Information system for generating recommendations for risk-oriented trading strategies based on deep learning

Nickolay Rudnichenko[1], Vladimir Vychuzhanin[1], Denys Shvedov[1], Tetiana Otradskya[1] and Igor Petrov[2]

[1]*Odessa Polytechnic National University, 1 Shevchenko Ave., Odessa, 65001, Ukraine*
[2]*National University "Odessa Maritime Academy", 8 Didrichson Str., Odessa, 65029, Ukraine*

Abstract
This article is devoted to the issue of formalization and description of technical aspects of development of an information system for generating recommendations for risk-oriented trading strategies on stock exchanges based on the use of deep learning models. The study used a data set representing information on exchange trading in Apple assets obtained from the Yahoo Finance system. The concept of a software system including three functionally independent modules was developed, their formal schematization was carried out. A project of the system with the construction of a diagram of the main components displaying the relationships between the elements was created. In the PyCharm development environment, a structure of directories and files was developed to organize the system software. A graphical user interface with interactive widgets was implemented, providing opportunities for entering, processing and visualizing data. An analysis of the work of the developed modules was carried out, including a description of the strategic recommendations they generate for making trading decisions. The obtained results were interpreted, their key features were identified. Promising areas of further research were determined and possible ways to improve the system were outlined.

Keywords
data analysis, recommendation systems, financial risk trading, deep learning, decision support, information systems development

1. Introduction

In the modern financial market, including the sphere of trading assets, there is a growing interest in participating in exchange trading, which is becoming a current trend among investors [1]. However, the decision-making process in this area is associated with numerous risks and difficulties, which are caused by various factors [2]. The volatility of the value of financial instruments is determined by the influence of difficult-to-formalize factors, such as the dynamics of interest rates, socio-geopolitical crises, as well as the instability of macro- and microeconomic ecosystems, which affects the behavior of market participants [3]. An important aspect is the formation of optimistic and pessimistic risk-oriented strategies for the target management of financial trading decisions. The situation is aggravated by the increase in the volume of data generated by exchanges, including information on price quotes, trading volumes, economic indicators and corporate reports [4]. Such an array of data creates additional pressure on traders, increasing the likelihood of errors associated with subjective factors of human perception [5]. In this regard, the key factors determining the success of investments are the efficiency and accuracy of risk-oriented analytics [6].

In order to effectively perform analytical procedures and compete with institutional investors to achieve target results in the formation of financial decisions, there is an increasing need for specialized tools that can automate the process of analyzing alternatives, identify trends and evaluate trading strategies [7].

CS&SE@SW 2024: 7th Workshop for Young Scientists in Computer Science & Software Engineering, December 27, 2024, Kryvyi Rih, Ukraine
nickolay.rud@gmail.com (N. Rudnichenko); vint532@gmail.com (V. Vychuzhanin); studylearner@gmail.com (D. Shvedov); tv_61@ukr.net (T. Otradskya); firmn@gmail.com (I. Petrov)
0000-0002-7343-8076 (N. Rudnichenko); 0000-0002-6302-1832 (V. Vychuzhanin); 0009-0002-4823-8782 (D. Shvedov); 0000-0002-5808-5647 (T. Otradskya); 0000-0002-8740-6198 (I. Petrov)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

Traditional methods of technical analysis often do not provide the necessary level of adaptability and accuracy in the conditions of the modern financial environment [8]. This explains the growing interest in the implementation of machine learning (ML), deep learning (DL) and artificial neural networks (ANN) methods [9]. These approaches allow you to automate the processing of large volumes of data, identify complex and non-obvious patterns, and create technological solutions to support trading decisions [10].

Thus, the development of our own analytical system in the field of issuing recommendations on risk-oriented strategies for conducting exchange trading is due to the need to improve the accuracy, speed and adaptability of decisions made. A promising direction is the integration of ML and DL methods to create comprehensive decision support tools [11].

Modern models of support for making trading decisions based on the use of ML and DL algorithms have been actively integrated in recent years to solve applied problems in the field of financial management and accounting, including on stock markets in order to automate data analysis processes, predictive analytics and improve the quality of choosing trading strategies. A feature of the data on trading systems is their temporal nature, i.e. the variability of parameters (price, demand, volatility, etc.) over time. In this regard, it should be noted that time series analysis methods are appropriate for use in short-term forecasting, which is due to the nature of the change in the forecast horizon, an increase in which may lead to the need to calculate new values based on the obtained indicators.

This is acceptable at some time intervals, but the accuracy can be significantly reduced, especially in the case of an increase in the forecast horizon [12]. At the same time, an important factor is the presence of a number of limitations, including those consisting in the need to adapt the planning conditions to stationary processes, i.e. ensuring a static probability distribution. A feature that must be taken into account when choosing ML and DL models and their parameterization for solving repressed models is the fact that time series in real economic, financial, marketing and trade problems are most often non-stationary and homogeneous. In fact, this means that the residual formed by subtracting a non-random component from a series is a stationary time series [13].

A feature that must be taken into account when choosing ML and DL models and their parameterization for solving repressed models is the fact that time series in real economic, financial, marketing and trade problems are most often non-stationary and homogeneous. In fact, this means that the residual formed by subtracting a non-random component from a series is a stationary time series [14].

In addition to classical ML methods used for such problems, the ARIMA, GARCH approaches and ANN models with efficient memory (for example, LSTM architecture) should be especially noted. The ARIMA model is the result of a kind of generalization of the autoregressive moving average model, its purpose is to build forecasts of time series of different types, in practice it is actively used to perform procedures for analyzing the value of shares and various financial indicators [15].

This model is a type of autoregressive models, the essence of which is to organize the process of calculating forecast values at fixed points in time based on previously obtained values. An important aspect of the operation of such models is the study of not the values of the process directly, but focusing on modifying its indicators relative to each other. Smoothing outliers of the series is carried out on the basis of the moving average approach by replacing the original calculated value with the arithmetic mean of the members closest to it [16]. Thus, in addition to classical MO models, there is a range of different models, the combination and comparison of which with each other in order to determine the most accurate and possessing the generalizing ability of the maximum level can be a promising area of research.

The aim of the work is to develop a project and software implementation of an information system for issuing recommendations on risk-oriented target trading strategies.

2. Results

2.1. Dataset description

In this paper, given the specifics of the topic under consideration, the choice of the Yahoo Finance (YF) platform as the initial data source seems reasonable and rational [17]. This resource provides access to reliable and detailed information on financial markets, providing a wide range of analytical capabilities. YF data can be used to forecast price trends, analyze financial statements, study the impact of economic events and monitor the news background.

To effectively work with YF data, it is advisable to use the yfinance library, which is implemented in the Python programming language. This tool provides a convenient interface for loading and processing financial information. The library allows you to extract a wide range of data, including:

- historical data on stock prices for a certain period (Open, High, Low, Close, Adjusted Close, Volume);
- real-time trading information (if available);
- company financial statements;
- information on dividends and stock splits;
- metadata on sectors, industries and option chains;
- currency rates and ETF data.

Thus, using the yfinance library in combination with Yahoo Finance data provides the necessary functionality for conducting complex financial market analysis, including developing predictive models and evaluating asset management strategies.

2.2. The implementation of the system concept

The system project was developed using the Python programming language in the PyCharm integrated development environment. To work with the data, libraries were used that provide processing, analysis and visualization (NumPy, Pandas, Seaborn, and Sklearn), as well as tools for implementing machine and deep learning methods (TensorFlow, Keras, Sklearn). The keras and pickle formats, as well as the joblib library, are used to save the created models. The recommendation generation system consists of three autonomous data analysis modules, the results of which are integrated to form risk-oriented trading strategies:

- Trading indicator analysis module for generation signals for opening and closing trades based on the intersection of two moving averages (SMA, EMA), also RSI, MACD and Bollinger Bands indicators are taken into account in proposed system;
- ARIMA-based time series analysis module (ARIMA module), which supports the creation of basic models, automatic parameter selection and seasonal models development;
- DL module, which usefull for building stock price forecasting models for a given time horizon.

The interaction between the modules is presented as a diagram (figure 1). Data preparation is carried out via API, which includes selecting information from Yahoo Finance based on specified parameters (time interval, ticker type), saving in a separate collection, trend analysis, calculating and displaying closing price values for the period, and taking into account the capitalization of splits in a specified time period. Based on the generated data, a preliminary study is carried out using a number of approaches:

- Moving average methods for smoothing out price changes and identify trends in the data. It allow us to identify optimal moments to enter or exit transactions.
- ARIMA models used for data analyzing with different trends (seasonality and pronounced) and future trades prices and volumes prediction.

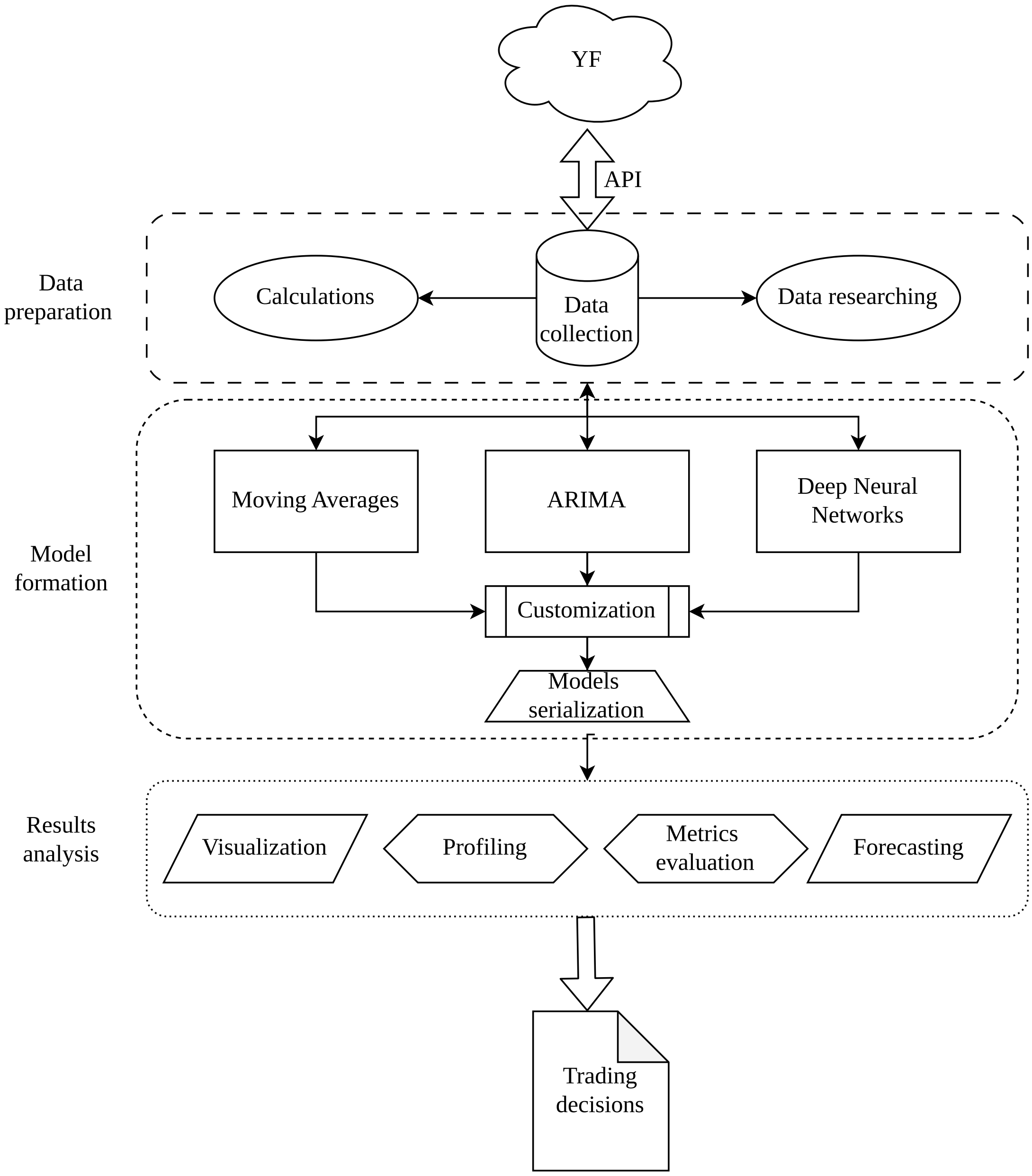

Figure 1: System structure.

- Deep learning models. Designed to predict possible market behavior scenarios. After creation, the models are adapted to the required time ranges and data parameters, which affects the accuracy of their operation. The final models are serialized and saved in pickle or keras format for subsequent use.

The models can then be loaded and used to perform analytical tasks separately from the working environment in which they were created. The next step is to evaluate the results of the generated models, including by visualizing two-dimensional graphical diagrams to highlight the necessary context of the obtained data estimates, profiling the model testing processes, assessing the quality metrics of the models, and performing the forecast procedure, which together are used to support the adoption of trading decisions on the principle of “buy/sell/wait”.

2.3. System project structure

The user of the first module has the ability to analyze the loaded data, perform calculations and evaluate the values of indicators (SMA, EMA, RSI, MACD), interpret the results in graphical form, and save text results of data analysis and generated visualizations. For the ARIMA model generation module, the general list of use cases is similar to the previous module, but a number of additional aspects have been implemented.

In particular, the user enters a ticker, time range and input parameters for building ARIMA models (simple, with automatic parameter selection and seasonal), he also has the ability to initialize the process of creating models, estimating error values based on the use of the RMSE metric, performing forecast operations, displaying analysis metrics and recommendations in the interface, including their output based on forecasts, as well as functions for serializing models of analysis results into object and text files, visualizations in graphic files. The diagram of the main component of the system is shown in figure 2.

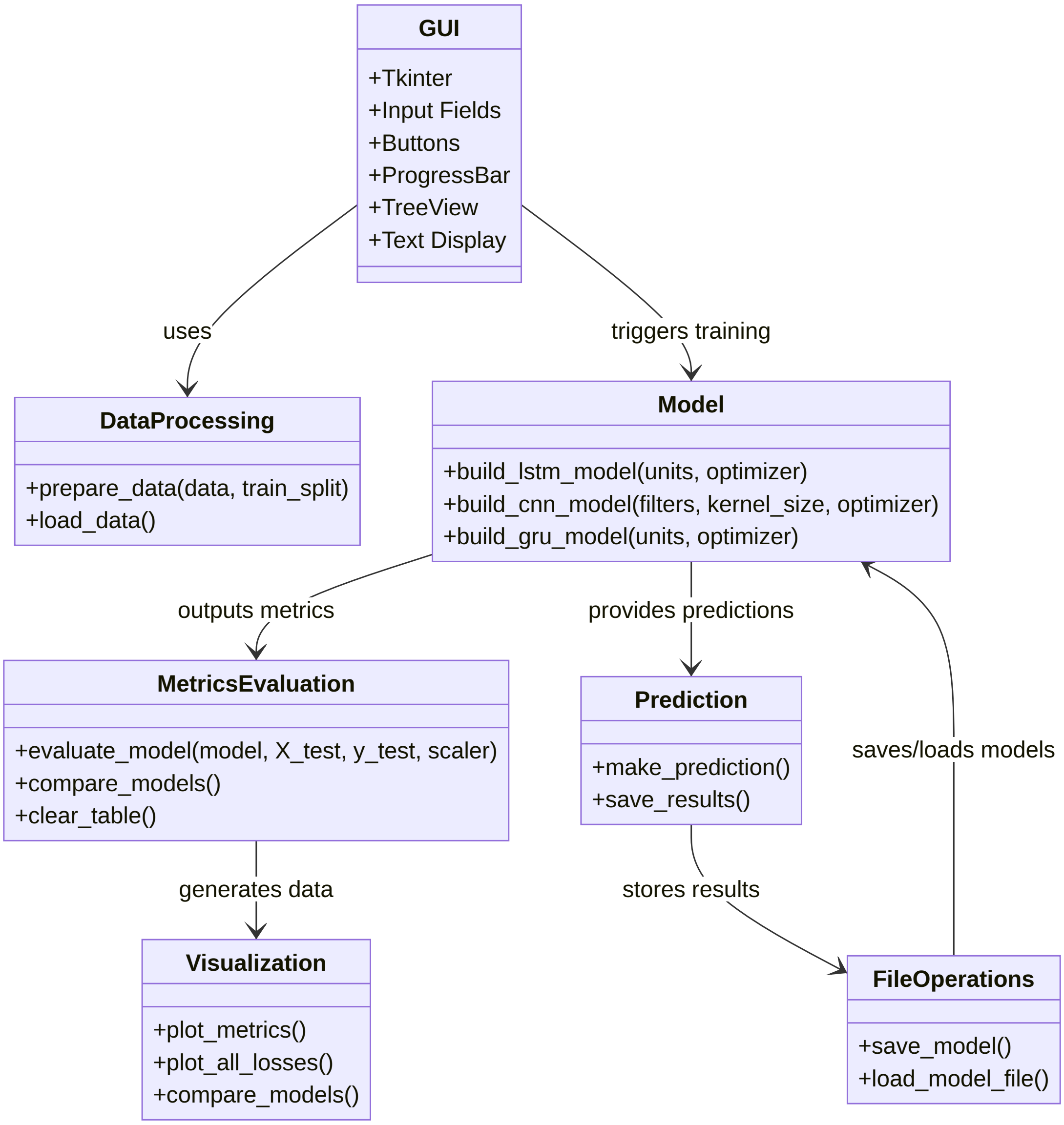

Figure 2: The diagram of the main component.

As part of the use of the DL module, the user is provided with the following options: entering input data; dividing the sample into proportions for the training and test subsets; creating DL models (LSTM, CNN, GRU); training and testing DL models; outputting values of model accuracy evaluation metrics;

building visualizations of loss values; evaluating the speed of creating DL models; performing forecasts and issuing recommendations; saving visualizations; saving DL model objects to keras objects. The key components, which are blocks of individual Python functions, are:

- GUI, provides various functionality for displaying widgets and special controls and user interaction with the application interface (including processing button clicks, adding data to tables, entering values and text fields);
- DataProcessing, performs data processing procedures, loading them and preparing them for the training and testing of models for their evaluation;
- Model, directly builds DL models (LSTM, CNN, GRU);
- MetricsEvaluation, calculates model quality metrics and compares them with each other;
- Prediction, implements the process of starting to build forecasts and providing the results of model operation;
- Visualization, performs procedures for constructing visual diagrams and graphs for analyzing the results of model operation, including evaluating their metrics;
- FileOperations, implements the functionality of serializing data and objects by saving and loading the created DL models and the results of their forecasts.

The graphical interface of the system is implemented as a desktop software application, with various widgets from the Tkinter library used as interface elements.

The interface includes buttons for training models, loading and saving to files, performing a forecast, plotting a graph based on loss values, saving forecast results and comparing models with each other, as well as clearing tables.

A separate table with scrolling was created to specify the hyperparameter values for each created model.

2.4. Results analysis and discussion

Based on the implemented modules of the system, a study and analysis of the results of practical use was performed, in particular, the module for assessing trading indicators. The conclusions made based on the results of the analysis are supported by visualizations, an example of which is shown in figure 3, below is an interpretation of the key results.

The RSI analysis revealed several key features. The RSI indicator more often reaches values corresponding to the overbought area, which indicates more frequent price declines in short time periods. Such dynamics indicate short-term corrections, typical for highly volatile markets. In some parts of the time series, there is a weak divergence between the RSI values and price movements. This may indicate a weakening of the current trend and the possibility of a reversal.

In the process of using the DL module, various models of artificial neural networks (ANN) were built, after which their results were compared by the loss metric, as shown in figure 4.

Main conclusions from the analysis:

- during training over 100 epochs, there is practically no overfitting effect. This indicates a good generalizing ability of the models;
- the GRU model demonstrates the fastest decrease in the loss metric values, already in the first 3 training epochs;
- the LSTM model stabilizes last of all, around the 20th epoch.

Comparison of the training speed of the models:

- the convolutional ANN model trains the fastest, but shows the lowest accuracy;
- training the LSTM model takes more than 6 times longer than the convolutional model;
- the GRU model requires the longest time to train among all, but it turns out to be the leader in accuracy labels.

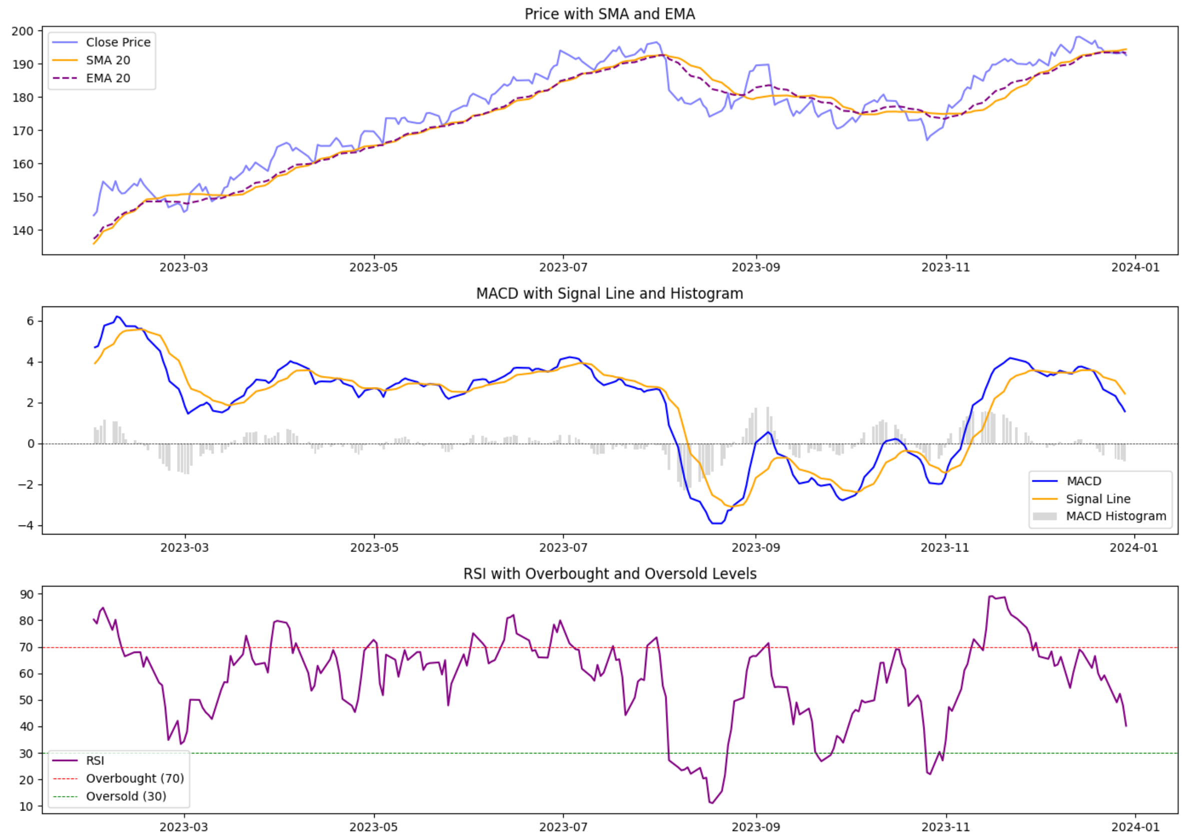

Figure 3: Composition of the output visualization of the results of data analysis based on the assessment of indicators.

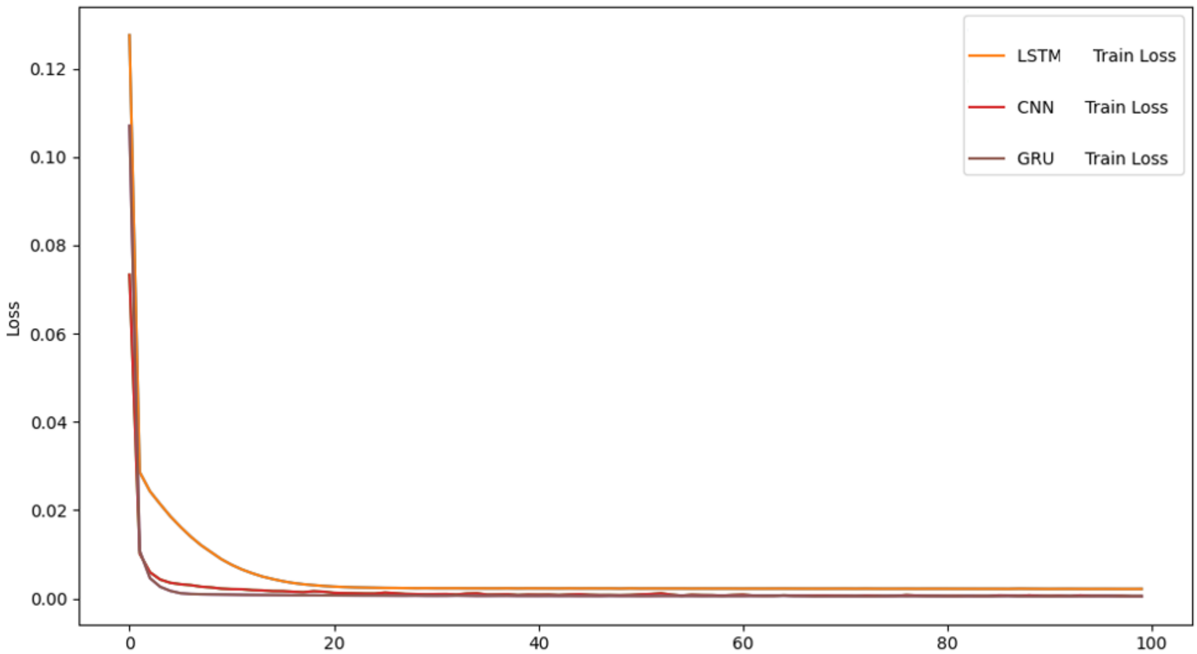

Figure 4: Comparison of loss metrics for the constructed DL models.

The GRU model demonstrated the highest values of the R^2 coefficient, which indicates its best predictive ability. All models used the same optimizer, Adam, which ensures uniformity in the training process and allows for a more objective comparison of their results. From the analysis, we can conclude that the GRU model is the most accurate in terms of accuracy marks and R^2 coefficient, despite the longer training time, while the LSTM model requires more time to stabilize and is less efficient in terms of training speed.

The GRU model (figure 5) demonstrates the ability to accurately account for local trend changes and effectively describe periods of market turbulence. When forecasting for the next 30 days (monthly time horizon), the GRU model predicts an optimistic increase in asset prices, which suggests a long-term purchase strategy. At the same time, the CNN model gives a pessimistic forecast, focusing on selling assets, and LSTM offers a neutral approach, closer to the asset holding strategy.

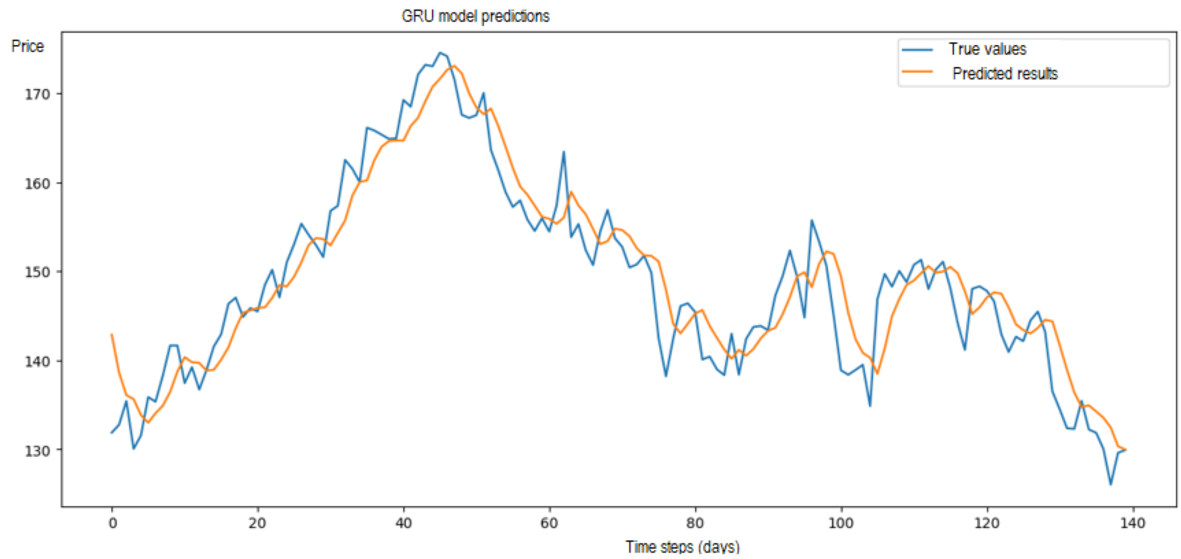

Figure 5: GRU model predictions.

Thus, if we compare the forecast results obtained on the basis of SMA, EMA, RSI, MACD, ARIMA, GRU, LSTM, CNN with each other in terms of accuracy and completeness, we should note higher values for deep learning models (especially GRU) and the seasonal ARIMA model, allowing us to make more balanced and confident decisions on selling or buying assets. Using only indicator values does not show high reliability of forecasts and is almost 1.5 times less accurate than LSTM models.

3. Summary

The conducted studies confirm that the developed system for analyzing data and supporting trading decisions on the exchange provides the user with the opportunity for a comprehensive assessment of various strategies. It is based on a combination of forecasts generated by trading indicators, ARIMA, GRU, LSTM and CNN models, which is especially important in the conditions of high uncertainty typical of exchange markets. These uncertainties are associated with the influence of many factors – social, political, environmental and economic – that are difficult to formalize.

The key novelty of the work lies in the adaptation, aggregation and hybrid software implementation of different approaches to the formation of recommendations for making trading decisions in a single system built on a modular architecture, as well as in the development and optimization of different deep learning models with an assessment of their effectiveness.

We can note that decisions should be based on the interpretation of forecasts from all models. To improve accuracy, it is important to consider both the coincidences in their results and the differences: if all models give consistent forecasts, this increases confidence in the choice of strategy and if there are any disagreements in prediction conservative strategy is more preferable; linear conducting additional analysis of external factors. Also, we want to accentuate that the final trading decision should be made jointly with the data analyst.

The integration of hybrid neuro-fuzzy methods into the system will increase the stability of forecasts, especially in conditions of uncertainty, and minimize the influence of external factors on decision-making.

Thus, using the system not only helps to improve the quality of forecasts, but also serves as a tool for building more sustainable trading strategies in turbulent conditions.

Declaration on Generative AI: The authors have not employed any Generative AI tools.

References

[1] N. Rudnichenko, V. Vychuzhanin, N. Shibaeva, I. Petrov, T. Otradskya, Intelligent Data Clustering System for Searching Hidden Regularities in Financial Transactions, in: A. Pakstas, Y. P. Kondratenko, V. Vychuzhanin, H. Yin, N. Rudnichenko (Eds.), Proceedings of the 11-th International Conference "Information Control Systems & Technologies", Odessa, Ukraine, September 21-23, 2023, volume 3513 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 163–176. URL: https://ceur-ws.org/Vol-3513/paper14.pdf.

[2] N. Rudnichenko, V. Vychuzhanin, T. Otradskya, D. Shvedov, Intelligent System for Processing and Forecasting Financial Assets and Risks, in: S. Subbotin (Ed.), Proceedings of The Seventh International Workshop on Computer Modeling and Intelligent Systems (CMIS-2024), Zaporizhzhia, Ukraine, May 3, 2024, volume 3702 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 251–262. URL: https://ceur-ws.org/Vol-3702/paper21.pdf.

[3] M. Bao, Deep Reinforcement Learning Based Optimization and Risk Control of Trading Strategies, Journal of Electrical Systems 20 (2024) 241–252. doi:10.52783/jes.1943.

[4] Z. Wang, Y. Zhao, Research on Quantitative Trading Based on Deep Learning, in: J. C. Hung, N. Yen, J.-W. Chang (Eds.), Frontier Computing on Industrial Applications Volume 2, volume 1132 of *Lecture Notes in Electrical Engineering*, Springer Nature Singapore, Singapore, 2024, pp. 23–30. doi:10.1007/978-981-99-9538-7_4.

[5] X. Tay, S. Lim, Deep Reinforcement Learning in Cryptocurrency Trading: A Profitable Approach, Journal of Telecommunications and the Digital Economy 12 (2024) 126–147. doi:10.18080/jtde.v12n3.985.

[6] M. Vishal, V. Ravi, R. Lal, Ensemble deep reinforcement learning for financial trading, in: L. A. Maglaras, S. Das, N. Tripathy, S. Patnaik (Eds.), Machine Learning Approaches in Financial Analytics, volume 254 of *Intelligent Systems Reference Library*, Springer Nature Switzerland, Cham, 2024, pp. 191–207. doi:10.1007/978-3-031-61037-0_9.

[7] Y. He, B. Xu, X. Su, High-Frequency Quantitative Trading of Digital Currencies Based on Fusion of Deep Reinforcement Learning Models with Evolutionary Strategies, Journal of Computing and Information Technology 32 (2024) 33–45. doi:10.20532/cit.2024.1005825.

[8] P. Priyadarshi, P. Kumar, Detecting Insider Trading in the Indian Stock Market: An Optimized Deep Learning Approach, Computational Economics (2024). doi:10.1007/s10614-024-10697-z.

[9] G. Cao, Y. Zhang, Q. Lou, G. Wang, Optimization of High-Frequency Trading Strategies Using Deep Reinforcement Learning, Journal of Artificial Intelligence General Science (JAIGS) 6 (2024) 230–257. doi:10.60087/jaigs.v6i1.247.

[10] J. Aken, D. Liang, Z. Lin, C. Wang, The Application of Deep Reinforcement Learning in Stock Trading Models, in: Proceedings of the 7th International Conference on Economic Management and Green Development, volume 39 of *Advances in Economics, Management and Political Sciences*, 2023, pp. 215–223. doi:10.54254/2754-1169/39/20231973.

[11] Y. Wang, G. Yan, Survey on the Application of Deep Learning in Algorithmic Trading, Data Science in Finance and Economics 1 (2021) 345–361. doi:10.3934/DSFE.2021019.

[12] Q. Zhou, T. Qu, Y. Han, F. Duan, Deep Reinforcement Learning with Comprehensive Reward for Stock Trading, in: M. Tanveer, S. Agarwal, S. Ozawa, A. Ekbal, A. Jatowt (Eds.), Neural Information Processing, volume 1794 of *Communications in Computer and Information Science*,

Springer Nature Singapore, Singapore, 2023, pp. 530–541. doi:10.1007/978-981-99-1648-1_44.

[13] J. Sim, B. Kirk, Generalized Deep Reinforcement Learning for Trading, Journal of Student Research 12 (2023). doi:10.47611/jsrhs.v12i1.4316.

[14] M. Tran, D. Pham-Hi, M. Bui, Optimizing Automated Trading Systems with Deep Reinforcement Learning, Algorithms 16 (2023) 23. doi:10.3390/a16010023.

[15] M. Sahib, H. Elkina, T. Zaki, From Technical Indicators to Trading Decisions: A Deep Learning Model Combining CNN and LSTM, International Journal of Advanced Computer Science and Applications 15 (2024) 847. doi:10.14569/IJACSA.2024.0150685.

[16] M. Mondol, A Deep Reinforcement Learning Approach with Explainability for Cryptocurrency Trading, Master's thesis, University of Zurich, 2024. doi:10.13140/RG.2.2.31230.19520.

[17] Yahoo, Carriage Services, Inc. (CSV) Stock Historical Prices & Data - Yahoo Finance, 2025. URL: https://finance.yahoo.com/quote/CSV/history/.