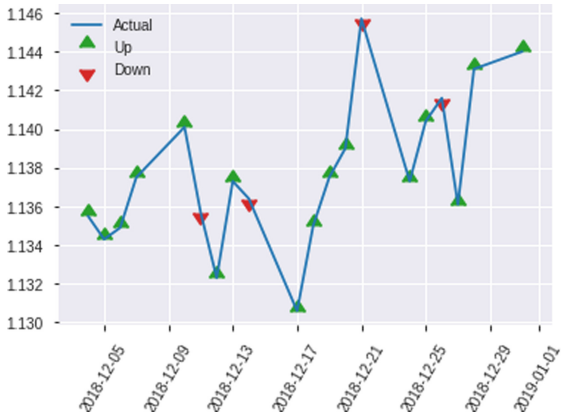

Fig. 3. Euro Daily-Close predictions from XGB with an accuracy of 75%.

The results of the proposed machine learning workflow in Table 2 shows improved results over the current actual research state that does not suffer from any data fallacies, with applicability for multiple markets, targets, and timeframes. With no selection bias, survivorship bias, nor look-forward bias; which is not the case with some of the published work in the field, that only works on a selected over-fitted dataset or with look-forward bias. The proposed model reports the accuracy of the Close price on worst-case 53% and up to 75% and for the High price 58% on worst-case, and up to 74%—those results of the experiments over ten years of data with multiple assets and timeframes.

Compared to Ding et al. [6] Reported results of 65.48%, Si et al. [7] Achieved an average accuracy of 68%, Jung et al. [8] Have best results of 74.3% and Zhang et al. [9] Report best results of 54.70%. The proposed approach reports better results than the sentiment analysis based research and technical indicators literature, on more recent data, on different markets and timeframes.

5 Conclusion

The stock market prediction problem is not easy. Even with the discovery of a trading edge, it will not work for long due to alpha decay. This paper focuses on experimenting with all the financial time series technical indicators available and explores their importance over multiple models, markets, and parameters with different feature elimination algorithms. To ensure the selection of the most valuable features to help with the high noise to signal ratio characteristics of the stock market and capture hidden patterns and asset unique characteristics. The reported results may seem weak individually; however, considering the entire workflow accuracy achieve the level of 72% predicting apple stock, 71% with bitcoin, and 75% EURUSD.

The public repository presents an opportunity to improve the state of research in the field with more than a hundred experiments results available to be evaluated and improved in the future.

Future directions would involve optimizing the implementation performance to allow the experimentations with more datasets and feature variations, which going to undergo general code optimizations and the development for clustered machines.

References

1. Taha, A.: feature importance for ml stock prediction. https://github.com/ahmedengu/feature_importance
2. Fama, E.F.: Random walks in stock market prices. Financ. Anal. J. **51**, 75–80 (1995). https://doi.org/10.2469/faj.v51.n1.1861
3. Fama, E.F.: Efficient capital markets: a review of theory and empirical work. J. Finance. **25**, 383 (1970). https://doi.org/10.2307/2325486
4. López de Prado, M.M.: Advances in financial machine learning (2018)
5. Murphy, J.J., Murphy, J.J.: Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications. Institute of Finance, New York (1999)
6. Ding, X., Zhang, Y., Liu, T., Duan, J.: Deep Learning for Event-Driven Stock Prediction. aaai.org
7. Si, J., Mukherjee, A., Liu, B., Li, Q., Li, H., Deng, X.: Exploiting topic based Twitter sentiment for stock prediction. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2 Short Pap.), pp. 24–29 (2013)
8. Jung, H.J., Aggarwal, J.K.: A binary stock event model for stock trends forecasting: Forecasting stock trends via a simple and accurate approach with machine learning. In: International Conference on Intelligent Systems Design and Applications, ISDA, pp. 714–719. IEEE (2011). https://doi.org/10.1109/ISDA.2011.6121740
9. Zhang, X., Hu, Y., Xie, K., Wang, S., Ngai, E.W.T., Liu, M.: A causal feature selection algorithm for stock prediction modeling. Neurocomputing **142**, 48–59 (2014). https://doi.org/10.1016/J.NEUCOM.2014.01.057
10. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (2002). https://doi.org/10.1007/b98835
11. Quinlan, J.R.: Learning efficient classification procedures and their application to chess end games. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning, pp. 463–482. Springer, Heidelberg (1983). https://doi.org/10.1007/978-3-662-12405-5_15
12. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B. **58**, 267–288 (1996). https://doi.org/10.1111/j.2517-6161.1996.tb02080.x
13. TA-Lib : Technical Analysis Library. http://ta-lib.org/
14. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015). https://doi.org/10.1038/nature14539
15. Chen, T., Guestrin, C.: XGBoost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 2016, pp. 785–794. ACM Press, New York (2016). https://doi.org/10.1145/2939672.2939785
16. McCue, T., Carruthers, E., Dawe, J., Liu, S.: Evaluation of generalized linear model assumptions using randomization (2008). http://www.mun.ca/biology/dschneider/b7932/B7932Final10Dec2008.pdf