Building an Optimal Investment Portfolio with Python Machine Learning Tools

Vitaliy Kobets and Serhii Savchenko

Kherson State University, 27 Universitetska St., Kherson 73003, Ukraine

Abstract

We have researched how machine-learning techniques can be applied in software tools for personal investment portfolio planning. This paper contains a brief overview of different types of machine learning algorithms and a description of the mathematical model used to build the optimal investment portfolio. The newly demanded and not implemented feature of Fintech software tools for personal investments (such as Robo-Advisor) is to forecast rates of financial instruments to quickly rebalance the structure of investment portfolio to increase potential income and decrease losses of investors. The price forecasting methods proposed in the paper can be used for this feature. During the experimental part of our research, we implemented a software tool that builds an optimal investment portfolio using price predictions created by LSTM neural network and linear regression method. The results were compared to a portfolio built using only historical data on financial instruments price. The research shows that using LSTM neural network forecasted values allows building better investment portfolios even during the global stock market recession.

Keywords [1]

Investment Portfolio, Automated Financial Software, Robo-Advisor, Machine Learning.

1. Introduction

Modern methods of machine learning are actively used in many areas of human activity. The most common use cases are optical character recognition, classification and clustering, time series forecasting, etc. Machine learning (ML) systems and artificial intelligence (AI) destroy routine jobs and create new human jobs. These jobs require new skills that have no analogues. Among these jobs are trainers, explainers and sustainers [1]. Trainers need to teach service chatbots how they should perform using ML need to detect the complexities of human communication and emotions to address client query with sympathy. Explainers can fill the gap between innovation technologists and business leaders. Decision making which is based only on “black box” nature of complex machine-learning algorithms can contradict to conventional wisdom, because EU’s General Data Protection Regulation creates a “right to explanation” for investor. Sustainers can help ensure that AI and ML systems operate as designed and that unintended consequences (such as unfairness, disauditability, discrimination etc.) are considered with the appropriate urgency [1]. All machine learning algorithms can be divided into three major categories: supervised, unsupervised, and reinforcement learning. Table 1 contains a brief comparison of different types of machine learning algorithms [2].

This paper is devoted to a review of the possibilities of using supervised machine learning techniques for the task of building an optimal investment portfolio for long-run investments of a client. This task is one of the features of software tools that help people to manage their personalized investment and insurance portfolios. Such software tools are called Robo-Advisors (RA). In one of our previous works [3] we have already developed software architecture and different modules of RA system. The practical outcome of this paper is the implementation of profitability forecasting and investment portfolio formation modules of RA system.

Information Technology and Implementation (IT&I-2022), November 30 – December 02, 2022, Kyiv, Ukraine
EMAIL: vkobets@kse.org.ua (A. 1); savchenko.serhii@gmail.com (A. 2);
ORCID: 0000-0002-4386-4103 (A. 1); 0000-0001-9706-5334 (A. 2);

Table 1
Comparison of different machine learning algorithm types.

Type		Description	Algorithms examples
Supervised learning	Classification	Determining whether each element belongs to one of the known categories based on the parameters of the object.	Random Forest Decision tree Linear SVM
	Regression	Defining relationships among variables and their weights.	Linear Regression Gradient Boosting Tree
	Forecasting	Predictions about the future based on present data.	Linear Regression Neural Network
Unsupervised learning	Clustering	Grouping elements to sets where all elements are similar to each other by defined criteria.	K-means K-modes DBSCAN
	Dimension reduction	Reducing the number of variables without significant data loss.	Singular Vector Decomposition
Reinforcement learning		Search for the best solution using step-by-step parameters optimization.	Deep Learning Recurrent Neural Networks

The **purpose** of this paper is to develop and implement a software tool for building an optimal investment portfolio using machine-learning techniques for time series of financial instruments' profitability predictions.

Research objectives include the examination of which machine learning techniques can be applied to financial instruments' price prediction and profitability analysis of a portfolio built using only historical data with portfolios built taking into account price predictions.

The paper is structured as follows: Part 2 addresses the literature review. Part 3 describes the theoretical model behind the optimal investment portfolio generation and the experimental part of the research. Part 4 presents obtained results. Finally, the last part concludes.

2. Literature Review

Prediction of risk issues can be done using both classical machine learning (ML) and deep learning (DL) techniques such as random forest, convolutional neural networks (CNNs) and long short-term memory (LSTM). Although deep learning models (DL) are good prediction systems, it must be confirmed whether in this field they behave better than other machine learning (ML) techniques. Classical Python libraries like pandas is used for this pre-processing of food security data [4]. Using probabilistic predictions as an advantage of neural models can optimize the number of inspections. The accuracy results of the neural models versus the average accuracy of the non-neural models shows a clear advantage of neural models [4]. Authors [5] proposed such measures of quality estimation as root mean square error and mean absolute percentage error associated to the ML models.

Other authors [6] who applied ML as well as traditional econometric methods found that overall prediction accuracy ranges between 60% and 70% for their selected ML and non-ML methods. They found that the ML and non-ML methods perform quite similarly in predicting overall accuracy, where non-ML method consists of either OLS or logistic regression. Which methods are likely to work best will depend on the pool of available predictor sets as well as the complexity of the functional forms linking predictors to outcomes [6]. Authors of investigation [7] reveal poor performance associated with the LSTM and disclosure that the CNN is also complicated because of its black box effect, which makes it also unsuitable for the use. Based on studied online resources, we can define next types of software tools that can be helpful for novice investors [8, 9]:

- Financial calendars
- Currency converters
- Trading calculators (Fibonacci calculator, pivot point calculator, etc.)
- Trading tools (real-time charts, historical datasets, indicators, indexes)
- Personal finance tools (budgeting, mortgage calculators, tax tools)

- Robo-Advisors

Robo-Advisors are complex software tools, usually presented as web applications, which provide all types of services for personal investments [10]. The common features that persist in Robo-Advisor system include a questionnaire module to determine the type of client, personalized investment portfolio formation, portfolio rebalancing, access to an information dashboard, and additional tools.

Linear regression (LR) is a classical machine learning technique that is used for modeling the correspondence between some dependent variable and one or more explanatory (independent) variables. Linear regression also shows which explanatory variables have more impact on the dependent variable value. It could be useful in case if we have a lot of input data and we want to reduce the number of inputs. We also used Long Short Term Memory (LSTM) neural network for predicting future price of different financial instruments. LSTM neural network is a kind of Recurrent Neural Network (RNN) that has special memory neurons, therefore they can store, filter and pass long-run dependencies from one layer to another. Such feature of LSTM neural networks makes them especially good in time series forecasting (e.g., return rates and standard deviations or risks of financial instruments).

Currently, using LSTM neural networks to predict prices for stocks, currency pairs, and other financial instruments is being actively studied. The study by A. Staffini [11] describes the way of using deep learning approaches for stock price forecasting. The paper is devoted to using Convolutional Generative Adversarial Network (GAN) for predicting stock time series. GAN is a deep learning approach, that consists of a generator model and discriminator model, and LSTM network could be used as GAN generator which generated predictions which then discriminator model evaluates. Another study [12] describes the method for predicting stock market index based on such inputs as historical data, macroeconomic data (Civilian Unemployment Rate, Consumer Sentiment Index, and US dollar index), and technical analysis indicators (Moving Average Convergence Divergence, Average True Range, Relative Strength Index).

Application of machine learning algorithms to predict return and risk metrics of financial instrument and corresponding development of backend development of web applications requires powerful programming language. Python is an open source general purpose programming language. According to the TIOBE Programming Community index, Python is the most popular programming language [13]. It has many third-party libraries, including those designed for machine learning and data analysis. Python supports almost all most popular machine learning frameworks, such as TensorFlow, Keras, MXNet, Theano, etc. [14]. Since Python is also suits for the backend development of web applications, the integration of machine learning modules will not cause difficulties.

3. Research Methodology

3.1. Model

We used the Markowitz portfolio model for generating optimal investment portfolio. Markowitz portfolio model is a mathematical framework that solves the optimization problem of assembling a portfolio where expected return is maximized for a given level of risk (1) or when the level of risk is minimized for a given level of expected return (2) [15-17].

$$\begin{cases} R_p \rightarrow max \\ \sigma_p \leq \sigma_g \\ w_1 + w_2 + \cdots + w_N = 1 \\ w_i \geq 0 \end{cases} \quad (1)$$

$$\begin{cases} R_p \geq R_g \\ \sigma_p \rightarrow min \\ w_1 + w_2 + \cdots + w_N = 1 \\ w_i \geq 0 \end{cases} \quad (2)$$

where N is a number of assets, R_p is an expected return of the portfolio, σ_p is a level of risk (standard deviation), and w_i is a percentage of asset i in portfolio p. For a risk neutral investor, we need to define a target function which minimizes the ratio of risk to profit (3).

$$\begin{cases} \dfrac{\sigma_p}{R_p} \rightarrow min \\ w_1 + w_2 + \cdots + w_N = 1 \\ w_i \geq 0 \end{cases} \quad (3)$$

The experimental part of this research includes comparison of the profitability of three portfolios. The first one is built based on the historical data on Close prices of different financial instruments for two years. We chose the following set of stocks for our experiment: AAPL, GOOG, MSFT, AMZN, INTC, AMD, NVDA, F, TSLA, JPM, MS, VOO. The set includes shares of companies from different sectors of the economy, such as high technology, microelectronics, engineering, banking, and finance. Data on the price history of all the companies listed above can be found in the public domain on the Yahoo Finance service. Close prices include all information after opening of assets trades, so close price is more relevant than open prices, maximal and minimal prices. Another two portfolios are built using additional data with forecasted Close prices for the next month. Predictions of the Close price for the next month were made for each financial instrument using the LSTM neural network and linear regression method. Comparisons of the obtained results are presented in Part 4. We used the Anaconda Data Science Platform for our machine learning tasks. Anaconda is a software distribution of the Python and R programming languages for scientific computing, including development, data processing, and visualization tools. It also aims to simplify package management and deployment on any operational system. The list of libraries that were used during development is presented in Table 2.

Table 2
Used machine learning and data processing libraries.

Library	Description
yfinance	Allows to download market data from Yahoo! Finance
pandas	Library used for data analysis and manipulating numerical tables.
numpy	Library with a large collection of high-level mathematical functions and tools for operations with multi-dimensional arrays and matrices.
matplotlib	Library for static, animated, and interactive visualizations.
keras	Open-source deep learning library.
sklearn	Open-source machine learning library.
pypfopt	Python library that provides portfolio optimization methods.

3.2. Program Architecture

Experimental part of this research consists of the following stages:

- Collecting historical data from open sources.
- Initial data processing. Removing rows with missing values and data normalization.
- Generate predictions on next month's Close price for each financial instrument using LSTM neural network and Linear Regression.
- Building an optimal investment portfolio based only on historical data. Building an optimal investment portfolio based on historical data and next month's predictions.
- Comparing the effectiveness of three portfolios.

Here is a brief overview of the experimental part of our research. Listings 1 through 4 correspond to the first four steps of the workflow. We have used historical data on twelve stocks monthly close prices for ten years to train our ML models. After that, we got a prediction on next month's close price for each ticker. All data is retrieved using yfinance library which returns values as pandas dataframe object. After all data is loaded, it should be filtered from null values and scaled to values from 0 to 1. On the next code snippet, you can see the main function that accepts a dataframe object and a ticker name. This function splits data to train and test datasets, creates an LSTM model, trains in on train dataset and evaluates root mean squared error (RMSE), and returns prediction on next month Close price for a given ticker. The dataset was divided into training and test parts in a ratio of 3 to 1.

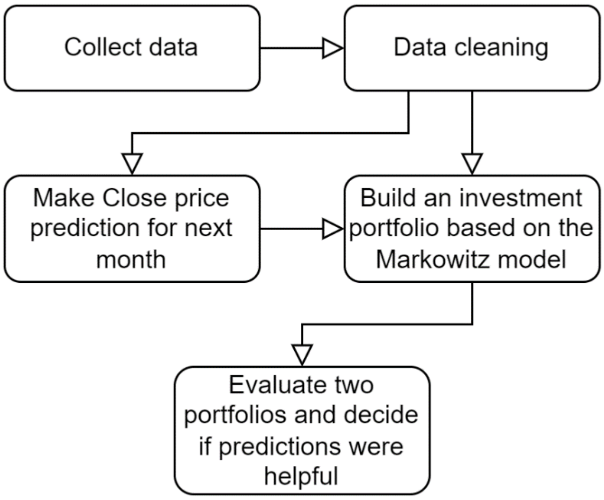

Figure 1: Workflow diagram.

```
tickers = ['AAPL', 'GOOG', 'MSFT', 'AMZN', 'INTC', 'AMD',
           'NVDA', 'F', 'TSLA', 'JPM', 'MS', 'VOO']

def get_month_data_from_yf(ticker, start_date, end_date):
  data = yf.download(ticker, start_date, end_date, interval='1mo', progress=False)
  return data.dropna()

data_ten_years = {}
for ticker in tickers:
  data_ten_years[ticker] = get_month_data_from_yf(ticker, '2012-01-01', '2022-01-01')
```

Listing 1: Loading data from Yahoo Finance.

To determine the final prediction, we used the average of three independent LSTM model predictions. Each iteration has random initial weights; network is trained for 100 epochs. The result of each prediction may vary due to the stochastic nature of the algorithms used during the training process of the neural network. We have also defined a separate function for building a prediction using linear regression approach. Here we have no need to run training process more than once, because this algorithm has no randomized parameters. Previous month close price was determined as explanatory variable. The source code of the function is presented in listing 4. The last listing shows the usage of pypfopt library, that generates an optimal (in terms of min volatility) investment portfolio based on monthly Close price data for two years plus one value with prediction on next month Close price, which we've generated by LSTM network or LR model. All source code is available on GitHub via the link https://github.com/serhii1savchenko/invest-portfolio-generator.

4. Results

Table 3 presents the results of the accuracy assessment of the LSTM and LR models on the test dataset by the RMSE score. Fourth and sixth columns show the ratio of the RMSE score to the value of the last known Close price. The most minor deviations for LSTM Close price predictions were obtained for tickers VOO, GOOG, and MSFT. The lowest deviations for LR predictions also were obtained for the same tickers. In two-thirds of the cases, the LR method gave better prediction results, however, the difference is not very high. Therefore, we have built investment portfolios using both LSTM and LR predictions.

```
def predict_nmcp(dataset, ticker):
    ...
    model = Sequential()
    model.add(LSTM(4, input_shape=(1, look_back)))
    model.add(Dense(1))
    model.compile(loss='mean_squared_error', optimizer='adam')
    model.fit(trainX, trainY, epochs=100, batch_size=1)
    ...
    prediction = model.predict(featureX)
    last_real_data_scaled = numpy.concatenate(
            (last_real_data_scaled, prediction), axis=0)
    next_month_inversed = scaler. inverse_transform(prediction)[0][0]
    return next_month_inversed
```

Listing 2: Main method for next month close price prediction using LSTM.

```
def wcp_pred(data, ticker):
    prediction_1 = predict_nmcp(data, ticker)
    prediction _2 = predict_nmcp(data, ticker)
    prediction _3 = predict_nmcp(data, ticker)
    return (prediction _1 + prediction _2 + prediction _3) / 3

predictions = {}
for ticker in tickers:
    nmcp = wcp_pred(data_ten_years, ticker)
    predictions[ticker] = nmcp
    print(ticker, "PREDICTION =", nmcp)
```

Listing 3: Determination of the final prediction by three independent runs.

```
def mlr_predict_close_price(data, ticker):
    dataset = data[ticker] X = dataset[['Close-Previous-Month']].values
```

```
    y = dataset['Close'].values
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33,
                            random_state = 0, shuffle = False)
    regressor = LinearRegression()
    regressor.fit(X_train, y_train)
    y_pred = regressor.predict(X_test)
    next_month_pred = regressor.predict([[y[len(y)-1]]])
    rmse = sqrt(mean_squared_error(y_test, y_pred))
    return [next_month_pred, rmse]
```

Listing 4: Next month close price prediction using linear regression.

. The distribution of the shares of financial instruments in the portfolio built exclusively on historical data (Portfolio 1), the portfolio where the LSTM forecast for the next month was used (Portfolio 2), and the portfolio where the LR forecast was used (Portfolio 3) is presented in Table 4. All portfolios are built for risk-neutral investors. This is achieved by introducing a condition for minimizing the level of allowable volatility (listing 5, line 13). Based on these weights we calculated the dynamics of changes in the value of each of the portfolios during the first half of 2022. The initial balance of each portfolio is USD 100,000.

```
data = yf.download(tickers, start='2020-01-01', end='2022-01-01', interval='1mo')
data = data['Close'] .dropna()
d = datetime.datetime.strptime("01/01/2022","%d/%m/%Y")
data_with_prediction = data.append(pandas.DataFrame(index=[d]))
for ticker in tickers:
   data_with_prediction.loc[d, ticker] =  predictions_next_month_close_price[ticker]
mu = mean_historical_return(data_with_prediction, frequency=12)
S = CovarianceShrinkage(data_with_prediction,
          frequency=12).ledoit_wolf()
ef = EfficientFrontier(mu, S, weight_bounds=(0,1))
weights = ef.min_volatility()
cleaned_weights = ef.clean_weights()
print(cleaned_weights)
ef.portfolio_performance(verbose=True)
```

Listing 5: Investment portfolio generation.

Table 3
LSTM and LR predictions results.

Ticker	Dec 2021 Close Price	LSTM test dataset RMSE	Deviation percentage	LR test dataset RMSE	Deviation percentage
AAPL	177.57	10.3	5.8%	8.21	**4.62%**
GOOG	144.67	6.1	**4.22%**	6.18	4.27%
MSFT	336.32	14.4	4.28%	11.84	**3.52%**
AMZN	166.71	10.61	**6.36%**	11.29	6.77%
INTC	51.5	4.02	**7.81%**	4.16	8.08%
AMD	143.89	9.53	**6.62%**	9.97	6.93%
NVDA	294.1	19.63	6.67%	17.37	**5.9%**
F	20.77	1.23	5.92%	1.18	**5.68%**
TSLA	352.26	39.88	11.32%	30.07	**8.53%**
JPM	158.35	9.62	6.08%	8.95	**5.65%**
MS	98.16	5.63	5.74%	5.35	**5.45%**
VOO	436.57	16.7	3.83%	15.17	**3.47%**

As we can see, all three portfolios lost in price, but the portfolio made taking into account LSTM price predictions shows better dynamics (less loss). Also this portfolio has the smallest difference between the expected and real value of 6-month income. Portfolio 1 (which was built only on historical dataset) has the biggest difference between the expected and real return. Figure 2 is a graph showing the percentage difference in the loss of the first portfolio relative to the second. For example, as of February 2022, Portfolio 1 lost USD 9653.74 while Portfolio 2 lost USD 9432.89. Thus, the portfolio made taking into account the forecast received a loss of 2.29% less than the base portfolio.

5. Conclusions

Thus, the paper contains a brief overview of different machine learning approaches which can be used in the scope of personal finance software tools. During the experimental part of our research, we developed a Linear Regression model and LSTM neural network that generate a prediction on next month's close price of defined financial instrument. The predictions are based on historical data of the financial instrument. We have also demonstrated that an investment portfolio that is build used not only historical data but also one-month prediction shows better results even during the global recession period. The paper doesn't cover more sophisticated deep learning approaches for time series predictions. We are going to expand current research on using more complex LSTM models, which includes the windows method, multivariate time series forecasting, and using other methodologies for an investment portfolio building, such as Black-Litterman portfolio optimization model.

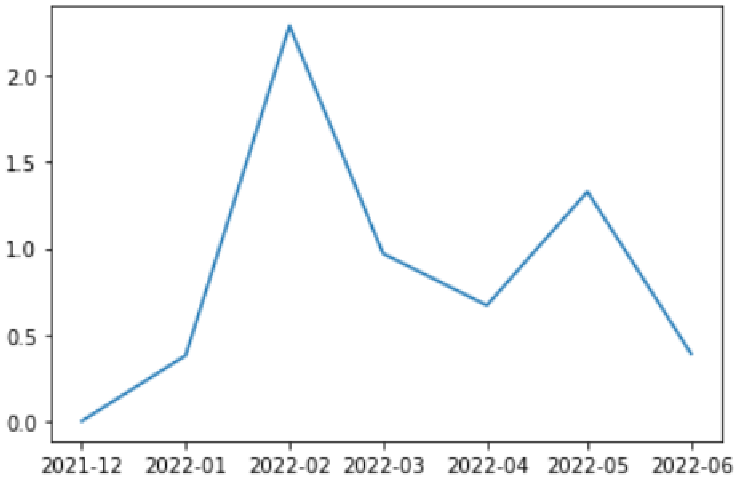

Figure 2: The difference in the value of the two portfolios.

Table 4
The distribution of the shares of financial instruments.

Ticker	Portfolio 1	Portfolio 2	Portfolio 3
AAPL	0	0	0
GOOG	0.05891	0.06402	0.05887
MSFT	0.17157	0.15357	0.1716
AMZN	0.14382	0.15413	0.14413
INTC	0.23247	0.25194	0.23252
AMD	0.00892	0.02014	0.00925
NVDA	0.09703	0.07661	0.09668
F	0.03027	0.02581	0.03003
TSLA	0	0	0
JPM	0.10737	0.1044	0.10767
MS	0	0	0
VOO	0.14964	0.14938	0.14926
Total sum	1	1	1

Table 5
The dynamics of changes in the value of each of the portfolios.

Portfolio	2022-01	2022-02	2022-03	2022-04	2022-05	2022-06
Portfolio 1	92412.17	90346.26	93524.80	79230.69	79893.63	70498.47
Portfolio 2	92440.91	90567.11	93587.40	79369.42	80160.87	70612.75
Portfolio 3	92412.64	90349.47	93528.66	79237.3	79900.72	70508.11

Table 6
Comparison of expected and real return rate.

Portfolio	Expected 6 month return	Real 6-month return
Portfolio 1	15.65%	-20.11%
Portfolio 2	12.65%	**-19.84%**
Portfolio 3	15.6%	-20.1%

6. References

[1] Wilson, H. J., Daugherty, P. R. and Morini-Bianzino, N. (2017). The Jobs that Artificial Intelligence will Create. MITSloan Management Review. Vol. 58 (4) URL: http://ilp.mit.edu/media/news_articles/smr/2017/58416.pdf

[2] Wakefield, K. A guide to the types of machine learning algorithms and their applications. URL: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html

[3] Savchenko, S., Kobets, V. (2021). Development of Robo-Advisor System for Personalized Investment and Insurance Portfolio Generation. ICTERI 2021 Workshops. ICTERI 2021. Communications in Computer and Information Science, vol. 1635, 2022. doi: 10.1007/978-3-031-14841-5_14

[4] Nogales, A., Díaz-Moron, R., García-Tejedor, A. J. (2022). A comparison of neural and non-neural machine learning models for food safety risk prediction with European Union RASFF data. Food Control, 134, 108697. doi: 10.1016/j.foodcont.2021.108697

[5] Aworka, R., Cedric, L.S., Adoni, W.Y.H., Zoueu, J.T., Mutombo, F.K., Kimpolo, C.L.M., Nahhal, T., Krichen, M. (2022). Agricultural decision system based on advanced machine learning models for yield prediction: Case of East African countries. Smart Agricultural Technology, 2, 100048. doi: 10.1016/j.atech.2022.100048

[6] Hossaina, M., Mullally, C., Asadullah, M.N. (2019). Alternatives to calorie-based indicators of food security: An application of machine learning methods. Food Policy, 84 , 77–91. doi: 10.1016/j.foodpol.2019.03.001

[7] Deléglise, H., Interdonato R., Bégué A., d'Hôtel, E.M., Teisseire M., Roche M. (2022). Food security prediction from heterogeneous data combining machine and deep learning methods. Expert Systems With Applications 190, 116189. doi: 10.1016/j.eswa.2021.116189

[8] Friedberg, B. A. 6 Best Portfolio Management Software Tools for Investors – 2022. URL: https://www.roboadvisorpros.com/best-portfolio-management-software-for-investors/

[9] Trading and Investment Tools. Investing.com. URL: https://www.investing.com/tools/

[10] Waliszewski, K., Zięba-Szklarska, M. (2020). Robo-advisors as automated personal financial planners – SWOT analysis. Journal of Finance and Financial Law, 3(27), pp. 155–173. doi: http://dx.doi.org/10.18778/2391-6478.3.27.09.

[11] Staffini, A. (2022). Stock Price Forecasting by a Deep Convolutional Generative Adversarial Network. Front Artif Intell, vol. 5, 2022, doi: 10.3389/frai.2022.837596.

[12] Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., Khatri, R. K. C. (2022). Predicting stock market index using LSTM. Machine Learning with Applications, vol. 9, 2022100320, doi: 10.1016/j.mlwa.2022.100320.

[13] TIOBE Index for September 2022. URL: https://www.tiobe.com/tiobe-index

[14] Project Pro: 15 Popular Machine Learning Frameworks to Manage Machine Learning Projects. URL: https://www.projectpro.io/article/machine-learning-frameworks/509

[15] Mangram, M. E. (2013). A Simplified Perspective of the Markowitz Portfolio Theory. Global Journal of Business Research, vol. 7 (1), 2013, pp. 59-70. URL: https://ssrn.com/abstract=2147880

[16] Snihovyi, O. , Kobets, V., Ivanov, O. Implementation of Robo-Advisor Services for Different Risk Attitude Investment Decisions Using Machine Learning Techniques. Communications in Computer and Information Science, 2019, vol. 1007, pp. 298–321. doi: 10.1007/978-3-030-13929-2_15

[17] Snihovyi, O., Ivanov, O., Kobets, V. Implementation of Robo-Advisors Using Neural Networks for Different Risk Attitude Investment Decisions 9th International Conference on Intelligent Systems 2018: Theory, Research and Innovation in Applications, IS 2018 - Proceedings, 2018, 8710559, pp. 332–336. doi: 10.1109/IS.2018.8710559