

# Portfolio Optimization Based on Return Prediction using Multiple Parallel input CNN-LSTM

Mahdi Ashrafzadeh<sup>a</sup>, Hatef Kiabakht<sup>a,\*</sup>

<sup>a</sup> Department of Industrial Engineering and Management Systems, Amirkabir University of Technology, Tehran, Iran

Received 13 October 2023; Accepted 12 May 2024

## Abstract

The success of any investment portfolio always depends on the future behavior and price events of assets. Therefore, the better one can predict the future of an asset, the more profitable decisions can be made. Today, with the expansion of machine learning models and their advanced sub-branch i.e. deep learning, it is possible to better predict the future of assets and make decisions based on those predictions. In this article, a deep learning method called CNN-LSTM with multiple parallel inputs is introduced and is shown that it is able to provide a more accurate prediction of asset returns for the next period than other machine learning and deep learning models. Then, these forecasts will be used in two stages to build the portfolio. First, the assets that have the highest predicted return are selected, and then in the second step, Markowitz's mean-variance model will be used to obtain the optimal ratio of the selected assets for trading in the next period. The model test is performed on the assets randomly selected from different New York Stock Exchange industries based on the 11 Global Industry Classification Standard (GICS) Stock Market Sectors.

**Keywords:** portfolio optimization, return prediction, multi-parallel input, mean-variance model

## 1. Introduction

Portfolio optimization, which includes the purposeful determination of the ratio of assets to increase returns and reduce risk, is necessary for investors who invest in financial assets, and the mean variance (MV) model presented by Markowitz (1952) is a successful example by which the trade-off point between return and risk can be obtained. Incorporating machine learning (ML) and deep learning (DL) models can further improve performance. By utilizing ML and DL as predictive models to select assets and predicted returns during the optimization process, investors can enhance portfolio performance. The pre-selection of assets is a critical step in portfolio management as it can impact a portfolio's overall performance and risk. Selecting the right assets can be challenging, and failure to do so can lead to suboptimal portfolios that do not meet investment objectives (Wang et al., 2020). Zolfani et al (2022) proposed using the LSTM to predict stock movements and construct an efficient portfolio. Portfolio optimization models were used to investigate performance, including equal-weighted modeling and optimization modeling the MV optimization. The results illustrated that the LSTM prediction model had high accuracy and outperformed other prediction models. They confirmed that combining the LSTM with the MV model is suitable for portfolio construction. Ta et al (2020) Built portfolios by using LSTM neural network and three portfolio optimization techniques, i.e., equal-weighted method, Monte Carlo simulation, and MV model. Also, they applied linear regression and SVM as comparisons in the stock selection

process. Experimental results showed that LSTM neural network owned higher predictive accuracy than linear regression and SVM, and its constructed portfolios outperformed the others. Paiva et al (2019) proposed a unique decision-making model for day trading investments on the stock market, which was developed using a fusion approach of SVM and MV models for portfolio selection. The proposed model was compared with two other models, i.e., SVM+ 1/N and Random+ MV. The experimental evaluation was based on assets from the Ibovespa stock market, which showed the proposed model performed best.

Aside from utilizing ML and DL models for portfolio optimization, another body of research has been dedicated to enhancing the MV model. Freitas (2009) proposed a new portfolio optimization model that utilizes neural network predictors to capture short-term investment opportunities. The model derives a risk measure based on the prediction errors and selects predictors with low and complementary pairwise error profiles to enable efficient diversification. The evaluation of the model using real data from the Brazilian stock market showed that it outperforms the MV model and market index by taking advantage of short-term opportunities and generating normal prediction errors despite the non-normality of stock return time series. Ma et al (2021) employed five different predictive models: the RF, SVR, LSTM, deep multilayer perceptron (DMLP), and CNN. These models were used to pre-select stocks for portfolio optimization, and the predictive results were incorporated into an MV model with forecasting (MVF). The research analyzed the historical data of China Securities 100 Index (CSI 100)

\*Corresponding Author. Email: hatefk2@aut.ac.ir

component stocks from 2007 to 2015. The study concluded that the RF+MVF model was the most suitable for daily investment trading. Lu et al (2020) provide reliable stock price forecasting with the CNN-LSTM model. The experimental result showed their proposed model had the highest prediction accuracy. In this paper a multiple parallel input CNN-LSTM (MPI CNN-LSTM) network is proposed to predict the return of selected assets with minimum prediction error, then the predicted returns are used in two stages like Ma et al (2021) and . In the first stage, assets with the highest predicted return of the next period are selected, which is called pre-selection in the literature. In the next step, the return of the selected assets with their covariance, which is obtained based on the historical data will be used in the Markowitz mean-variance (MV) model, to obtain the optimal ratio of assets in the portfolio and daily rebalancing. More details related to the assumptions, model, and contribution of the paper are discussed in the next sections.

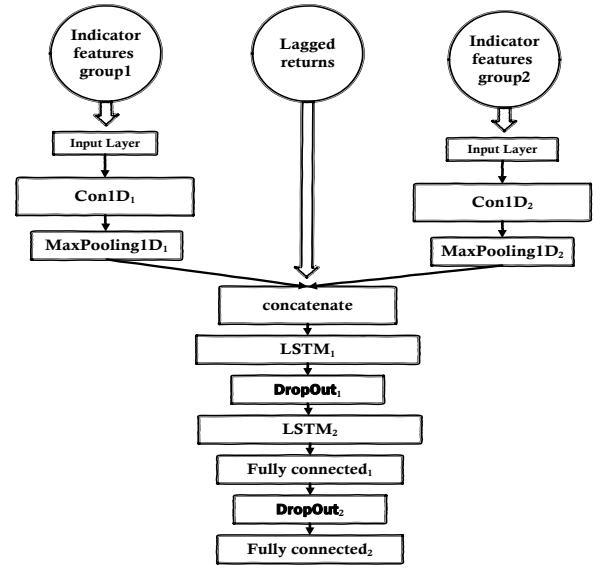


Fig1: MPI CNN-LSTM structure

## 2. Methodology

### 3.1. Multiple parallel input CNN-LSTM (MPI CNN-LSTM)

CNN has the characteristic of paying attention to the most obvious features in the line of sight, so it is widely used in feature engineering. Next is the max pooling layer to reduce the dimensions of the extracted features from data by convolution. LSTM has the characteristic of expanding according to the sequence of time, and it is widely used in time series like Lu et al (2020) therefore, by having a combined model of CNN and LSTM, the power and ability of both neural networks can be simultaneously used to predict returns. The DL model proposed in this article for predicting asset returns is CNN-LSTM with multi-parallel inputs that can be seen in Figure1. There are two types of data used to predict the return of each asset: one is technical indicators that are calculated based on the asset price, and the other are lagged return observations. The technical indicator data are randomly and equally divided into two groups based on the proposed neural network structure. And each group of indicator data is entered into a convolution layer. The output of each convolution network is useful extracted features from the technical indicator data. Then, the extracted features from both parallel structures are concatenated with the lagged return observations and are considered as the input of the LSTM neural network. It is expected that with this structure, useful features that can be effective for predicting the return in LSTM are extracted by CNNs, and it is no longer necessary to use other dimensionality reduction methods separately outside the neural network structure.

Table 1

Applied features and hyperparameters of proposed MPI CNN-LSTM structured in Figure1

	Categories	hyperparameters
Features Group <sub>1</sub>		macd, roc, stochrsi, rsi
	Filters	250
	kernel_size	3
Con1D <sub>1</sub>	activation	selu
MaxPooling1D <sub>1</sub>	maxp	2
Features Group <sub>2</sub>		atr, psar, stochastic, ema
	Filters	250
	kernel_size	3
Con1D <sub>2</sub>	activation	selu
MaxPooling1D <sub>2</sub>	maxp	2
	Unit number	64
LSTM <sub>1</sub>	activation	selu
DropOut <sub>1</sub>	Dropout rate	0.2
	Unit number	64
LSTM <sub>2</sub>	activation	linear
	Neuron number	32
Fully connected <sub>1</sub>	activation	tanh
DropOut <sub>2</sub>	Dropout rate	0.5
	Neuron number	1
Fully connected <sub>2</sub>	activation	linear
	Optimizers	Adam
	learning_rate	0.0001
	epochs	150
	batch_size	512

The input features of each CNNs and the hyperparameters of the proposed model, which their optimal form was obtained by trial and error are shown in Table 1 in detail.

### 3.2. Mean-Variance with Forecasting (MVF) Model

As said before, the mean-variance model proposed by Markowitz in order to solve the optimal portfolio selection issue, which initiates the foundation of Modern Portfolio Theory (MPT). In this model, the investment return and risk are quantified by expected return and variance, respectively. According to Zhou (2019) the most important issue in stock portfolio formation is which stock to keep and which to sell in order to minimize the risk and maximize the profit. Hereby, rational investors always prefer the lower risk portfolios with constant expected returns or the higher expected return portfolios with a constant risk level. To solve this issue, a set of optimal solutions is generated, named an efficient investment frontier. The model can be described by the following formulas overall:

$$\min \sum_{i,j=1}^N X_i \cdot X_j \cdot \delta_{ij} - \sum_{i=1}^N X_i \cdot \hat{\mu}_i \quad (1)$$

$$\sum_{i=1}^N x_i = 1 \quad (2)$$

$$0 \leq X_i \leq 1 \quad i=1, 2, \dots, N \quad (3)$$

Where  $X_i$  means the proportion of asset  $i$  in the portfolio,  $N$  is the number of assets in the portfolio,  $\delta_{ij}$  is the covariance of asset  $i$  and  $j$  which is calculated with

historical data for two years before the test duration and  $\hat{\mu}_i$  denotes the predicted return of asset  $i$ . In order to use the predicted return in the MV model as MVF based Ma et al (2021) and Yu (2020), equation 1 is replaced with equation 4:

$$\min \sum_{i,j=1}^N x_i \cdot x_j \cdot \sigma_{i,j} - \sum_{i=1}^N x_i \cdot \hat{r}_i - \sum_{i=1}^N x_i \cdot \bar{e}_i \quad (4)$$

Where  $\hat{r}_i$  is the predicted return of asset  $i$  at time  $t$ .  $\bar{e}_i$  is the average prediction error of the asset  $i$  which is calculated for the last 20 days before the day  $t$ .  $e_i$  is obtained by  $e_i = r_i - \hat{r}_i$ , where  $r_i$  is actual return of asset  $i$ . In this paper the sample period is considered as 20 days to calculating  $\bar{e}_i$  and covariance matrix like Ma et al (2021) and Yu (2020).

## 4. Experiment and Results

### 4.1 Data and Selected Features

#### 4.1.1 Selected Assets

In this paper, 11 large-cap stocks from different industries of the New York Stock Exchange according to the GICS standard are selected to show the superiority of the proposed model. The time series data from 2012 to 2022 are collected. From 2012 to 2019 considered as training data and from 2020 to 2021 as test data. Table 2 shows some statistical attributes of the selected asset prices.

Table 2  
Summary Statistics for Selected Assets

	mean	std	min	max	range	25%	50%	75%
XOM	59.71	10.33	26.77	102.76	75.99	56.67	60.22	63.95
SHW	129.27	77.91	27.17	348.82	321.65	66.20	105.04	178.93
BA	181.59	96.60	55.67	430.30	374.63	111.95	145.98	239.85
DUK	66.46	18.36	38.64	112.19	73.56	51.09	63.69	77.60
UNH	193.10	132.41	42.55	544.93	502.38	75.28	158.59	262.46
BRK-B	176.18	63.45	76.29	359.57	283.28	128.71	167.47	210.63
AMZN	66.91	54.28	8.80	186.57	177.77	17.53	46.90	96.87
KO	38.92	9.69	23.78	64.80	41.02	31.35	36.74	45.86
MSFT	104.64	88.87	21.47	339.92	318.46	37.06	63.32	148.80
GOOGL	54.26	35.03	13.99	149.84	135.85	27.73	46.20	64.66
AMT	136.90	72.04	48.22	295.19	246.97	78.04	113.23	210.42

4.1.2 Features

As mentioned earlier, in this article, two classes of input based on technical indicators and lagged return observations are applied to predict the next day's return of assets. Moving average convergence divergence (MACD), Price rate-of-change (ROC), Average True Range (ATR), Parabolic SAR (PSAR), Relative Strength Index (RSI), Stochastic Oscillator (Stochastic), Stochastic RSI (StochasticRSI) and Exponential Moving Average (EMA) are 8 technical indicators which are used in this study and also have been used in some other similar studies such as Box et al (2015) and Basak et al (2018). Since our prediction problem is related to financial time series forecasting, it is appropriate to use data from the target variable, which is the return of assets, considering the time lag in them as part of the input features. In this regard, four lagged return observations were also used as another category of variables. Before using the expressed features as input for DL and ML models, they are scaled by the following relation:

$$\text{scaled}_i = \frac{x_i - \mu_i}{\sigma_i} \tag{5}$$

Which is a standard scalar and  $x_i$  means feature  $i$ ,  $\mu_i$  is its expected value and  $\sigma_i$  is its standard deviation. In the experiment, the CNN-LSTM, LSTM neural network and CNN model are implemented based on Keras deep learning package as deep learning models and the SVR, RF and XGB are prepared based on Scikit-learn and xgboost machine learning package as machine learning models to show the superiority of proposed method.

4.2. Prediction

This section first presents the predictive results of different models in stock return prediction during the whole test period. The metrics of mean squared error (MSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) which are expressed in equations 6 to 8, respectively, are used to compare the performance of different ML and DL models.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (r_i - \hat{r}_i)^2 \tag{6}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |r_i - \hat{r}_i| \tag{7}$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{r_i - \hat{r}_i}{r_i} \right| \tag{8}$$

Where  $r_i$  represents actual return and  $\hat{r}_i$  represents predicted return of day  $i$ . Table 3 shows the prediction result over test data. First, by comparing the results of

machine learning models and deep learning models, it can be understood the superiority of deep learning models in forecasting with less error. Between the machine learning models, RF has better performance than others. And between all models, our proposed MPI CNN-LSTM has better performance based on the discussed metrics.

Table 3  
The predictive performance of different DL and ML models

Model		MAE	MSE	MAPE
MPI CNN-LSTM	mean	<b>0.01304</b>	<b>0.00040</b>	8.33098
(The proposed method)	<sup>a</sup> SD	<b>0.00418</b>	<b>0.00029</b>	3.15470
CNN-LSTM	mean	0.01686	0.00072	<b>4.67327</b>
	SD	0.00812	0.00068	<b>1.72892</b>
LSTM	mean	0.01414	0.00049	5.91698
	SD	0.00480	0.00029	2.49926
CNN	mean	0.01635	0.00058	17.43406
	SD	0.00414	0.00032	19.64951
SVM	mean	0.02011	0.00079	6.05492
	SD	0.00737	0.00050	8.56330
RF	mean	0.01558	0.00056	9.55165
	SD	0.00389	0.00034	8.05952
XGB	mean	0.01884	0.00075	5.93299
	SD	0.00536	0.00038	3.05328

<sup>a</sup>SD means standard deviation

4.3. Model Performance

After selecting the stocks with higher predicted returns for the next trading day, MVF is applied to calculate the optimal proportion of each asset in the portfolio. So next day trading action will be taken based on those obtained proportions. This paper simulates buying and selling behaviors as a typical investor. Specifically, an investor decides to buy or sell a certain proportion of each stock from the market before each trading day to achieve the calculated proportion of each stock in the portfolio. To show the superiority of proposed model, the trading simulation is implemented all over the testing period, including 505 samples, and the transaction cost is considered to make the simulation more similar to the reality. The performance of models will be shown in two terms, first with considering 0.5% transaction cost and second with 1% transaction cost. In the following, the results of the performance simulation of the models are shown using statistical and financial criteria and also in the form of diagrams.

#### 4.3.1. Details on Financial Performance

Tables 4 and 5 provide insights into the financial performance of the MPI CNN-LSTM+MVF as the proposed model, compared to the baselines, including transaction cost (0.5%,1%) separately. Hence, Panel A, B, and C depict daily return characteristics, daily risk characteristics, and annualized risk-return metrics respectively. Return characteristics: In panel A of Table 4, we can see that the MPI CNN-LSTM+MVF exhibits a favorable daily mean return of 0.0045 considering 0.5% transaction cost. After including transaction cost of 1%, in panel A of Table 5, we can find that MPI CNN-LSTM+MVF has the highest expected daily return of

0.0019. Risk characteristics: In panel B of Tables 4 and 5, we can see a mixed picture corresponding to risk characteristics. By including 0.5% and 1% transaction cost, RF+MVF achieved the best place with 5 percent VaR and 5 percent CVaR. Annualized risk-return metrics: In panel C of Tables 4 and 5, we discuss risk-return metrics on an annualized basis. For annually expected return, the MPI -CNN-LSTM+MVF exhibits the best performance than others in all tables. It can be seen that MPI CNN-LSTM+MVF has the best annualized sharp ratio than other models in all tables.

Table 4  
Performance characteristics with transaction cost (0.5%)

Model	MPI CNN-LSTM+MVF	LSTM+MVF	CNN+MVF	RF+MVF	XGB+MVF
<b>Panel A: Daily return characteristics</b>					
Expected Return	0.0045	0.0034	0.0025	0.0029	0.0026
<b>Panel B: Daily risk characteristics</b>					
Standard Deviation	0.0273	0.0288	0.0236	0.0194	0.0208
Value at Risk_5%	0.0409	0.0439	0.0365	0.0291	0.0316
Conditional Value at Risk_5%	0.0718	0.0779	0.0626	0.0497	0.0556
<b>Panel C: Annualized risk-return metrics</b>					
Expected Return	2.0625	1.3166	0.8501	1.0487	0.9307
Standard Deviation	0.4315	0.4546	0.3731	0.3069	0.3287
Sharpe ratio	4.7796	2.8961	2.2782	3.4174	2.8313

Table 5  
Performance characteristics with transaction cost (1%)

Model	MPI CNN-LSTM+MVF	LSTM+MVF	CNN+MVF	RF+MVF	XGB+MVF
<b>Panel A: Daily return characteristics</b>					
Expected Return	0.0019	0.0010	-0.0002	0.0003	0.0002
<b>Panel B: Daily risk characteristics</b>					
Standard Deviation	0.0280	0.0291	0.0239	0.0196	0.0211
Value at Risk_5%	0.0441	0.0470	0.0396	0.0319	0.0344
Conditional Value at Risk_5%	0.0722	0.0835	0.0671	0.0514	0.0573
<b>Panel C: Annualized risk-return metrics</b>					
Expected Return	0.6222	0.2746	-0.0594	0.0730	0.0544
Standard Deviation	0.4422	0.4607	0.3780	0.3091	0.3329
Sharpe ratio	1.4070	0.5961	-0.1572	0.2362	0.1635

#### 4.3.2. Visualization of Model Performances

To better show the superiority of the proposed MPI CNN-LSTM+MVF, we visualize the cumulative returns. Figure.2 and Figure.3 shows accumulative returns of each

model during test by considering respectively, 0.5% and 1% transaction cost. The cumulative return of each model decreases ignificantly, but the MPI CNN-LSTM+MVF maintains the highest cumulative return.

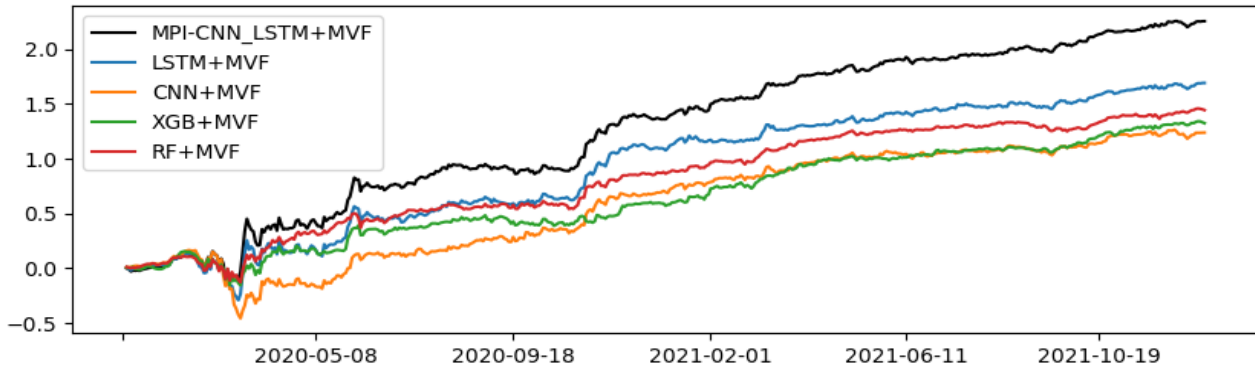


Fig 2. Cumulative return of the portfolio with 0.5% transaction cost

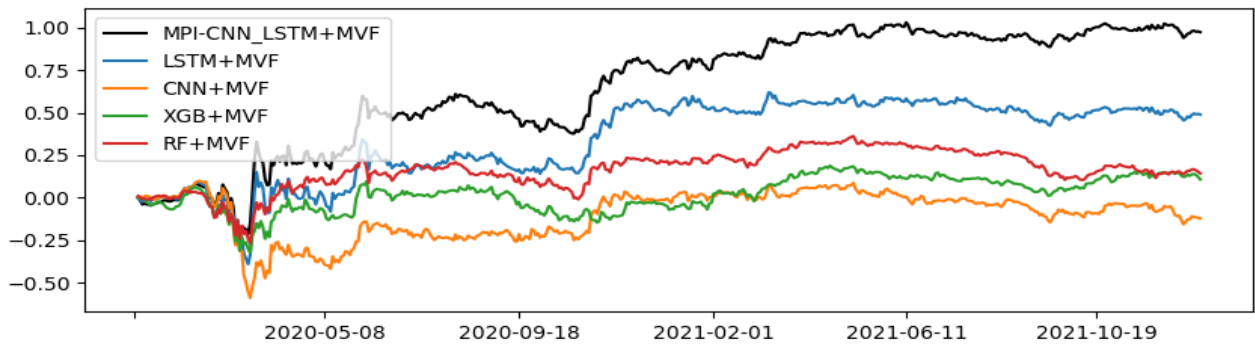


Fig 3. Cumulative return of the portfolio with 1% transaction cost

## 5. Conclusion

This study aims to develop the existing literature on portfolio construction with return prediction by introducing a different prediction method based on artificial neural networks that can predict the return of assets with less error. First, this paper compares the predictive abilities of deep learning models including MPI CNN-LSTM, CNN-LSTM, LSTM, and CNN, and machine learning models that include RF, SVR, and XGB and it was shown that between all, our proposed MPI CNN-LSTM based on MAE, MSE and MAPE metrics outperforms the other models. In the next stage, this paper discusses the performance of MVF with different

predictive models including our proposed MPI CNN-LSTM considering transaction fees, and applies daily and annual risk and return metrics to comprehensively measure their differences. Experiments' results present that MPI CNN-LSTM+MVF outperforms others. To better understand the performance of the built portfolios and compare their performance and identify the best model, the cumulative return charts have been drawn during the test period that through them can see the superiority of MPI CNN-LSTM+MVF over other models. Therefore, this paper recommends building MVF model with MPI CNN-LSTM return forecasts for daily trading investment.

## References

- [1] Markowitz, H.M. (1952) Markowitz, Portfolio selection. *The Journal of Finance*, 7(1) 77-91.
- [2] Wang, W., Li, W., Zhang, N., & Liu, K. (2020) Portfolio formation with pre-selection using deep learning from long-term financial data. *Expert Systems with Applications*, 143, 113042.
- [3] Zolfani, S.H., Taheri, H.M., Gharehgozlou, M., & Farahani, A. (2022) an asymmetric PROMETHEE II for cryptocurrency portfolio allocation based on return prediction. *Applied Soft Computing*, 131, and 109829.
- [4] Ta, V. D., Liu, C. M., & Tadesse, D. A. (2020) Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences*, 10, 437.
- [5] Paiva, F. D., Cardoso, R.T.N., Hanaoka, G.P., & Duarte, W.M. (2019) Decision making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, 115, 635-655.
- [6] Freitas, F.D., De Souza, A.F., De Almeida, A.R. (2009) Prediction-based portfolio optimization model

- using neural networks. *Neuro computing*, 72(10-12), 2155-2170.
- [7] Ma, Y., Han, R., Wang W. (2021) Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165, 113973.
- [8] Lu, W., Li J., Li, Y., Sun, A., & Wang, J. ( 2020) A CNN-LSTM-Based Model to Forecast Stock Prices, *Complexity*, vol, 6622927, 10 pages.
- [9] Yu, J.R., Paul Chiou, W.J., Lee, W.Y., & Lin, Sh.J. (2020) Portfolio models with return forecasting and transaction costs, *International Review of Economics & Finance*, Volume 66, Pages 118-130.
- [10] Zhou, F., Zhang, Q., Sornette, D., & Jiang, L. (2019) Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices, *Applied Soft Computing* 84, 105747.
- [11] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015) *Time series analysis: forecasting and control*. John Wiley & Sons.
- [12] Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. (2018) Predicting the direction of stock market prices using tree-based classifiers, *The North American Journal of Economics and Finance* . 47, 552–567.