Predict Stock Market using Machine Learning under the Help of Smoothening Function

Abstract

Accurate stock market prediction can enable the investors to earn consistently higher risk-adjusted returns, and ML (machine learning) techniques demonstrates great potential to this area, which arouses lots of researchers' interest recently. In this paper, current progress made on the stock market prediction by ML algorithms is reviewed, and a new method is raised to train ML models, which is to apply a smoothening function to smooth out the noises in original datasets and train models on the smoothened datasets so as to improve training score (accuracy). Then back-testing on several stocks (Nasdaq, Apple, Amazon, Alphabet, Tesla & Meta) is conducted to test the profitability of the trained model, and the method is proved to be very effective. Meantime, smoothening degree is counted as an indicator to control the difference between original datasets and smoothened datasets, and the relationship between smoothening degree and training score (accuracy) conforms the law of diminishing returns, hence, a tradeoff should be made when using smoothening function.

Introduction

With the development of computers and internets, financial market becomes increasingly accessible to the public. Obviously, people always invest in stock market based on some prediction about the future values, and accurate prediction can lead to lucrative profits by minimizing investment risks, which attracted many researchers and investors to look for superior techniques to predict the stock market and yield high returns. But the prediction is very challenging, because stock market is non-linear, non-stationary and dynamic in nature.

Although EMH (efficient market hypothesis) associated with the idea of "random walk" argues that the future stock price cannot be predicted according to past trading history, technical analysis, the study of examining past patterns to forecast the future market, is still used by most of major stock traders (approximately 90%) [**5**] and the largest type by number of features researched by academics (about 62%) [**52**]. Some studies challenged the validity of EMH and random walk hypothesis, even refuted it, and proved technical trading rules can be profitable [**1, 5, 24**]. However, traditional technical analysis, which is too simple and the profit based on it can vanish very soon [**47**], does not guarantee decent and stable returns. In recent years, ML (machine learning) / DL (deep learning) models gain traction by the virtue of their great capabilities to process complicated datasets.

Technically, the reason why it is hard to adapt models into financial market is there are too many noises in the stock market datasets, which makes ML models prone to overfit or underfit, resulting in unsatisfactory prediction accuracy. In this paper, a new method is brought up to train ML models by transforming original datasets full of noises to much smoother datasets (namely, smoothened datasets) by a smoothening function, and the difference between original datasets and smoothened datasets are measured and controlled by smoothening degree that shows positive relationship with training accuracy of the models. Then, an algo-trading strategy is built based on the trained models, back-testing is conducted to compare the performance of algo-trading strategy and base-level strategy (buy-and-hold). Ultimately, the new method to train ML models is proved to be effective.

Literature Review

The traditional statistical models employed to predict stock market include: ARIMA and its variants such as ARIMA with genetic programming [**15**], GARCH or time series analytics, etc., and they are compared with the popular machine learning models by some researchers [**3, 44**]. Although there are a few studies indicating that the performance of ANN (artificial neural networks) is lower than that of the statistical models when forecasting short-term stock market [**4**], machine learning techniques can generally achieve more accuracies [**5**, **18**].

Simultaneously, there are numerous works that compared different ML techniques such as SVM (support vector machine) / SVR (support vector regression), ensemble methods like random forest, Bayes-based models and NN (neural networks) [**12, 21, 26, 27, 30, 37, 41, 43**], while one of the most attractive and widely used ML models is NN (neural networks) or its variants [**35, 36, 45, 52**]. Most of related studies aim at the prediction of stock prices or returns (regression task), namely, what the exactly point/index is, time-frequency ranges from minutely, daily, every several-day to monthly [**2, 19, 26, 31, 32, 34, 39, 40, 48**], whereas some research turn to the forecast of stock movements or trends (classification task), namely, whether it will rise or fall (increase/decrease) [**17, 22, 37, 41, 49**]. Additionally, there are also studies concentrating on predicting the market volume [**29**] and forecasting stock market crisis [**33**] and so on, which are quite novel but not mainstream up to now.

As a means to adapt machine learning models to the volatile and uncertain financial market and increase prediction accuracy, various extra-helpers are proposed. Noticeably, the commonly used is GA (genetic algorithms) that helps to search optimal parameters for machine learning models, such as NN variants with GA [**6, 7, 13, 20, 51**], and SVM/SVR variants with GA [**9, 10, 16**]. Apart from GA, diverse other methods are also taken to assist machine learning models, for example, improved bacterial chemotaxis optimization (IBCO) to improve learning ability [**11**], genetic fuzzy systems (GFS) "with the ability of rule base extraction and data base tuning" [**14**], firefly algorithm to optimize hyperparameters for SVR [**23**], particle swarm optimization (PSO) to find optimal parameters [**25**], ensemble adaptive neuro fuzzy inference system (ENANFIS) to determine Close prices [**46**], and scale-free differential evolution (SFDE) as the training algorithm [**53**], and so forth.

Except basic variables and technical indicators (internal factors) used as features for machine learning, a group of researchers also introduced external factors, such as event information [**8**], Oil rates, Gold & Silver rates, Interest rate, Foreign Exchange (FEX) rate [**28**], sentiment analysis extracted from social media is also center-of-attention [**38, 42**], even the housing market information are also worth considering [**50**], and suchlike. Various works show those external factors also play a very significant role in stock market prediction.

Methodology

This Section briefly describes 3 machine learning models applied in our research for the classification task, that is to determine the stock market trends: rise or fall. And more information about those models can be found on Wikipedia.

Machine learning models	details
LR (Logistic Regression)	LR is one kind of supervised learning, which is usually employed to solve binary classification task by predicting the probability of one event occurring out of two alternatives, to be specific, a linear combination of several independent variables is constructed and followed by having a sigmoid function.
SVM (Support Vector Machine, linear core)	SVM is considered to be a popular tool for binary classification problems. SVM maps training examples to points in space in order to create the best line / decision boundary (hyperplane) that maximize the width of the gap between the two categories, and the extreme points that help to create the hyperplane are call support vectors.
RF (Random Forest)	RF is an ensemble learning and efficient for binary classification tasks, it operates by constructing certain number of decision trees, and the output of RF is the class selected by the majority of them. A decision tree can be established by splitting the training examples into subsets based on an attribute. Repeat this process on each derived subset in recursive manner (recursive partitioning) until the splitting no longer adds value to the predictions.

(Table 1, ML models)

Proposed Methods

This part consists of the illustration of smoothening function & smoothening degree, the procedure of feature engineering and labelling, and the training of ML models as well as back-testing to assess the performance.

(1) Smoothening function & smoothening degree

Why to use smoothening function

To draw an analogy with NLP (natural language processing) that processes texts and CV (computer vision) that processes images, NLP usually has pre-processing such as punctuations / stop-words removal, stemming & lemmatization, etc., similarly, the CV frequently includes pre-treating such as image-scaling, mean-filter / median-filter, etc. before training their models. Apparently, they share the same goal to remove noises and transform datasets to be more consistent, which can be regarded as a function of smoothening in some degree.

Financial market dataset is notoriously volatile, in other words, noisy, if training ML models directly based on the original datasets without any pre-processing, the models are regularly vulnerable to overfit or underfit, extremely hard to converge for gradient descent algorithms like LR, SVM and NN. Thus, to pre-process or smoothen the original datasets overwhelmed by noises is very necessary and helpful to further train models.

How to denoise

Using more infrequent data can be a solution. Noises in daily Close points are less than hourly Close point, so are weekly Close points than daily Close points, and monthly Close points than weekly Close points. That may be the reason in part why accuracy improves significantly when time-horizon is expanded to forecast 10-trading-day trend (longer term, more infrequent data) from predict the following-trading-day trend (shorter term, less infrequent data) [**39**]. Nevertheless, if using too infrequent data, there will be very small amount of data available, for example, there are 200+ records of daily Close points, about 50 records of weekly Close points and only 12 records of monthly Close points per stock every year.

Another approach is to use smoothening function. Moving average is a practical tool to smooth out short-term fluctuations and emphasize long-term trends in statistics, namely, to remove some noises by averaging a window-size of datasets and make the time series much smoother. Normally, the bigger the window-size is, the more noises and short-term patterns will be wiped out, the smoother the time series will be and the more long-term trends will be highlighted. Therefore, for the purpose of retaining the short-term patterns and rendering the datasets smoother together, since both are very significant for training ML models, the window-size should not be too great or too small. A tradeoff should be made here.

Smoothening degree,

Evidently, it is needed to invent a measurement to quantify the difference between original dataset and smoothened dataset so as to know and control how much influence the smoothening function exerts on original dataset. Specifically, the measurement is composed of two parts: point difference and rise/fall discrepancy.

Point difference: it is to calculate how much exactly the percentage difference between the points in original datasets and the points in smoothened datasets:

Point difference = AVERAGE (| (points of original dataset – points of smoothened dataset) / points of original dataset |) * 100%

Rise / fall discrepancy: it is to measure how many buy-sell signal discrepancies among original dataset and smoothened dataset. Theoretically, if next day point rises when compared with current day point in original dataset, it should be a buy signal (defined as 1), while after the process of smoothening function, the next day point is instead lower than current day in smoothened dataset which shows a sell signal (defined as 0), conflicting against the buy signal in original dataset, and vice versa. The percentage of rise/fall discrepancy should also be evaluated to gauge the movement distortion after transforming the original dataset with more noises to smoothened dataset with less noises.

Rise / fall discrepancy = SUM (| (buy-sell signals in original dataset – buy-sell signals in smoothened dataset) |) / LENGTH (original dataset) * 100%

Smoothening degree is the multiplication of the two factors:

Smoothening degree = Point difference * Rise / fall discrepancy

Principally, when using smoothening function such as EMA (exponentially moving average) and SMA (simple moving average), the bigger the window-size is, the more short-term noises will be removed and the more long-term trends will be emphasized, the greater the smoothening degree will be.

(2) Feature engineering & labelling

Feature engineering:

In this paper, several technical indicators are used as input data to feed to ML models with the aim of predicting stock market movements. Those features are all based on Close points and are provided in Table 2. More details can be found on Wikipedia.

Features	details
SMA / simple moving average	Mean of the previous k data-points, k is the window-size
EMA / exponentially moving average	EMA(t) = a * x(t) + (1-a) * EMA(t-1), x(t) is the value/point in the current period, a = 2/(k+1), k is the window-size
MACD / moving average convergence/divergence	MACD = EMA (k=12, Close) – EMA (k=26, Close), Signal = EMA (k=9, MACD), Histogram = MACD -Signal, k is the window-size.
RSI / relative strength index	RS = Avg (Gain) / Avg (Loss), RSI = 100 - 100/(1+RS), Avg (Gain) is the average percentage gain during a period controlled by k the window-size. Avg (Loss) is the average percentage loss during a period controlled by k the window-size.
KDJ / stochastic oscillator	%K = (C – L14) / (H14 – L14) x 100, %D = 3-day SMA of %K, %J = 3 * K - 2 * D, C is the current closing price.

	L14 is the lowest price traded of the 14 previous trading sessions. H14 is the highest price traded during the same 14-day period.
Bollinger bands	UP = SMA (k=14, Close) + 2 * STD (k=14, Close), Down = SMA (k=14, Close) - 2 * STD (k=14, Close), k is the window-size

(Table 2, technical indicators)

Labelling:

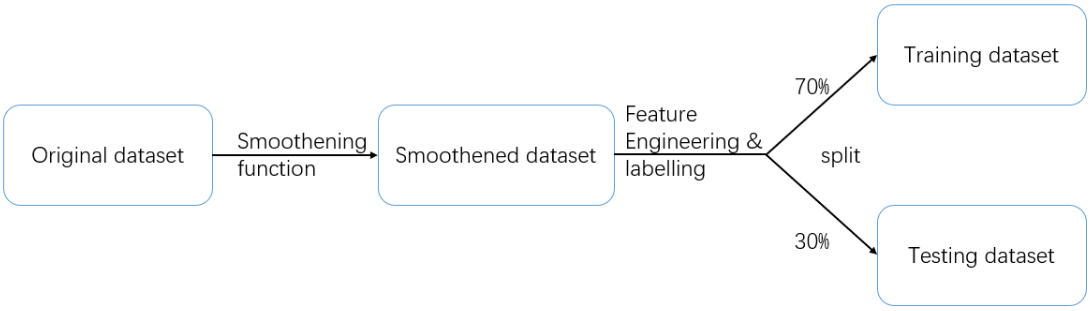
In the paper, ML models are to predict the stock market movements, namely, whether the stock market will rise or fall. If next-week Close point is greater than current-week one (weekly data are used in the experiments), it is rise (buy-signal), and fall (sell-signal) vice versa. Rise signal is marked as 1, and fall signal as 0.

(3) ML models training

Obtain weekly data from 2011-01 to 2022-05 for different stocks (Nasdaq, Apple, Amazon, Alphabet, Tesla and Meta) with the help of Yahoo finance API, the first five have their IPO before 2011, while Meta had its IPO in 2012 so its data is from 2012 to 2022, and these are the original datasets.

Then, original datasets are handled by smoothening function, here it is the EMA, to generate smoothened datasets, which is followed by feature engineering and labelling. Afterwards, smoothened datasets are split into training datasets (70%) and testing datasets (30%) (Figure A).

Calculate the smoothening degrees when imposing smoothening function on original datasets, which reflects the difference between original dataset and smoothened dataset. Next, use training dataset to train ML models and testing dataset to test ML models. Besides, record how the training scores (accuracy) change with smoothening degrees.



(Figure A, the preparation of training & testing datasets)

(4) Back-testing

Back-test to check the profitability of trained model on historical data, namely, from 2019-01-04 to 2022-05-27 for the first 5 stocks (Nasdaq, Apple, Amazon, Alphabet, Tesla), and from 2019-06-07 to 2022-05-27 for Meta because its IPO was after 2011 while the other 5 stocks were before 2011.

There are 2 trading strategies are fabricated to contrast with each other. The first one is base-level strategy (buy-and-hold), and the other is algo-trading strategy based on the trained ML model that forecasts the buy-signal.

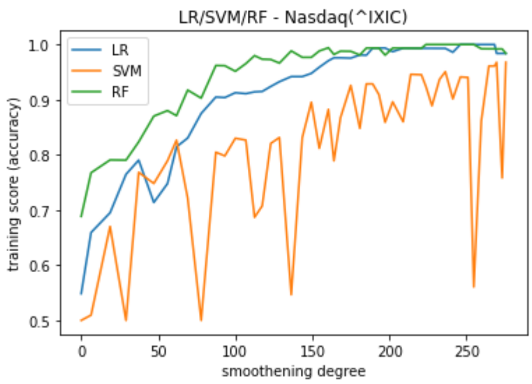
Base-level strategy: buy and hold the stock for the whole testing period.

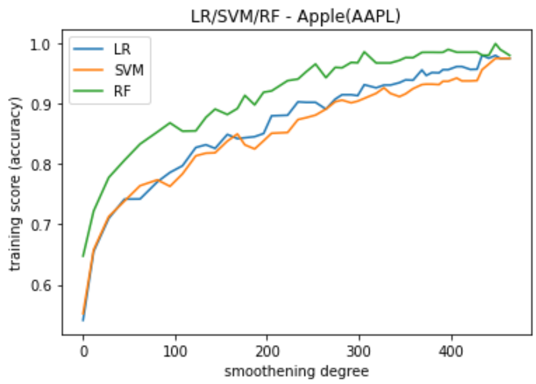
Algo-trading strategy is to, according to the rise / fall signals predicted by ML models, trade every week (because the datasets are weekly). If there is a rise signal predicted by ML model for next week, then buy at the close point (or price) of current week, and sell at the close point (or price) of next week. If there is a fall signal predicted by ML models for next week, do nothing.

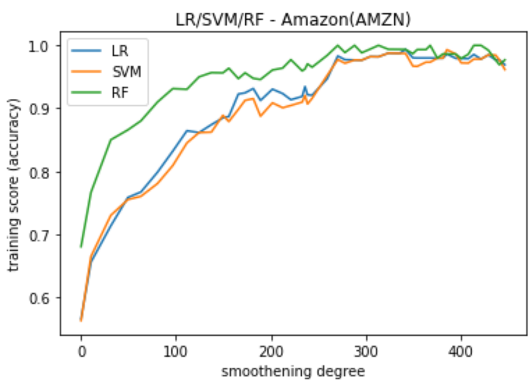
Calculate the profit of executing base-level strategy and algo-trading strategy, while the transaction fees are not considered for simplicity.

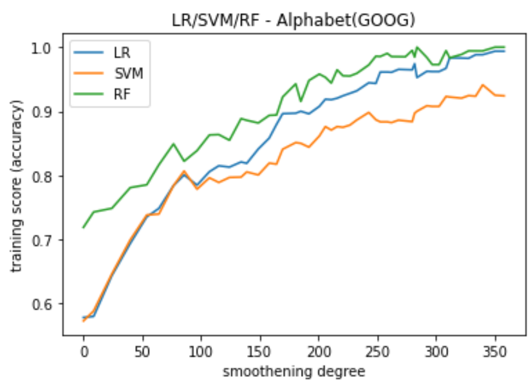
Results & Analysis

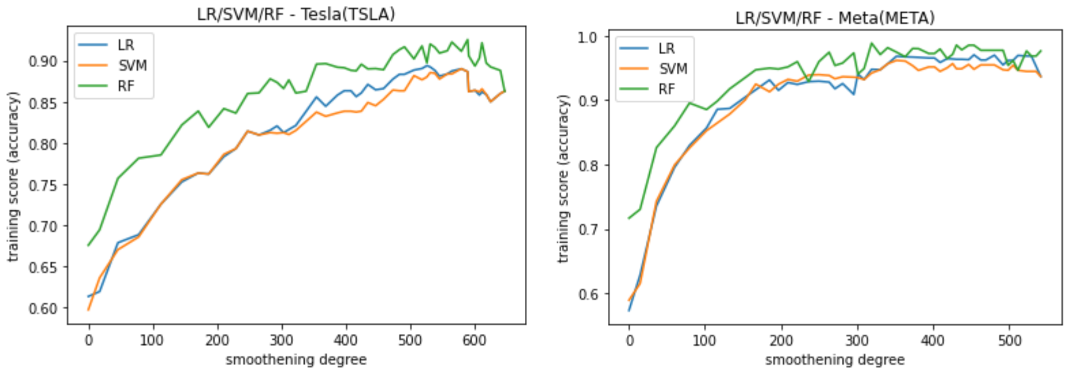
In this paper, smoothening function is introduced to pre-process the original datasets then ML models are trained on the smoothened datasets, and smoothening degree is also presented to measure and control the difference amid these 2 datasets then its relationship with training score (accuracy) is discussed. Further, back-testing on 6 stock data is conducted to assess the model trained by this new way, which shows great profitability when compared with base-level strategy (buy-and-hold).











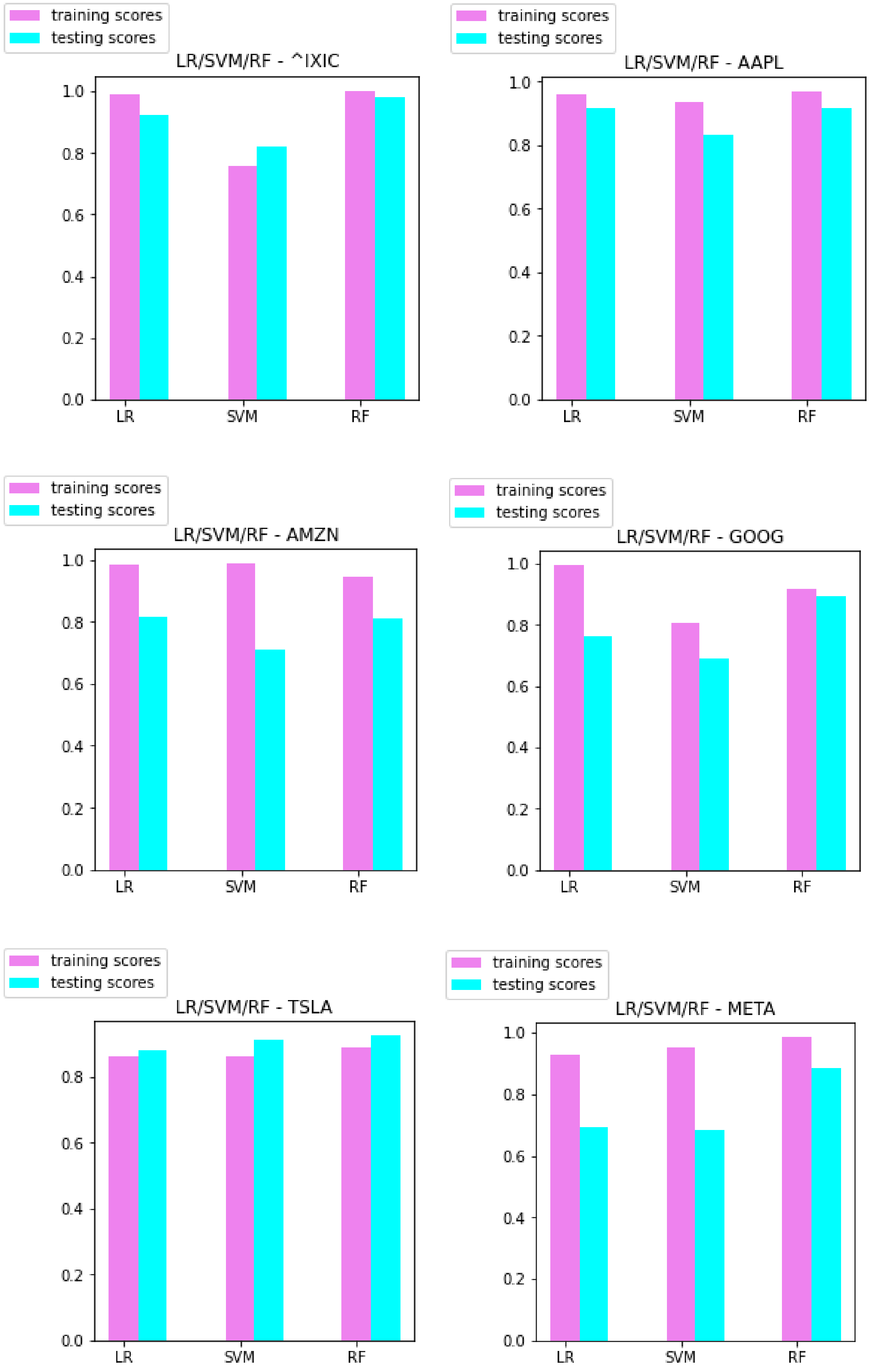
(Figure B, training scores (accuracy) vs smoothening degrees of 3 models on 6 stocks)

(1) Smoothening degree & training scores (accuracy)

Create three models, adjust the parameters for the 3 models (LR, SVM and RF), and hold those parameters fixed. Then continuously increase the parameters of the smoothening function (EMA) to improve the smoothening degree, test the 3 models on 6 stocks (Nasdaq, Apple, Amazon, Alphabet, Tesla & Meta) and observe the changes of the training score (accuracy).

There are 6 graphs above (Figure B) and each image displays how training scores (accuracy) of the 3 models (LR, SVM and RF) vary according to smoothening degree. Generally, positive correlation is clear between training score (accuracy) and smoothening degree for all the 3 models, with the increase of smoothening degree, training scores rise but the momentum decelerates and finally arrives to saturation points. There are a few fluctuations of SVM model performance in the first chart, that is because SVM (linear-core) is quite difficult to converge in nature, at the same time, Nasdaq is mixed index and noisier.

In detail, when smoothening degree is 0, which means there is no smoothening at all, the training score (accuracy) is relatively low because original datasets are full of noises and hard to train models. With the smoothening degree rising, original datasets transformed to smoother datasets, the training of models tends to be more satisfactory. The goal of smoothening function is to denoise original dataset, which is proved to be much helpful to train ML models with higher accuracy, however, the rise of smoothening degree yields diminishing increases for training score (the law of diminishing returns). Obviously, if smoothening function is used excessively, which means greater difference between original dataset and smoothened dataset, it will inevitably lead to ML models' terrible performance when back-testing due to the gap between original dataset and smoothened dataset. Therefore, a balance should be struck for smoothening degree and training score.



(Figure C, training scores vs testing scores of 3 models on 6 stocks)

(2) Different models tested on various stocks

After selecting the appropriate smoothing function parameters (namely, suitable smoothening degrees) for each model, test the 3 models on 6 different stocks, and compare the training and testing scores for each model and contrast their overall performance with each other. Noticeably, both training scores and testing scores of RF model on 6 stocks are very high (about 90%) and their differences are small, which means there is little overfitting or underfitting. Comprehensively, the performance of RF model is more stable and competent than LR and SVM.

Stock	Listing year	Start-date	End-date	Buy & hold profit (%)	Algo-trading profit (%)	Benefit (%)
Nasdaq (^IXIC)	1971	2019-01-04	2022-05-27	80.02	91.50	+11.48
Apple (AAPL)	1980	2019-01-04	2022-05-27	317.02	342.93	+25.91
Amazon (AMZN)	1997	2019-01-04	2022-05-27	46.18	50.35	+4.17
Alphabet (GOOG)	2004	2019-01-04	2022-05-27	110.70	141.94	+31.24
Tesla (TSLA)	2010	2019-01-04	2022-05-27	1095.55	2467.97	+1372.42
Meta (META)	2012	2019-06-07	2022-05-27	12.56	111.09	+98.53

(Figure D, base-level vs algo-trading (RF model) on 6 stocks)

(3) Back-testing on multiple stocks

Choose the best performer RF model and make up an algo-trading strategy based on it, back-test on the 6 stocks and compare with the base-level strategy (buy-and-hold).

For Tesla, base-level strategy (buy-and-hold) gains almost 11 times profit, while algo-trading strategy achieves 24+ times that is 13 times more. And on Meta, more than 110% profit is earned by algo-trading strategy, nearly 10 times the profit acquired by base-level strategy (12%). Meanwhile, algo-trading strategy also obtains 31%, 25% and 11% more profits than base-level strategy on Alphabet, Apple and Nasdaq respectively. Nonetheless, in regard to Amazon, the two strategies perform almost the same with algo-trading strategy winning only 4% more profits.

Conclusion & Future Work

After reviewing the current progress made on the stock market prediction by machine learning techniques, in this paper, a new method is introduced to train ML models, and that is to apply smoothening function to smoothen the original noisy datasets and use smoothening degree to measure and control to what extent the smoothening function impacts original datasets. Experiments show that the smoothening function helps to train models, increasing training accuracy, but the improvement is diminishing until the saturation point is to be reached as a result. Therefore, proper use of smoothening function can be very helpful to train ML models and induce great performance of ML model on back-testing, achieving decent profits.

Future work should focus on 3 main directions for this topic. Firstly, introduce more features, because of limited computing resources, this paper only uses a handful of features and all of them are based on Close points, while more features can be taken into account such as other indicators based on Open / Low / High points, and sentiment analysis extracted from social media and the like in the future work. Secondly, use better smoothening functions that can both preserve short-term patterns and emphasize long-term trends, which not only help to train models but also limit the difference among original datasets and smoothened datasets. Thirdly, try various models or their variants, in fact, neural networks such as ANN, RNN, LSTM and others can be much more powerful than the models used in this paper, while NN algorithms frequently need huge amount of data to train models, or it will be easy to overfit and should be carefully tested. Furthermore, some scholars state reinforcement learning shows great potential to outperform all other algorithms by profitability [**47**], which also deserves more attentions.

Reference

[1] Lo, Andrew W., and A. Craig MacKinlay. "Stock market prices do not follow random walks: Evidence from a simple specification test." The review of financial studies 1.1 (1988): 41-66.

[2] White, Halbert. "Economic prediction using neural networks: The case of IBM daily stock returns." ICNN. Vol. 2. 1988.

[3] Yoon, Youngohc, and George Swales. "Predicting stock price performance: A neural network approach." Proceedings of the twenty-fourth annual Hawaii international conference on system sciences. Vol. 4. IEEE, 1991.

[4] Schumann, Matthias, and T. Lohrbach. "Comparing artificial neural networks with statistical methods within the field of stock market prediction." [1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences. Vol. 4. IEEE, 1993.

[5] Lawrence, Ramon. "Using neural networks to forecast stock market prices." University of Manitoba 333 (1997): 2006-2013.

[6] Phua, Paul Kang Hoh, Daohua Ming, and Weidong Lin. "Neural network with genetically evolved algorithms for stocks prediction." Asia-Pacific Journal of Operational Research 18.1 (2001): 103.

[7] Kim, Kyoung-jae, and Ingoo Han. "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index." Expert systems with Applications 19.2 (2000): 125-132.

[8] Yoo, Paul D., Maria H. Kim, and Tony Jan. "Machine learning techniques and use of event information for stock market prediction: A survey and evaluation." International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06). Vol. 2. IEEE, 2005.

[9] Choudhry, Rohit, and Kumkum Garg. "A hybrid machine learning system for stock market forecasting." International Journal of Computer and Information Engineering 2.3 (2008): 689-692.

[10] Huang, Shian-Chang, and Tung-Kuang Wu. "Integrating GA-based time-scale feature extractions with SVMs for stock index forecasting." Expert Systems with Applications 35.4 (2008): 2080-2088.

[11] Zhang, Yudong, and Lenan Wu. "Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network." Expert systems with applications 36.5 (2009): 8849-8854.

[12] Ou, Phichhang, and Hengshan Wang. "Prediction of stock market index movement by ten data mining techniques." Modern Applied Science 3.12 (2009): 28-42.
[13] Hadavandi, Esmaeil, Arash Ghanbari, and Salman Abbasian-Naghneh. "Developing an evolutionary neural network model for stock index forecasting." International Conference on Intelligent Computing. Springer, Berlin, Heidelberg, 2010.
[14] Hadavandi, Esmaeil, Hassan Shavandi, and Arash Ghanbari. "Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting." Knowledge-Based Systems 23.8 (2010): 800-808.
[15] Lee, Yi-Shian, and Lee-Ing Tong. "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming." Knowledge-Based Systems 24.1 (2011): 66-72.
[16] Pai, Ping-Feng, et al. "Time series forecasting by a seasonal support vector regression model." Expert Systems with Applications 37.6 (2010): 4261-4265.
[17] Kumar, Lokesh, et al. "A hybrid machine learning system for stock market forecasting." Journal of International Technology and Information Management 20.1 (2011): 3.
[18] Shen, Wei, et al. "Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm." Knowledge-Based Systems 24.3 (2011): 378-385.
[19] Zarandi, MH Fazel, Esmaeil Hadavandi, and I. B. Turksen. "A hybrid fuzzy intelligent agent-based system for stock price prediction." International Journal of Intelligent Systems 27.11 (2012): 947-969.
[20] Asadi, Shahrokh, et al. "Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction." Knowledge-Based Systems 35 (2012): 245-258.
[21] Cheng, Cheng, Wei Xu, and Jiajia Wang. "A comparison of ensemble methods in financial market prediction." 2012 Fifth International Joint Conference on Computational Sciences and Optimization. IEEE, 2012.
[22] Aldin, Mahmood Moein, Hasan Dehghan Dehnavi, and Somayye Entezari. "Evaluating the employment of technical indicators in predicting stock price index variations using artificial neural networks (case study: Tehran Stock Exchange)." International Journal of Business and Management 7.15 (2012): 25.
[23] Kazem, Ahmad, et al. "Support vector regression with chaos-based firefly algorithm for stock market price forecasting." Applied soft computing 13.2 (2013): 947-958.
[24] Hung, Nguyen Hoang, and Yang Zhaojun. "Profitability of applying simple moving average trading rules for the Vietnamese stock market." Journal of Business Management 2.3 (2013): 223-1.
[25] Hegazy, Osman, Omar S. Soliman, and Mustafa Abdul Salam. "A machine learning model for stock market prediction." arXiv preprint arXiv:1402.7351 (2014).
[26] Patel, Jigar, et al. "Predicting stock market index using fusion of machine learning techniques." Expert Systems with Applications 42.4 (2015): 2162-2172.
[27] Patel, Jigar, et al. "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques." Expert systems with applications 42.1 (2015): 259-268.
[28] Usmani, Mehak, et al. "Stock market prediction using machine learning techniques." 2016 3rd international conference on computer and information sciences (ICCOINS). IEEE, 2016.

[29] Ghazanfar, Mustansar Ali, et al. "Using machine learning classifiers to predict stock exchange index." International Journal of Machine Learning and Computing 7.2 (2017): 24-29.
[30] Gurav, Uma, and Nandini Sidnal. "Predict stock market behavior: role of machine learning algorithms." Intelligent Computing and Information and Communication. Springer, Singapore, 2018. 383-394.
[31] Parmar, Ishita, et al. "Stock market prediction using machine learning." 2018 first international conference on secure cyber computing and communication (ICSCCC). IEEE, 2018.
[32] Reddy, V. Kranthi Sai. "Stock market prediction using machine learning." International Research Journal of Engineering and Technology (IRJET) 5.10 (2018): 1033-1035.
[33] Chatzis, Sotirios P., et al. "Forecasting stock market crisis events using deep and statistical machine learning techniques." Expert systems with applications 112 (2018): 353-371.
[34] Abe, Masaya, and Hideki Nakayama. "Deep learning for forecasting stock returns in the cross-section." Pacific-Asia conference on knowledge discovery and data mining. Springer, Cham, 2018.
[35] Henrique, Bruno Miranda, Vinicius Amorim Sobreiro, and Herbert Kimura. "Literature review: Machine learning techniques applied to financial market prediction." Expert Systems with Applications 124 (2019): 226-251.
[36] Gandhmal, Dattatray P., and K. Kumar. "Systematic analysis and review of stock market prediction techniques." Computer Science Review 34 (2019): 100190.
[37] Nabipour, Mojtaba, et al. "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis." IEEE Access 8 (2020): 150199-150212.
[38] Khan, Wasiat, et al. "Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis." Soft Computing 24.15 (2020): 11019-11043.
[39] Cervelló-Royo, Roberto, and Francisco Guijarro. "Forecasting stock market trend: A comparison of machine learning algorithms." Finance, Markets and Valuation 6.1 (2020): 37-49.
[40] Vijh, Mehar, et al. "Stock closing price prediction using machine learning techniques." Procedia computer science 167 (2020): 599-606.
[41] Ampomah, Ernest Kwame, Zhiguang Qin, and Gabriel Nyame. "Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement." Information 11.6 (2020): 332.
[42] Khan, Wasiat, et al. "Stock market prediction using machine learning classifiers and social media, news." Journal of Ambient Intelligence and Humanized Computing (2020): 1-24.
[43] Strader, Troy J., et al. "Machine learning stock market prediction studies: Review and research directions." Journal of International Technology and Information Management 28.4 (2020): 63-83.
[44] Rao, Polamuri Subba, Kudipudi Srinivas, and A. Krishna Mohan. "A survey on stock market prediction using machine learning techniques." ICDSMLA 2019. Springer, Singapore, 2020. 923-931.
[45] Kumar, Deepak, Pradeepta Kumar Sarangi, and Rajit Verma. "A systematic review of stock market prediction using machine learning and statistical techniques." Materials Today:

Proceedings (2021).
[46] Zhang, Jun, Lan Li, and Wei Chen. "Predicting stock price using two-stage machine learning techniques." Computational Economics 57.4 (2021): 1237-1261.
[47] Prasad, Akhilesh, and Arumugam Seetharaman. "Importance of Machine Learning in Making Investment Decision in Stock Market." Vikalpa 46.4 (2021): 209-222.
[48] DEMİREL, Uğur, Ç. A. M. Handan, and Ü. N. L. Ü. Ramazan. "Predicting stock prices using machine learning methods and deep learning algorithms: The sample of the Istanbul Stock Exchange." Gazi University Journal of Science 34.1 (2021): 63-82.
[49] Ampomah, Ernest Kwame, et al. "Stock market prediction with gaussian naïve bayes machine learning algorithm." Informatica 45.2 (2021).
[50] Park, Daehyeon, and Doojin Ryu. "A machine learning-based early warning system for the housing and stock markets." IEEE Access 9 (2021): 85566-85572.
[51] Shahvaroughi Farahani, Milad, and Seyed Hossein Razavi Hajiagha. "Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models." Soft Computing 25.13 (2021): 8483-8513.
[52] Kumbure, Mahinda Mailagaha, et al. "Machine learning techniques and data for stock market forecasting: a literature review." Expert Systems with Applications (2022): 116659.
[53] Tang, Yajiao, et al. "Adopting a dendritic neural model for predicting stock price index movement." Expert Systems with Applications (2022): 117637.