



A federated learning-enabled predictive analysis to forecast stock market trends

Saeid Pourroostaei Ardakani^{1,2} · Nanjiang Du² · Chenhong Lin² · Jiun-Chi Yang² · Zhuoran Bi² · Lejun Chen²

Received: 13 September 2022 / Accepted: 9 February 2023
© Crown 2023

Abstract

This article proposes a federated learning framework to build Random Forest, Support Vector Machine, and Linear Regression models for stock market prediction. The performance of the federated learning is compared against centralised and decentralised learning frameworks to figure out the best fitting approach for stock market prediction. According to the results, federated learning outperforms both centralised and decentralised frameworks in terms of Mean Square Error if Random Forest (MSE = 0.021) and Support Vector Machine techniques (MSE = 37.596) are used, while centralised learning (MSE = 0.011) outperforms federated and decentralised frameworks if a linear regression model is used. Moreover, federated learning gives a better model training delay as compared to the benchmarks if Linear Regression (time = 9.7 s) and Random Forest models (time = 515 s) are used, whereas decentralised learning gives a minimised model training delay (time = 3847 s) for Support Vector Machine.

Keywords Federated learning · Decentralised learning · Centralised learning · Stock market trend prediction

1 Introduction

Stock market development suffers from uncertainties caused by a variety of social, financial, and business factors. It results in stock price fluctuations and macroeconomic issues such as inflation and deflation during different periods (Edwards et al. 2018). By this, economists, governors,

and investors are interested in stock market forecasting that allows them to model the market, manage the resources and enhance stock profits.

Machine learning (ML) techniques are widely used to extract knowledge and data pattern from statistics and automate data analysis (Wang et al. 2022). They have the capacity to build classification, prediction and/or regression models for various applications -mainly stock markets. ML models have the capacity to predict stock prices resulting in efficient decision-making and market profit enhancement (Pang et al. 2020). However, ML-enabled stock market predictions are challenging due to the uncertain and dynamic behaviours of the market, data sensitivity concerns, and the complexity of the historical and/or time-series data (Gandhmal and Kumar 2019; Malle et al. 2016).

Federated learning (FL) builds a collaborative ML framework to analyse and explore data patterns (McMahan et al. 2017). It supports data isolation, minimises data sharing and maximises computing parallelism (Yang et al. 2019; Hauschild et al. 2022).

This article aims to build an FL framework for stock trend prediction. For this, three ML models including Linear Regression (LR) (Cakra and Trisedya 2015), Random Forest (RF) (Iannace et al. 2019) and Support Vector Machine (SVM) (Patle and Chouhan 2013) are trained and tested

✉ Saeid Pourroostaei Ardakani
spourroostaei@lincoln.ac.uk

Nanjiang Du
scynd1@nottingham.edu.cn

Chenhong Lin
biycl13@nottingham.edu.cn

Jiun-Chi Yang
scyjy2@nottingham.edu.cn

Zhuoran Bi
scyzb2@nottingham.edu.cn

Lejun Chen
scylc4@nottingham.edu.cn

¹ School of Computer Science, University of Lincoln, Lincoln LN6 7TS, UK

² School of Computer Science, University of Nottingham Ningbo China, 199 Taikang Street, Ningbo 315100, Zhejiang, China

using a public stock market dataset (Quant 2021) which comprises Chinese stock market data. The performance of the proposed FL framework is evaluated and compared against centralised and decentralised learning to find the best-fitted approach in terms of Mean Square Error (MSE) and model training time. The key contributions of this research are listed below:

- Propose a data pre-processing approach to clean and prepare stock market datasets.
- Deploy an FL framework for stock trend predictions, and evaluate its performance against centralised and decentralised learning.
- Compare the performance of three ML models including LR, RF, and SVM to find the best technique for stock market trend prediction.

The remainder of this paper is organised as follows. Section 2 reviews the literature, introduces the relevant state-of-the-art ML techniques for stock market prediction, and highlights the existing research gaps. Section 3 introduces the research methodology and experimental plan. Section 4 presents the experimental results, while Sect. 5 discusses the results to outline the research findings. Section 6 concludes the paper and highlights the future works.

2 Related works

This section introduces ML frameworks, particularly FL, and describes the distinctive ML techniques that are widely used in stock market predictions.

2.1 ML frameworks

ML models can be built and trained via three key frameworks including centralised, distributed and federated learning (Elbir et al. 2021). Centralized learning forms a framework that uses an integrated dataset to build and train an ML model at once (Abadi et al. 2016). However, it may become slow and expensive if a huge and complex dataset is used. Distributed learning aims to resolve this drawback by utilising distributed computing platforms (e.g., Apache Spark) (Chen et al. 2018). It partitions the dataset into several parts (e.g., Resilient Distributed Datasets (Zaharia et al. 2012)) and trains the ML model based on a parallel processing paradigm (Geyer et al. 2017). Federated learning (FL) is a comparatively new approach that supports collaborative machine learning. As Fig. 1 depicts, it partitions the dataset into several parts, each of which trains a local model. In turn, the local models are aggregated to form a master model which is used to analyse the dataset. It offers benefits

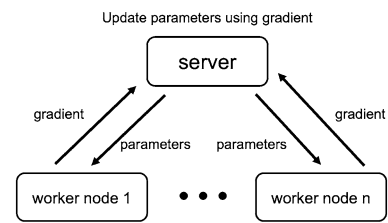


Fig. 1 Federated learning architecture (Tran et al. 2019)

-mainly model training speed-up and data leakage avoidance as compared to centralised and distributed learning.

2.2 Federated learning technology

FL frameworks are categorised into three classes according to data distribution patterns (Yang et al. 2019): horizontal, vertical, and federated transfer learning. Horizontal FL (e.g., Vanilla FL (McMahan et al. 2017)) takes place on datasets that share the same feature space with different samples, whereas vertical FL is used for the same data samples with different feature space (Konečný et al. 2016). Federated transfer learning is used to train machine learning models when the overlap of both features and samples is minimal (Wang et al. 2019).

FL approaches need to use two technologies (Li et al. 2021): Distributed Machine Learning (DML) and encryption technology. DML (e.g., MapReduce) aims to train machine learning models on several computing nodes in parallel. However, FL and DML are different according to the following reasons (Bonawitz et al. 2017; Konečný et al. 2016; Abdul et al. 2021):

1. Control: FL does not allow the server to either directly or indirectly manipulate worker nodes' data, whereas traditional DML such as Mapreduce (Dan et al. 2006) lets the server control the worker nodes.
2. Data distribution and load balancing: DML usually supports independent and identical data distribution (IID) to enhance the efficiency of model training and support load balancing. However, FL does not support IID and may assign different data portions to the worker nodes.
3. Communication cost: DML applications address low communication costs as worker and server nodes are usually located at the same geographical location. However, FL applications suffer from high communication overhead because of interconnecting the nodes through cloud-based communication links.
4. Communication quality: DML's nodes are usually provided with high-speed broadband as they are well-located. Hence, the DML network and operating environments are stable. On the other hand, FL's worker

nodes may experience different connection quality due to network/bandwidth restrictions.

2.3 Stock market prediction

There are a large number of non-FL frameworks and ML models that have been used for stock market predictions. Patel et al. (2015) aim to train the Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest, and Naive-Bayes algorithms to forecast Indian stock markets during a decade. Hassan and Nath (2005) use a Hidden Markov Model to predict the stock prices of four international airlines. Shen and Shafiq (2020) collect, pre-process and analyze 2 years of Chinese stock market records to propose an LSTM deep learning model for stock market trend prediction. Hong (2020) collects and analyses a financial dataset from an international company to study the correlation of stock prices. It uses BLSTM to reduce the errors that usually occur in LSTM models for one-way forecasting. For this, it trains a two-way prediction model by adding macro indicators including economic growth rates, economic indicators, and interest rates, and analyzing the trade balance, exchange rates, and currency volumes.

Big data analysis techniques -mainly in sentiment analysis and text mining- have the capacity to offer stock market prediction benefits. Awan et al. (2021) combine fundamental analysis and Big data techniques to propose a machine learning model (i.e., linear regression, generalized linear regression, random forest, and decision trees) to help investors decide whether to buy or sell a stock. The proposed model uses sentiment analysis and text classification for stocks, tweets, and social media news to predict stock movements. The results show that linear regression, random forest, and generalized linear regression models are able to address an acceptable prediction accuracy. Attigeri et al. (2015) use technical analysis to process social media data in real time. However, they report that fundamental analysis is still required as social media is enormous, unstructured, and rapidly changing. Therefore, they extract sentiment expressed by individuals from social media and use Big data text mining techniques to analyze the correlation between the sentiments and stock values to train a stock market prediction model.

According to the literature review, there is still a gap to propose an FL framework for stock market prediction and compare its performance with non-FL frameworks including centralised or decentralised learning. FL builds predictive models according to a distributed and collaborative machine learning model training fashion. However, the performance of FL can be influenced by dataset partitioning and distribution, especially if the dataset is large and integrated (highly correlated features) (Lundberg 2021). This paper aims to investigate these issues by evaluating and comparing the

performance of FL and centralised/decentralised frameworks in stock market prediction applications.

3 Methodology

This section explains the research methodology aiming to propose a horizontal federated learning framework for stock market prediction.

3.1 Dataset selection and preprocessing

A public dataset of Chinese stock markets for nine provinces including Hubei, Fujian, Sichuan, Shandong, Beijing, Zhejiang, Jiangsu, Guangdong, and Shanghai has been chosen for this research. The key rationale for choosing these nine provinces is that they are the top nine and the most dynamic and strongest stock regions in China (Chinadaily 2022). The dataset is live and contains 2,699,730 samples and eight features including closing price, Low stock Limitation, High stock Limitation, stock ID, total money, stock volume, high price, and low price. A new feature, named price, is also added to the dataset as the result of Money (the total stock market income) over Volume (the total number of sold stocks). This feature is used to study the stock trends as refers to the average price of each stock.

A data cleaning, normalisation and correlation analysis approach is used to prepare the dataset for ML model training. Data cleaning is used to remove the missing and NAN values, while the standard-scaler library is used for data normalisation. Moreover, the Pearson technique (Benesty et al. 2009) is used to analyse the feature correlations. According to the Pearson Correlation Coefficient matrix, the dataset features are highly correlated unless Volume and the stock ID. As a result, the dataset with seven features and stock records from 2014 to 2020 is used to train the models after the data preprocessing. It is randomly partitioned into two parts, the training dataset 80%, and the test dataset 20%. Moreover, a 5-fold cross-validation approach is used to avoid over-fitting and achieve solid results.

3.2 ML framework deployment

This research deploys three frameworks including FL, centralised and decentralised to test and analyse the ML approaches. The given dataset contains stock data samples of nine Chinese provinces. For this, a horizontal FL is required to divide the dataset based on the data samples (stock regions), and assign the data partitions with the same feature space to the worker nodes for processing. As Fig. 2a shows, the horizontal FL framework is set up using nine worker nodes, each of which is assigned by a training dataset (80%) of one province.

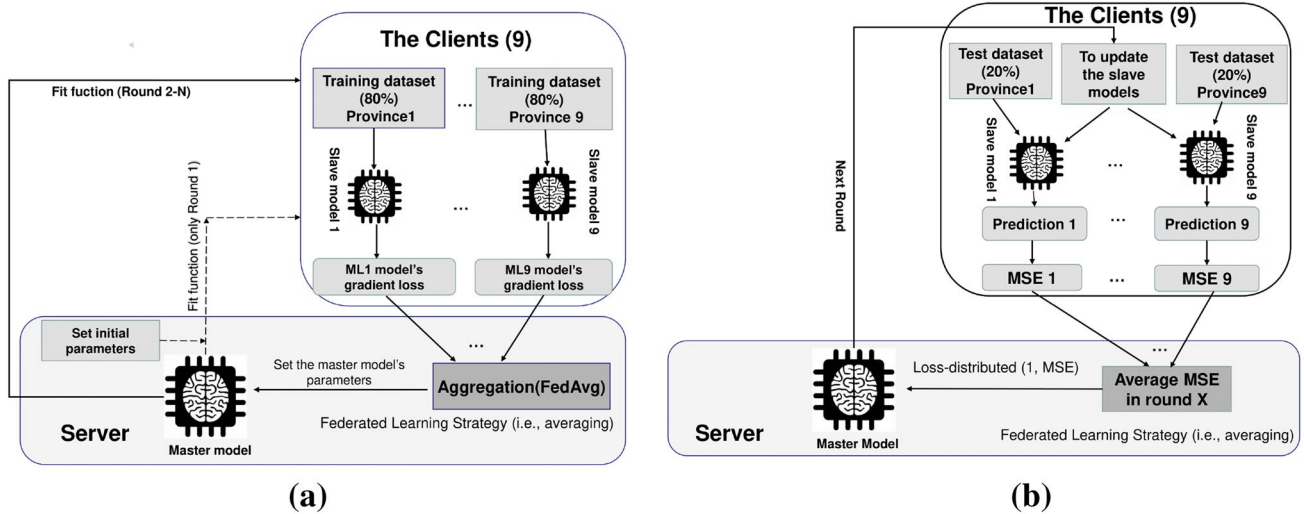


Fig. 2 **a** FL model training. **b** FL model test

Flower framework (Flower 2022) with scikit-learn is used to implement the proposed FL. For this, the minimum number of clients (worker nodes) is set to nine, and the Mean Squared Error (MSE) is used as the framework's evaluation function. By this, each worker trains a local ML using one province data and then sends the model's gradient of loss to the server. The server utilizes a Federated Averaging strategy (FedAvg) to generate a new set of model parameters. The new parameters are sent to the worker nodes to update each local model. This is iteratively repeated based on a convergence fashion and

without data sharing until the application's requirements are met.

According to Fig. 2b, the FL framework evaluates the trained models using local test datasets (20%) for each province. By this, each worker node measures Mean Squared Error (MSE) for an ML prediction and sends the values to the server node for aggregation (i.e., averaging).

Figure 3a shows the decentralised learning framework which is implemented on Apache Spark (Apache 2021) using Spark RDDs. By this, the dataset is portioned into nine RDD, each of which is assigned to a worker thread for

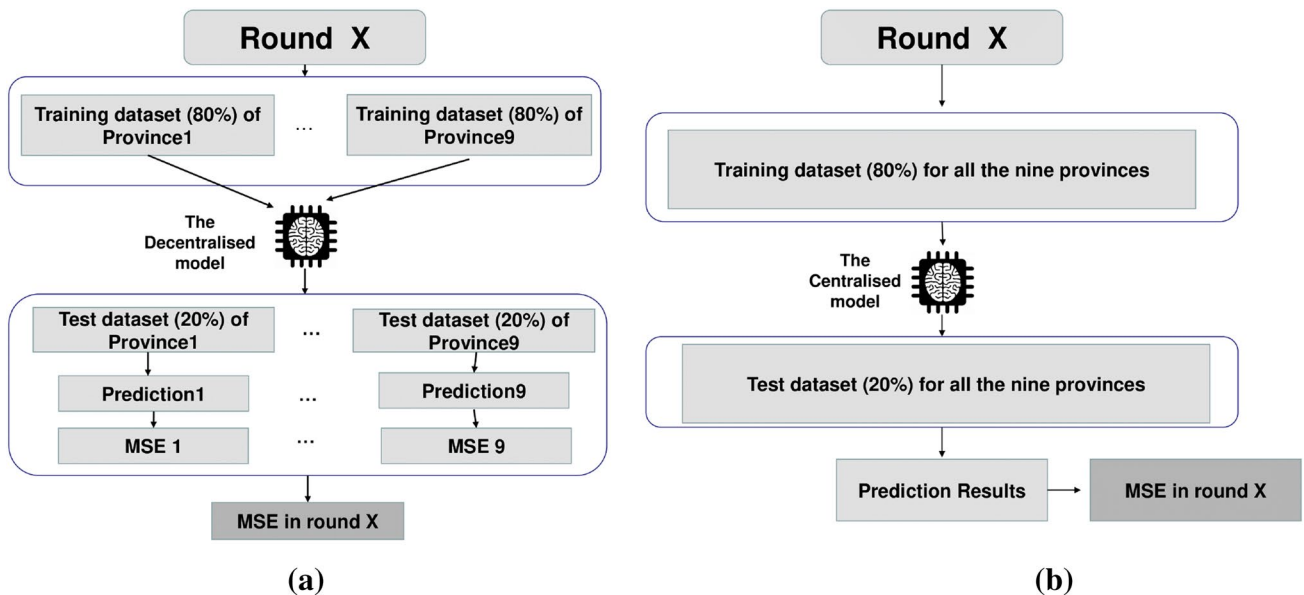


Fig. 3 **a** Decentralized framework. **b** Centralized framework

Table 1 Machine learning model's parameters

	Setup parameters
LR	fit_intercept=True; normalize='deprecated' copy_X=True; n_jobs=None; positive=False
RF	n_estimators=100; criterion='squared_error'; min_samples_split=2 min_weight_fraction_leaf=0.0; max_features=1.0; bootstrap=True
SVM	kernel='rbf'; degree=3; gamma='scale'; verbose=False; coef0=0.0; C=1.0 tol=0.001; epsilon=0.1; shrinking=True; cache_size=200; max_iter=-1

Table 2 MSE results

	FL	Centralised	Decentralised
LR	27.209	0.011	0.012
RF	0.021	0.507	0.025
SVM	37.596	8550.537	49.547

Table 3 ML training time (seconds)

	FL	Centralised	Decentralised
LR	9.7	20	19
RF	515	1083	695
SVM	4989	3951	3847

processing. A MapReduce function (Dan et al. 2006) is used to combine the RDD results and form the final output.

As Fig. 3b depicts, centralised learning uses the whole dataset to train the ML models without parallel processing. Spyder IDE (Gerlach 2022) is used to implement centralised learning. It is a Python development environment that is widely used to build data analysis and ML applications.

4 Experimental result

This section evaluates and compares the performances of federated, centralized, and decentralised Learning frameworks. Each framework builds three ML models including LR, RF, and SVM. They are tested and evaluated in terms of MSE and Model Training Time to study their similarities, differences, and superiorities. Table 1 summarises the setup parameters of the machine learning models.

MSE is measured via Eq. 1, where P and \hat{P} are true price and predicted price respectively, and n refers to the number of samples in the test dataset.

$$MSE = (P - \hat{P})^2/n \tag{1}$$

Training time is measured to study the latency of ML model training. It is increased depending on the model complexity, dataset size, and processing framework performance. Training delay reduction offers real-time prediction/classification benefits.

Table 2 shows the MSE results of the LR model running on the ML frameworks. According to the results, centralized and distributed learning outperform FL if the LR model is used, while FL outperforms centralized and distributed learning when an RF model is used.

Using a Grid Search approach, the SVM model gives the best results if it is an RBF kernel with a degree and C parameter of 1. According to the results, FL with SVM outperforms centralized and distributed learning. However, SVM's MSE is significantly increased as compared to LR and RF on all three frameworks.

Table 3 shows ML model training delay for three ML, each of which is trained via three ML frameworks. According to the results, SVM is the slowest ML model due to the model convergence delay, while LR is the fastest one. Moreover, FL and decentralized learning reduce the ML training delay as compared to a centralised learning framework. This is because both FL and decentralized frameworks use parallel data processing which results in ML training delay reduction. However, FL with SVM increases ML training time as compared to centralized and distributed learning. It is because of the model convergence delay, parameter distribution, and iteration frequency in SVM.

5 Discussion

FL framework works better than centralized and distributed learning to forecast stock market trends according to the following circumstances:

- FL gives better prediction results when RF and SVM models are used as the MSE of FL is lower than the centralised and decentralised learning.
- FL reduces ML training time as compared to the benchmarks if LR and RF models are used. However, decentralised learning gives a better ML training delay for SVM.

- FL shares only model parameters and supports data privacy as compared to centralised and decentralised learning.

The performance of LR model underperforms RF in decentralised learning framework. It is because of the increased error rates caused by data distribution (each province) on the worker nodes in decentralised learning frameworks. However, it outperforms RF if a centralised learning is used. This is because of utilising linear relationships between the selected features at once to train the LR model on a centralised learning framework.

RF gives a better result than LR if an FL is used. It is built based on collaborative decision trees that each of which is established on a worker node to process data features of one province. RF utilises the trained decision trees to predict the market trends that result in MSE reduction.

SVM has the worst performance as compared to LR and RF due to high MSE results. It is because of the large and complex dataset that leads to SVM model convergence failure. As the dataset is huge, the convex optimization approach is unable to support the model convergence (Fine and Scheinberg 2002).

FL has the capacity to minimise data sharing and leakage. Using FL, each worker node locally uses the stock market data of a province to build a slave model. The worker nodes share only the model parameters (i.e., gradient loss) to train the master model without sharing stock data. It would be beneficial for predictive analysis applications as the data owners are not in favor of sharing their market data.

6 Conclusion

This research proposes an FL framework to predict stock market trends. The proposal is established via nine worker nodes, each of which is fed by 6-year stock data (2014-2020) of a Chinese province. A data pre-processing approach is used to clean and prepare the dataset and three ML techniques including LR, RF, and SVM are used to forecast the stock market trends.

An extensive experimental plan is conducted to evaluate and compare the performance of the ML techniques and frameworks for stock market trend prediction. According to the results, FL gives the best performance as compared to centralized and decentralised learning if RF or SVM models are used. However, it underperforms centralized and distributed learning if a LR model is built. As the results show, SVM underperforms LR and RF due to the lack of model convergence during the model training process.

The performance of FL still needs to be evaluated and analysed with true parallelism on multicomputer platforms. This paper utilises a hyper-threading approach to

build the FL framework. However, it is unable to simultaneously run all the available threads due to the restriction of the computing platforms. Service Oriented Architecture (SOA) can be used to establish a distributed computing environment to train the FL model on multiple computing workstations instead of threads.

Author contributions Conceptualization: S.P.A; methodology: S.P.A; programming: N.D, C.L, J.C.Y, and Z.B; results analysis: S.P.A, N.D, C.L, J.C.Y, L.C, and Z.B; writing-original draft preparation: S.P.A, N.D, C.L, J.C.Y, Z.B and L.C.

Funding Not applicable.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. ACM SIGSAC Conference on Computer and Communications Security (ACM CCS), Vienna, Austria, pp 308–318
- Abdul RS, Tout H, Ould-Slimane H, Mourad A, Talhi C, Guizani M (2021) A survey on federated learning: the journey from centralized to distributed on-site learning and beyond. *IEEE Internet Things J* 8(7):5476–5497
- Apache S (2021) <https://spark.apache.org/>. Retrieved on Aug 2022
- Attigeri GV, Pai MMM, Pai RM, Nayak A (2015) Stock market prediction: a big data approach. In: *IEEE Region 10 International Conference TENCON*, Macao
- Awan MJ, Rahim MSM, Nobanee H, Munawar A, Yasin A, Zain AM (2021) Social media and stock market prediction: a big data approach. *Comput Mater Continua (CMC)* 67(2):2569–2583
- Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K (2017) Practical secure aggregation for privacy-preserving machine learning. In: *ACM SIGSAC Conference on Computer and Communications Security*, Texas, USA, pp 1175–1191
- Cakra YE, Trisedya BD (2015) Stock price prediction using linear regression based on sentiment analysis. In: *2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, IEEE, pp 147–154
- Chen T, Giannakis GB, Sun T, Yin W (2018) Lag: Lazily aggregated gradient for communication-efficient distributed learning. In:

- 32nd International Conference on Neural Information Processing Systems, Montreal, Canada, pp 5055–5065
- Chengxi L, Gang L, Varshney PK (2021) Decentralized federated learning via mutual knowledge transfer. *IEEE Internet Things J* 9:1136
- Chinadaily (2022) Top 10 chinese provinces with most stock market investors. https://usa.chinadaily.com.cn/2015-07/14/content_21279336.htm. Retrieved Aug 2022
- Dan G, Faria A, DeNero J (2006) *Mapreduce: distributed computing for machine learning*. Berkley, Cham
- Edwards RD, Magee J, Bassetti WHC (2018) *Technical analysis of stock trends*, vol 11. CRC Press, London
- Elbir AM, Coleri S, Mishra KV (2021) Hybrid federated and centralized learning. In: 29th European Signal Processing Conference (EUSIPCO), pp 23–27
- Fine S, Scheinberg K (2002) Efficient svm training using low-rank kernel representations. *J Mach Learn Res* 2:243–264
- Flower (2022) Flower a friendly federated learning framework, 2022. <https://flower.dev/>. Retrieved Aug 2022
- Gandhmal DP, Kumar K (2019) Systematic analysis and review of stock market prediction techniques. *Comput Sci Rev* 34:100190
- Gerlach CAM (2022) Spyder: the scientific python development environment, 2022. <https://www.spyder-ide.org/>. Retrieved Aug 2022
- Geyer RC, Klein T, Nabi M (2017) Differentially private federated learning: a client level perspective. In: 31st Conference on Neural Information Processing Systems (NIPS 2017) Long Beach
- Hassan MR, Nath B (2005) Stock market forecasting using hidden markov model: a new approach. In: 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Wroclaw, Poland
- Hauschild A-C, Lemanczyk M, Matschinske J, Frisch T, Zolotareva O, Holzinger A, Baumbach J, Heider D (2022) Federated random forests can improve local performance of predictive models for various healthcare applications. *Bioinformatics* 38(8):2278–2286
- Hong S (2020) Research on stock price prediction system based on blstm. *J Korea Converg Soc* 11(10):19–24
- Iannace G, Ciaburro G, Trematerra A (2019) Wind turbine noise prediction using random forest regression. *Machines* 7(4):69
- Jacob B, Jingdong C, Yiteng H, Israel C (2009) Pearson correlation coefficient. *Noise reduction in speech processing*. Springer, Berlin, pp 1–4
- Konečný J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D (2016) Federated learning: strategies for improving communication efficiency. arXiv preprint [arXiv:1610.05492](https://arxiv.org/abs/1610.05492)
- Lundberg O (2021) *Decentralized machine learning on massive heterogeneous datasets: A thesis about vertical federated learning*
- Malle B, Kieseberg P, Weippl E, Holzinger A (2016) The right to be forgotten: towards machine learning on perturbed knowledge bases. *Lecture notes in computer science*. Springer International Publishing, Cham, pp 251–266. https://doi.org/10.1007/978-3-319-45507-5_17
- McMahan HB, Moore E, Ramage D, Hampson S, Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. *Artificial intelligence and statistics*. Springer, Cham, pp 1273–1282
- Pang X, Zhou Y, Wang P, Lin W, Chang V (2020) An innovative neural network approach for stock market prediction. *J Supercomput* 76:2098–2118
- 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 Syst Appl* 42(1):259–268
- Patle A, Chouhan DS (2013) Svm kernel functions for classification. In: 2013 International Conference on Advances in Technology and Engineering (ICATE), IEEE, pages 1–9
- Quant J (2021) Chinese stock data. <https://www.joinquant.com/>. Retrieved Aug 2021
- Shen J, Shafiq MO (2020) Short-term stock market price trend prediction using a comprehensive deep learning system. *J Big Data* 7(1):1–33
- Tran NH, Bao W, Zomaya A, Nguyen MNH, Hong CS (2019) Federated learning over wireless networks: Optimization model design and analysis. In: IEEE Conference on Computer Communications, Paris, France, IEEE, pp 1387–1395
- Wang J, Li L, Wang H (2022) Machine learning concept in de-spiking process for nuclear resonant vibrational spectra - automation using no external parameter. *Vib Spectrosc* 119:103352
- Wang G, Dang CX, Zhou Z (2019) Measure contribution of participants in federated learning. In IEEE International Conference on Big Data, Los Angeles, USA, IEEE, pp 2597–2604
- Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. *ACM Trans Intell Syst Technol (TIST)* 10(2):1–19
- Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I (2012) Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: 9th USENIX conference on Networked Systems Design and Implementation, San Jose, pp 25–27

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.