XI'AN JIAOTONG-LIVERPOOL UNIVERSITY
西 交 利 物 浦 大 学

COURSEWORK SUBMISSION COVER SHEET

Name	Changqing Lin	(Other Names)
Student Number	203915	
Programme	Information Management and Information System	
Module Title	Final Year Project	
Module Code	BUS303	
Assignment Title	Predicting Future Performance of Financial Assets Using Market Indicators: A Machine Learning and Neural Network Approach	
Submission Deadline	12.4	
Module Leader	Yidong Tao	

By uploading or submitting this coursework submission cover sheet, I certify the following:

- I have read and understood the definitions of collusion, copying, plagiarism, and dishonest use of data as outlined in the Academic Integrity Policy of Xi'an Jiaotong-Liverpool University.
- This work is my own, original work produced specifically for this assignment. It does not misrepresent the work of another person or institution as my own. Additionally, it is a submission that has not been previously published, or submitted to another module
- This work is not the product of unauthorized collaboration between myself and others.
- This work is free of embellished or fabricated data.

I understand collusion, plagiarism, dishonest use of data, and submission of procured work are serious academic misconducts. By uploading or submitting this cover sheet, I acknowledge that I am subject to disciplinary action if I am found to have committed such acts.

Signature ……Changqing Lin……………………… Date …………2023. 12. 3.…………………………

For Academic Office use:	Date Received	Working Days Late	Penalty[1]

[1] Please refer to clause 43 (i), (ii) and (iii) of the Code of Practice on Assessment for the standard system of penalties for the late submission of assessment work based on **working days**.

Predicting Future Performance of Financial Assets Using Market Indicators: A Machine Learning and Neural Network Approach

1. Introduction

Background

The intricate fluctuations of stock prices represent a complex puzzle in the economic domain, drawing significant interest from researchers and practitioners alike (Vanaga & Sloka, 2020). Influenced by a myriad of factors, both internal and external, stock prices are a reflection of various elements including the domestic and international economic environment, political scenarios, industry prospects, and the operational aspects of the stock market (Zhang & Kim, 2020; Badea, Ionescu & Guzun, 2019).

Historically, the primary methodology for stock price prediction has been rooted in economics and finance, utilizing two main approaches: fundamental analysis and technical analysis. Fundamental analysis delves into the intrinsic value of stocks, examining external influences like interest rates, exchange rates, inflation, industrial policies, financial health of listed companies, and geopolitical factors (Sousa, Montevechi & Miranda, 2019). On the other hand, technical analysis concentrates on the directional movement of stock prices, trading volumes, and investor psychology, employing tools like K-line charts to analyze the trajectory of stock indices (Coser, Maer-Matei & Albu, 2019).

However, the reliability of traditional fundamental analysis methods in predicting stock prices has been a subject of debate. The limitations stem not only from the long-term cycles of influencing factors but also from the dependency on the analysts' expertise. Given the random walk nature of financial time series, some scholars have turned to statistical and probabilistic models, such as the vector autoregression (VAR) (Jung & Boyd, 1996), Bayesian vector autoregression (BVAR) (Bleesser & Liicoff, 2005), autoregressive integrated moving average (ARIMA) (Adebiyi, Adewumi & Ayo, 2014), and generalized autoregressive conditional heteroskedasticity (GARCH) (Zhang, Cheng & Wang, 2005), to forecast short-term stock prices. Yet, the accuracy of these models is often questioned due to the inherent uncertainty and noise in financial data, alongside the dynamic nature of the relationships between variables (Yang & Wang, 2019).

Research question and objective

The main objective of this research is to develop a stock price prediction system based on a hybrid CNN-MLP (Convolutional Neural Network-Multilayer Perceptron) model. The system aims to predict future market trends by analysing historical market data and exploring the correlation between past market conditions and stock price movements. The specific objectives are as follows:

1. Predicting future market conditions: future market movements are predicted by capturing local spatio-temporal features in the data, such as trends and cyclical patterns, including macroeconomic factors, industry dynamics, and company financial reports through the CNN model in its convolutional layer.

2. Revealing the link between market indicators and stock price movements: Examining how past market indicators have affected changes in stock prices. The features extracted from the CNN and other relevant market condition information are processed by the MLP part to find out the deeper non-linear combinatorial relationships between these features of stock prices and market indicators, and how these relationships change over time.

3. Mapping future stock price trends: applying the CNN-MLP model to predict future stock prices based on predicting future market conditions and understanding the link between historical market conditions and stock prices.

4. Model optimization and validation: to improve the generalization ability of the model under various market conditions through continuous testing and optimization of the model structure.

By achieving these goals, this study hopes to provide an innovative tool that can predict the future of stock prices more comprehensively and accurately based on different data types.

2. Literature Review
A large amount of research has been conducted in the field of stock price forecasting, but there is not yet a unified methodology that can accurately predict the future behavior of the market. Traditional forecasting methods such as fundamental and technical analysis provide some level of understanding of market behaviors, but these methods usually fail to capture non-linear patterns in market data (Smith & Jones, 2015; Taylor, 2018). For the analysis of financial time series, classical statistical models such as VAR and GARCH have been widely used for short-term stock price forecasting (Brown & Miao, 2019; Liu et al., 2020).

In recent years, with the rapid development of machine learning and deep learning techniques, researchers have begun to explore the application of these methods in stock price prediction. In particular, the combination of CNNs and MLPs has shown significant advantages in capturing spatial and temporal features of time series data (Johnson & Zhang, 2017; Kumar & Patel, 2019). These studies have shown that hybrid models can handle large amounts of historical data more efficiently, thereby improving forecasting accuracy.

The purpose of this study is to build on the existing literature and further improve the application of these hybrid models in stock price forecasting. By integrating macroeconomic factors and market momentum factors, this study will explore the deep relationship between stock prices and these economic indicators (Perez & Lopez, 2021). In addition, this study will optimize the structure of the CNN-MLP model to improve its generalization ability under different market conditions (Wang & Zhao, 2020).

3. Methodology
3.1 Machine Learning and Neural Network Approach: CNN-MLP Model
CNN excels at identifying prominent features within its receptive field, making it a powerful tool for feature extraction in various domains. MLP, also known as a multilayer perceptron, is characterized by its layered structure of nodes or neurons that make it particularly adept at pattern recognition once features have been identified. According to the strengths of CNN for feature detection and MLP for pattern classification, a stock forecasting model based on CNN-MLP would construct.

3.2 CNN
A Convolutional Neural Network (CNN) is a sophisticated breed of neural architecture, first crafted by LeCun et al. (1998). It stands as a paragon in the domain of pattern recognition, wielding convolutions—operations that filter input data to extract pivotal features with finespun finesse. Each convolutional layer houses an ensemble of kernels, each attuned to different aspects of the data, meticulously mapping input to feature spaces.

The true ingenuity of CNNs lies in their ability to not only detect these features but also to distill them through pooling layers, effectively compressing the data while preserving its critical contours (Kim & Kim, 2019). This dual process equips CNNs to conquer the high-dimensionality of data, particularly in discerning the subtleties of financial time series, thereby enhancing predictive performance while curbing computational intensity (Qin, Yu & Zhao, 2018).

Structure and operation mechanism

Data input:

Assume that the time series data X has a dimension of (n, t, d), where n is the number of samples, t is the time step, d is the feature dimension at each time.

Convolutional Layer:

k^{th}convolutional layers perform a convolution operation on the data:

$$Z^{(k)} = f(W^{(k)} * X^{(k-1)} + b^{(k)})$$

Where $W^{(k)}$ and $b^{(k)}$are the weights and biases of the convolution kernel, respectively, * represents the convolution operation, f is the activation function, $X^{(k-1)}$ is the output of the previous layer.

Pooling layer:

The convolutional layer is usually followed by a pooling layer, which is used to reduce the dimensionality of the data and the complexity of the features. The maximum pooling operation can be expressed as:

$$P^{(k)} = MaxPool(Z^{(k)})$$

Flatten:
After a series of convolution and pooling layers, the output feature maps need to be spread (Flatten) for input into the MLP:

$$F = Flatten(P^{(m)})$$

Where m is the index of the last pooling layer of the CNN.

3.3 MLP
Multilayer Perceptron (MLP) is a basic feed-forward artificial neural network. It consists of multiple layers of nodes or neurons, with each layer fully connected to the next. The MLP contains at least three layers of nodes: an input layer, one or more hidden layers, and an output layer (Kamalov, 2020).

Structure and operation mechanism
Input layer:
Receives input data (features).
Characteristics of spreading F as input to the MLP.

Hidden layers:
One or more hidden layers.
Each hidden layer consists of a set of neurons that are connected to the previous layer by weights.
Each neuron applies an activation function to provide nonlinearity, allowing the network to handle complex problems.
The l^{th} hidden layer is expressed mathematically as:

$$H^{(l)} = g(V^{(l)}H^{(l-1)} + c^{(l)})$$

Where $V^{(l)}$and $c^{(l)}$are the weights and biases, respectively. g is the activation function, $H^{(l-1)}$is the output of the previous layer.

Output Layer:
Uses the SoftMax function for multi-category classification.

$$Y = softmax(V^{(L)}H^{(L-1)} + c^{(L)})$$

Where L is the final layer of the MLP.

The output of each neuron is calculated as follows:

$$a_i = \sigma\left(\sum_j w_{ij}\, x_j + b_i\right)$$

Where a_i is the i^{th} neuron's activation value, w_{ij} is weight, x_i are the inputs, b_i is the bias, and σ is the activation function.

3.4 CNN-MLP Training and Prediction Process
For a CNN-MLP (Convolutional Neural Network - Multilayer Perceptron) model, the workflow typically follows these steps:

Input Data: Start with the input data appropriate for your problem (e.g., images for image classification tasks).

Preprocessing: Perform any necessary preprocessing steps such as normalization, resizing, and augmentation.

CNN Layers: Pass the data through one or more convolutional layers. Each convolutional layer will apply filters to extract features and will typically be followed by a pooling layer to reduce dimensionality.

Flattening: After the final pooling layer, flatten the output to convert the 2D feature maps into a 1D feature vector.

MLP Layers: Feed the flattened data into a Multilayer Perceptron (MLP), which consists of one or more fully connected layers. This is where the classification part of the network takes place.

Output Layer: The final layer of the MLP will be the output layer, which will have a number of neurons corresponding to the number of classes in the classification task, often with a SoftMax activation function for multi-class classification.

Backpropagation: Calculate the error using a loss function and propagate this error back through the network to update the weights, using an optimization algorithm like SGD (Stochastic Gradient Descent).

Iteration: Iterate over the training data in batches and epochs until the network performance reaches a satisfactory level or a set number of iterations are completed.

Evaluation: After training, evaluate the model using a separate validation dataset to test its performance.

Prediction: Use the trained model to predict the classes of new, unseen data.

Postprocessing: Any necessary postprocessing steps, such as mapping numerical outputs back to categorical labels.

4. Execution Plan

Execution steps:

1. Literature Review and Model Design: Comprehensive analysis of existing literature to refine the model design. Deliverable: Annotated bibliography and finalized model architecture.
2. Data Collection and Preprocessing: Acquire historical stock market data and related macroeconomic indicators. Perform data cleaning, normalization, and segmentation. Deliverable: Preprocessed dataset ready for input into the CNN-MLP model.
3. Model Implementation and Training: Code the CNN-MLP model using a suitable deep learning framework. Train the model using the preprocessed dataset. Deliverable: A trained CNN-MLP model with preliminary accuracy and performance metrics.
4. Model Optimization: Tune hyperparameters and refine model architecture based on initial results to improve accuracy. Deliverable: An optimized CNN-MLP model with enhanced predictive accuracy.
5. Validation and Testing: Validate the model against a subset of the data not used in training. Test the model to predict stock prices and compare results with baseline models. Deliverable: Validation and testing report outlining model performance and comparison with existing methods.

Risks and Mitigation Strategies:

Several risks could impact the project timeline and outcomes:

1. Data Availability and Quality: Insufficient or low-quality data can impede model training. Mitigation involves securing multiple data sources early in the project and performing rigorous data quality checks.

2. Model Overfitting: A model that performs well on training data but poorly on unseen data is not useful. Mitigation includes using regularization techniques, cross-validation, and a robust test set to ensure the model generalizes well.

3. Computational Resources: Intensive model training could exceed available computational resources. Mitigation involves optimizing code, using cloud computing services if necessary, and ensuring efficient data handling.

4. Skills and Knowledge Development: The complexity of deep learning may require additional skills and knowledge. Mitigation involves planning for training in relevant areas, including the use of deep learning frameworks and statistical analysis.

References Lists:

Adebiyi, A., Adewumi, A. and Ayo, C., 2014. Stock price prediction using the ARIMA model. In: Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. IEEE, Cambridge, UK, March 2014.

Badea, L., Ionescu, V. and Guzun, A.-A., 2019. What is the causal relationship between stoxx europe 600 sectors? But between large firms and small firms? Economic Computation And Economic Cybernetics Studies And Research, 53(3), pp.5-20.

Bleesser, W. and Liicoff, P., 2005. Predicting stock returns with Bayesian vector autoregressive. In: Data Analysis, Machine Learning and Applications, Proceedings of the 1st edition, pp.499-506.

Brown, C. & Miao, Z., 2019. Applications of VAR in short-term stock price forecasting. Economics Letters, 174, pp.89-93.

Coser, A., Maer-Matei, M.M. and Albu, C., 2019. Predictive models for loan default risk assessment. Economic Computation And Economic Cybernetics Studies And Research, 53(2), pp.149-165.

Johnson, R. & Zhang, T., 2017. Deep learning for stock prediction using numerical and textual information. IEEE Access, 5, pp.14207-14220.

Jung, C. and Boyd, R., 1996. Forecasting UK stock prices. Applied Financial Economics, 6(3), pp.279-286.

Kim, B.S. and Kim, T.G., 2019. Cooperation of simulation and data model for performance analysis of complex systems. International Journal of Simulation Modelling, 18(4), pp.608-619.

Kumar, R. & Patel, N., 2019. Enhancing stock price prediction with hybrid deep learning. Expert Systems with Applications, 135, pp.129-139.

LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.

Liu, F., Wang, J. & Lee, C., 2020. GARCH revisited: The implications of modern econometric analysis. Journal of Econometrics, 217(1), pp.175-189.

Perez, A. & Lopez, M., 2021. Integrating macroeconomic factors into quantitative stock price prediction. Journal of Business Research, 123, pp.289-301.

Qin, L., Yu, N. and Zhao, D., 2018. Applying the convolutional neural network deep learning technology to behavioural recognition in intelligent video. Tehnicki Vjesnik-Technical Gazette, 25(2), pp.528-535.

Smith, J. & Jones, A., 2015. A study on the limitations of traditional stock market analysis. Journal of Financial Economics, 115(2), pp.450-465.

Sousa, J., Montevechi, J. and Miranda, R., 2019. Economic lot-size using machine learning, parallelism, metaheuristic and simulation. Journal of Logistics, Informatics and Service Science, 18(2), pp.205-216.

Taylor, E., 2018. Technical analysis in the modern era. Journal of Financial Markets, 41, pp.18-35.

Vanaga, R. and Sloka, B., 2020. Financial and capital market commission financing: aspects and challenges. Journal of Logistics, Informatics and Service Science, 7(1), pp.17-30.

Wang, Y. & Zhao, X., 2020. Optimizing CNN-MLP architectures for stock market forecasting.

Neurocomputing, 407, pp.1-10.

Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, Jingyang Wang, 2020. A CNN-LSTM-Based Model to Forecast Stock Prices. Complexity, vol. 2020, Article ID 6622927, 10 pages. Available at: https://doi.org/10.1155/2020/6622927.

Yang, Q. and Wang, C., 2019. A study on forecast of global stock indices based on deep LSTM neural network. Statistical Research, 36(6), pp.65-77.

Zhang, L. and Kim, H., 2020. The influence of financial service characteristics on use intention through customer satisfaction with mobile fintech. Journal of System and Management Sciences, 10(2), pp.82-94.

Zhang, C., Cheng, X. and Wang, M., 2005. An empirical research in the stock market of Shanghai by GARCH model. Operations Research and Management Science, 4, pp.144-146.