Applied Intelligence (2024) 54:11690–11708
https://doi.org/10.1007/s10489-024-05770-x

Multi-agent platform to support trading decisions in the FOREX market

Marcin Hernes[1] · Jerzy Korczak[2] · Dariusz Krol[3] · Maciej Pondel[1] · Jörg Becker[4]

Accepted: 11 August 2024 / Published online: 30 August 2024

Abstract

Trading decisions often encounter risk and uncertainty complexities, significantly influencing their overall performance. Recognizing the intricacies of this challenge, computational models within information systems have become essential to support and augment trading decisions. The paper introduces the concepts of trading software agents, investment strategies, and evaluation functions that automate the selection of the most suitable strategy in near real-time, offering the potential to enhance trading effectiveness. This approach holds the promise of significantly increasing the effectiveness of investments. The research also seeks to discern how changing market conditions influence the performance of these strategies, emphasizing that no single agent or strategy universally outperforms the rest. In summary, the overarching objective of this research is to contribute to the realm of financial decision-making by introducing a pragmatic platform and strategies tailored for traders, investors, and market participants in the FOREX market. Ultimately, this endeavor aims to empower people with more informed and productive trading decisions. The contributions of this work extend beyond the theoretical realm, demonstrating a commitment to address the practical challenges faced by traders and investors in real-time decision-making within the financial markets. This multidimensional approach to financial decision support promises to enhance investment effectiveness and contribute to the broader field of algorithmic trading.

Keywords Multi-agent system · Supporting trading decision-making · FOREX market

1 Introduction

A combination of statistical analysis, financial mathematics, econometrics, and, increasingly, artificial intelligence often informs trading decisions. These methods are frequently integrated into multi-agent systems to enhance trading activities in the foreign exchange market (FOREX) [1]. These systems strongly emphasize high-frequency trading (HFT), short-term position openings/closings, and sophisticated algorithms that leverage robust indicators and modern technology. The goal is to generate profits by capitalizing on minimal price fluctuations, characterized by high-frequency occurrences, where profits often arise from market liquidity imbalances or short-term pricing inefficiencies.

In general, the trading platforms must offer real-time guidance on trading positions, such as when to open/close positions, whether to go long or short or when to step away from investments. These guidelines form specific trading strategies, defined by their verifiability, quantifiability, consistency, and objectivity [2].

A trading strategy should outline the assets, entry / exit points, and money management rules, drawing from fundamental, technical, or behavioral analysis. These strategies are validated through backtesting (historical data) or forward testing (simulated trading environments compared to real-world results). Online trading adds further challenges [3], requiring the real-time use of one or multiple algorithms, often implemented as software agents. Currently, most trading systems are based on single or numerous algorithms without employing agents [4–10]. They are also based on the single agent architecture [11]. This paper introduces A-Trader, a multi-agent platform designed to support financial

✉ Marcin Hernes
marcin.hernes@ue.wroc.pl

[1] Department of Process Management, Wrocław University of Economics, Komandorska 118/120, Wrocław 53-345, Poland

[2] International University of Logistics and Transport in Wroclaw, Sołtysowicka 19B, Wrocław 51-168, Poland

[3] Department of Applied Informatics, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, Wroc,ław 50-370, Poland

[4] Department of Information Systems, University of Münster, Leonardo Campus 3, Münster, 9, Germany

decision-making within the FOREX market. As the A-Trader platform is presented, several critical issues in designing advisory systems for stock markets will be addressed. These challenges encompass:

1. Integration of Diverse Decision Sources: Harmonizing many decision sources, offering insight into effectively integrating varied inputs for informed decision-making.
2. Selection of Recommendation Methods and Algorithms: Exploring the crucial task of selecting the most suitable recommendation methods and algorithms.
3. Cooperation and Control of Advisory Algorithms: The importance of seamless collaboration and control of advisory algorithms is emphasized, providing valuable insights into optimizing algorithmic efficiency.
4. Composition of the Global Investment Strategy Evaluation Criterion: We underscore a holistic approach to performance assessment by examining the need for a comprehensive evaluation criterion for global investment strategies.
5. System Openness and Interoperability: The authors discuss the importance of system flexibility and adaptability, offering an in-depth understanding of the prerequisites for system openness.

The solutions implemented in the A-Trader platform will exemplify the issues mentioned above. A-Trader is a dynamic multi-agent experimental platform for constructing, simulating, and assessing investment strategies, catering to various investor types. Technically, A-Trader is integrated with an online data system, MetaTrader, which provides raw and preprocessed data and buy-sell decisions generated by agents using various methods. The platform develops investment strategies and continuously evaluates them based on the open/close and short/long positions determined by the most highly rated agents. The significant advantage of A-Trader over other trading platforms lies in its use of a multicriteria function to evaluate the strategy unlike platforms that rely solely on return-based metrics, A-Trader calculates a return rate based on risk-based measures, including factors like the number of transactions, gross profit, gross loss, profitable trades, consecutive profitable transactions, non-profitable successive transactions, Sharpe ratio, average volatility coefficient, and average return per transaction [12, 13].

In this paper, it will be demonstrated that:

1. The use of advanced technologies and a system architecture offers better performance and greater openness than existing solutions.
2. Provides a flexible and agile methodology for the development of investment strategies.
3. It ensures more realistic trading performance analysis based not only on return-based metrics but also on risk assessment and endogenous benchmarks.
4. The approach allows for the creation of strategies with superior performance compared to other methods.
5. The multi-agent approach enables the simulation of trader behavior, which can be used to enhance FOREX decision-making processes.

The paper is structured as follows. The first part of the paper introduces A-Trader's architecture and functionalities. The second part delves into the specifications of various trading agents. Three categories of trading agents are examined: agents based on technical analysis, agents based on macroeconomic and fundamental analysis, and behavior-based agents. Subsequently, it outlines trading strategy-building approaches using the set of available agents, and concludes with an analysis of the results from research experiments evaluating the performance of selected trading strategies on FOREX.

2 Background

The design and implementation of multi-agent systems in stock trading has been a focal point for numerous projects and research reports.

2.1 Multi-agent systems for financial decision support

This paper [14] proposes a modular multi-agent reinforcement learning-based system for financial portfolio management (MSPM) to address the challenges of scalability and reusability in adapting to ever-changing markets. Using evolving agent modules (EAMs) for generating information and Strategic Agent Modules (SAMs) for portfolio optimization, the system ensures improved adaptability and performance, evidenced by significant outperformance in US stock market data. The multi-agent deep reinforcement learning framework proposed in [15] leverages the collective intelligence of expert traders, each focused on different timeframes, to improve trading outcomes. It employs a hierarchical structure in which knowledge flows from agents trading on higher time frames to those on lower time frames, improving robustness against noise in financial data. Other examples of multi-agent architectures based on the deep reinforcement learning framework are shown in papers [16] and [17].

A proposal for a framework for evolutionary multi-agent trading for FOREX was introduced in [18]. In this paper, the authors focused on currency trading and included the impact of FX trading spread. They used technical indicators to provide temporary features from which a decision tree defined the training strategy. Tree representation classifiers were built with Genetic Programming (an evolutionary technique). The authors proposed a general FOREX Genetic Programming Framework (FXGP), and the proposed simplified framework (sFXGP) has been deployed to construct multiple agents operating concurrently [19]. The works [20, 21] present an approach for financial market prediction, where agents examine the similarities between the ask and bid asset histories to predict quotes in real time. The paper [22] shows development of ForexMA, a multi-agent system that enhances decision-making in Forex trading by integrating both qualitative and quantitative information. The architecture includes three agents, namely, the Facts Analyzing Agent, the Decision Agent, and the Performance Analyzing Agent. The authors demonstrated that ForexMA outperforms human expert traders by delivering high-frequency, rapid solutions in a matter of seconds. This system was tested and proven to generate more accurate predictions than those made by human experts, who typically operate on lower frequency timeframes and require several hours to analyze the information.

2.2 Advanced methods for financial decision supporting

This section analyses the methods developed not as agent-based approaches but can be transformed into agent structures in multi-agent systems.

The works [23, 24] present the use of neuro-fuzzy computing and neural networks for making quotation predictions based on analysis of a financial time series's geometrical patterns. Another paper proposing a behavioral approach for trading decisions is [25]. Some authors present strategies based on trading bots [26] or deep belief networks (DBN) [27] to build investment decisions based on the S&P500. Deep learning techniques, in turn, are presented in [28]. The deep learning approach is based on such methods, as recurrent neural networks, including Long Short-Term Memory [60], spiking neural networks [29–31]. Machine learning (ML) techniques significantly impact on the automatic identification of trading agents to identify profitable strategies to trade in the stock or currency market. Financial predictions incorporating ML approaches construct training, test, and off-sample data sets as a collection of instances using commonly used technical indicators. An example of ML models applied to trading scenarios in the FOREX market was discussed in [32]. The authors wanted to verify whether, using these models, it is possible to obtain consistently profitable returns. The authors proved that while getting good returns using simple classifiers is possible, each model needed a specific setup, including variables such as the retraining period, the size of the retraining set, and the number and type of attributes selected to construct the model. The complexities of the market require a combination of parameters that, for the same instrument, could change under different market conditions and seasons. The models needed to learn new patterns to cope with the dynamics of the market, but at the same time, to avoid noisy ways that might not be related to the current market situation could only be based on training comprised of current values of the time series using a sliding window approach.

An ensemble approach was proposed in [33] where the authors classify the FOREX market, and behavioral analyzes are considered by a certain amount, 2) downtrends when FX rates decrease by a certain amount, and 3) sideways trends. They extract features from these trends using multi-scale features. Multiple classifiers are trained using these features. Bayesian voting was used to create an ensemble of these classifiers, which can recognize trends in the market. The experimental results showed that the proposed system could accurately identify up and down trends in the FX rate signal.

Mayo states that a significant amount of intraday market data is noise or redundant, and if it is eliminated, then predictive models built using the remaining intraday data will be more accurate. He proposed an algorithm known as Evolutionary Data Selection (EDS), which uses a model building algorithm in conjunction with the available training data to find an optimal subset of those data [34].

Until now, articles have discussed the competition between multi-agent trading systems and their performance in trading scenarios [50]. Some of them explore advances in artificial economics, including agent-based models, and their applications in finance and game theory [51]. Focusing on the evolution of multi-agent foreign exchange (FX) traders, Longinov analyzes their performance in FX markets [18]. Currently, there are many platforms for HFT decision support in FOREX, such as FinEXo, Trade360, AvaTRADE, EXsignals, and Trade Chimp.

2.3 Assessment of existing approaches

The presented theoretical and practical approaches and solutions are often insufficient for HFT decision support. They are characterized by low performance (insufficient to support HFT) and costly maintenance. Moreover, the problem with openness and integration of the technologies appears in most cases.

The existing platforms are mainly based on technical analysis. Fundamental analysis and behavioral analysis are considered to a low degree. The disadvantage of existing approaches is also a performance measurement process. Mainly return-based measures are only taken into consideration, and it causes to limitation of personalization of the strategies evaluation byways(for example, a specific group of users can take into consideration mainly the rate of return-based measures, and other groups of users want to take into consideration combining-based measures). Therefore, investors often also need other classes of measures, for example, risk-based measures, to properly manage risk. The existing platforms are also not fully open-accessible, and users can develop their strategies using the tools proposed by the given platform. It is very difficult to integrate strategies developed by a user in other software environments with the given trading platform. Therefore, the main research problem undertaken in this paper is to develop an approach that overcomes the presented disadvantages of existing approaches. For this purpose, we developed the conception and prototype of a multi-agent platform in our research.

3 Architecture and functionalities of A-trader

The primary strengths inherent in multi-agent systems, including A-Trader, lie in their openness to integrate novel trading algorithms and specific functionalities that enable model-building communication among various agents. These systems operate on the principles of collective intelligence, allowing for tailored solutions using diverse market monitoring methods. Multi-agent technology facilitates the customization of solutions through agents that evaluate existing methods and preprocess learning datasets. These agents have learning capabilities, evolving their knowledge about financial market behavior. Overcoming computational challenges are achieved by leveraging a service-oriented architecture and cloud computing (SOAP). The SOAP communication protocol, as implemented in A-Trader, greatly simplifies the integration of individual solutions due to its open and easily implementable nature[1]. Incorporating PUSH technology, a common feature in distributed systems, notably accelerates information propagation within the system, as discussed in further detail in [35]. The system retrieves real-time data from the currency market using MetaTrader or JForex software. A-Trader analyzes quotation data using many criteria, ensuring near real-time processing and the capability to handle diverse data sources. For a more in-depth understanding of A-Trader's architecture, system elements, and agent details, refer to [13, 36, 37].

In general terms, A-Trader is composed of agents capable of generating independent decisions. These decisions can be characterized by model building by consistency or contradiction, e.g., the two independent agents can simultaneously generate open and closed positions. Figure 1 presents an overview of the architecture and functional concept of A-Trader.

[1] *W3C SOAP*, https://www.w3.org/TR/soap/

The main goal of the Supervisor Agent (SA) is to generate profitable trading advice to achieve a specific rate of return and reduce investment risk. This agent performs based on Basic and Intelligent agents' decisions. It provides different trading strategies and final open/close long/short positions to the trader or automatically to the market. The Supervisor also resolves Computing Agent knowledge conflicts within the Cloud and evaluates their performance. Based on collected knowledge, this agent determines which decisions are considered in a given strategy and which are ignored.

The Notification Agent (NA) receives the data (quotations), distributes messages (signals) to various agents, and controls the system operation running in a multi-threaded manner. Information about the message flow (which agent sends signals to which agent) is read during the NA initialization from the Routing Table.

Figure 2 shows an example of the data flow inside the NA. This agent "listens" at the given port, and if information from Agent A5 is received, then NA searches, in the Routing Table, the agents who listen to messages (signals) from Agent A5. In the considered example, these are Agents A7 and A9. Next, the NA agent searches for threads being sent (Sending Threads Table) to Agents A7 and A9 and sends them through.

The Cloud of Computing Agents (CCA) consists of the Basic Agents Cloud (BAC) and Intelligent Agents Cloud (IAC). BAC consists of agents that preprocess the data and calculate the fundamental technical analysis indicators. IAC consists of agents with a knowledge base. They can perform the learning process and can change their internal state and parameters. This group of agents uses methods based on artificial intelligence (neural networks, rule-based systems, genetic algorithms, cognitive technologies, etc.), agents observing market behavior and agents analyzing text messages. User-defined Intelligent Agents Cloud (UAC) consists of agents created by external users. Integration of User-defined Agents within the system without installing the agent on the servers is possible in A-Trader. The result of the Basic Agents and the Intelligent Agents activity is a decision that the NA transfers to the Supervisor Agent.

The Market Communication Agents (MCA) communicate between A-Trader and the external environment. MCA provides the actual values of quotations, and they are responsible for performing open/close long/short position orders.

A visualization agent (VA) visualizes quotations, decisions, and long/short positions in the form of charts.

The layer of Cloud Computing Agents is the system's core that analyzes signals contained in notifications and delivers decision recommendations to the Supervisor Agent. The Supervisor Agent then generates the final decision, as previously stated. Selected agents (especially belonging to CCA) running on A-Trader architecture are described in the next section.

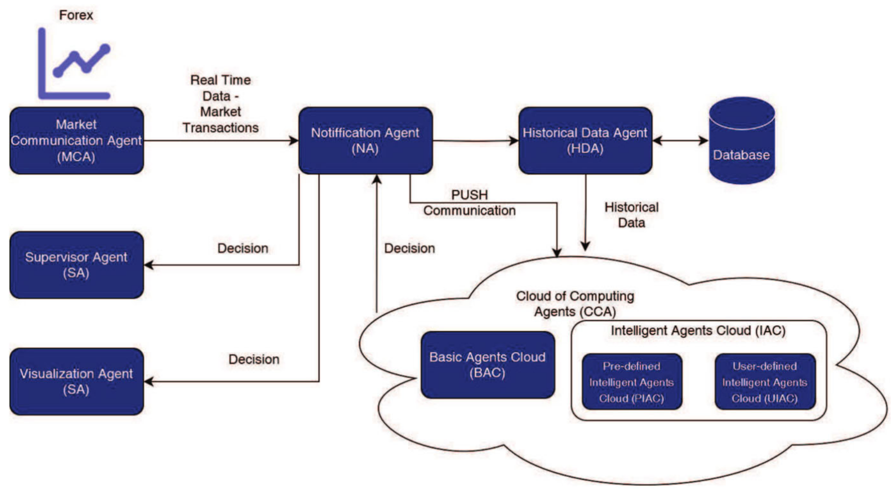


Fig. 1 A-Trader system architecture

Analyzing the computational complexity of a-Trader, it should be noted that it depends on the computational complexity of the algorithms of the individual agents. However, the architecture of the system makes it possible to determine decisions within 5 to 20 milliseconds of receiving the last quotation as an input signal (server parameters: Intel Core i7-9700K 8 cores, RAM 16 GB, NVIDIA GeForce RTX 2060 16GB, SSD M.2 480 GB, HDD SATA 7200 2000 GB).

4 Agent descriptions

A software agent is an intelligent program that not only executes based on acquired data but also takes specific actions to achieve a specified goal (for example, making satisfactory decisions in the FOREX market). A-Trader contains various types of agent, as mentioned in the previous chapter: Market Communication Agents, Notification Agents, Visualization Agents, Supervisor Agents, Historical Agents, and agents belonging to the cloud of computing agents (Basic Agents, Intelligent Agents, and User Agents), and currently, approximately 1600 agents are implemented on the platform. In cloud of computing agents These there are 800 basic agents (BAC) processing data agents (these agents calculate mainly technical analysis indicators related to FOREX market quotations), 500 intelligent agents (IAC), running in different aggregates and generating buy-sell decisions (about 250 agents based on three-valued logic, 250 agents based on fuzzy logic) and 300 agents generating open/close long/short positions (thus providing the strategies).

An experimental platform was designed to easily integrate new agents (user agents) and allow the reuse of existing agent decisions in new strategies. We adopted three conventions for generating agent responses / signals: three-value logic, fuzzy logic, and our signals. Three-value logic is a manner for the representation of agents' knowledge to provide buy / sell decision signals, generated as the agent's output signal, where the value 1 denotes a *buy* decision, the value -1 denotes a *sell* decision, and the value 0 denotes *don't care*. For a trading decision, fuzzy logic agents are more appropriate. The confidence range for decisions on A-Trader is $[-1 \ldots 1]$, where '-1' denotes a *strong sell* decision, '0' denotes a *strong leave unchanged* decision and '1' denotes a *strong buy* decision. The signal for open/close positions can then be generated based on a given decision's confidence level. For example, a short position is opened when a confidence level is greater than -0.8, whereas a long position is opened when a confidence level is more significant than 0.6.

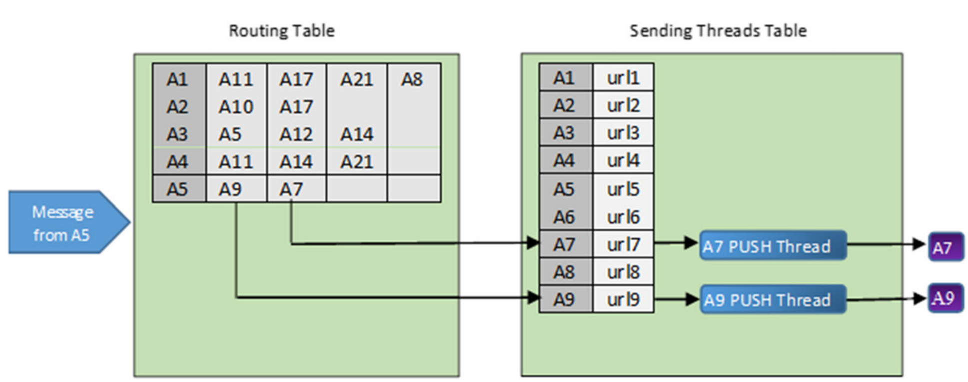


Fig. 2 Data flow inside the Notification Agent

As a result, open/closed positions can achieve more profitable results than positions generated based on three-value logic. It should be stressed that the level of confidence for open/close positions is very important, and it can be determined by considering trader experience or automatically determined by the Supervisor using, for example, a genetic algorithm. Specific signals are generated by agents which do not have simple/linear interpretation, for example, signals from agents with unsupervised learning.

Currently, A-Trader consists of three groups of buy/sell decision agents.

4.1 Agents based on technical analysis

Agents based on technical analysis use three-valued logic or fuzzy logic. Technical indicators have interpretations such as the market is oversold, the power of buyers is exhausted, etc. where assembling some of these may give satisfactory results. The shorter the investment horizon, the greater the effectiveness of technical analysis. To illustrate how an agent works, let us present an example of a fuzzy logic agent called *FuzzyTrendLinearRegression*. This agent makes decisions in the following manner. A given number of M quotations is approximated by the equation: $y = ax+b$ (straight line). The inclination of this line depends on the value of the coefficient *"a"* or the tangent value of the inclination angle using linear regression.

The agent generates a buy signal when the coefficient value of *"a"* changes from positive to negative, and it generates a sell signal when the coefficient *"a"* changes from negative to positive. The transition of the agent's decision is performed using the hysteresis level, defined by the coefficient value δ.

4.2 Agents based on macro-economic and fundamental analysis

A-Trader also consists of agents based on fundamental analysis and behavioral data. The fundamental analysis in FOREX is related to the economic, social, and political forces driving demand and supply on the currency market. The level of the supply and demand balance is affected by two main factors:

- Interest rates can strengthen or weaken a particular currency where a high level of interest rates (as compared to those in other currencies) can increase the level of foreign investment in a currency, which in turn, leads to a strengthening of the currency.
- The international trade balance deficit (higher value of imports than the value of exports) can usually adversely affect a currency. In this case, the currency is transferred out of a country to buy foreign products, which can lead to a devaluation of the currency.

Algorithm 1 The *FuzzyTrendLinearRegression* agent specification.

Input: $< w_1, w_2, \ldots, w_M >, prev\_a$
/* The vector of the quotation values of M is determined by the trader or by the genetic algorithm, the previous value of the coefficient "a" */
Output: Decision D (fuzzy logic - value range [-1…1]
$sum\_y \leftarrow 0; sum\_x \leftarrow 0; sum\_xy \leftarrow 0; sum\_x2 \leftarrow 0;$
/ * sum_y - the sum of the M quotations values, sum_x - the sum of the quotation indices in the input vector, sum_xy - the sum of the products of the quotations values and quotation indices, sum_x2 - the sum of the squares of quotation indices in the input vector */
$D \leftarrow 0; TRL\_count \leftarrow 0;$
/ * counter needed for fuzzification * /
$Max\_count \leftarrow 0;$
/ * maximum limit of counter needed for fuzzification * /
for $k \leftarrow 1$ to M **do**
  $sum\_y \leftarrow sum\_y + w_k; sum\_xy \leftarrow sum\_xy + w_k * k;$
  $sum\_x \leftarrow sum\_x + k; sum\_x2 \leftarrow sum\_x2 + k * k;$
  $k \leftarrow k + 1; c \leftarrow sum\_x2 * M - sum\_x * sum\_x;$
end for
if $c = 0$ **then**
  $c \leftarrow 0.1; a \leftarrow (sum\_xy * M - sum\_x * sum\_y)/c;$
end if
if $(a = prev_a = 0) \vee (a < 0 \wedge prev_a < 0) \vee (a > 0 \wedge prev_a > 0$ **then**
  $D \leftarrow 0;$
end if
if $a > 0 \wedge prev_a < 0$ **then**
  if $TRL\_count > 0$ **then** $TRL\_count \leftarrow 0;$
  end if
  $TRL\_count \leftarrow TRL\_count - 1;$
  if $TRL\_count < max\_count$ **then** $TRL\_count \leftarrow max\_count;$
  end if
end if
$D \leftarrow TRL\_count/max\_count;$
if $a < 0 \wedge prev_a > 0$ **then**
  if $TRL\_count < 0$ **then** $TRL\_count \leftarrow 0;$
  end if
  $TRL\_count \leftarrow TRL\_count + 1;$
  if $TRL\_count > max\_count$ **then** $TRL\_count \leftarrow max\_count;$
  end if
end if
$D \leftarrow TRL\_count/max\_count; prev_a \leftarrow a;$

Other factors, such as central bank interventions (e.g., by increasing / reducing foreign exchange reserves) strengthen / reduce demand for a specific currency. Fundamental analysis is based on an examination of asset markets, macroeconomic indicators, and political considerations of the country to evaluate the development of the exchange rate of a particular currency. Asset markets include stock exchanges, bond markets, and real estate. Macroeconomic indicators are measured by Gross Domestic Product, Money Supply (M1, M5, D1, W1, etc.), unemployment, inflation, foreign exchange reserves, interest rates, and productivity. Political considerations can influence the level of certainty of stability and the level of confidence in a nation's government. The fundamental analysis agents also consider indicators such as the Consumer Price Index (CPI), Durable Goods Orders, Producer Price Index (PPI), Purchasing Managers Index (PMI) and retail sales.

However, often online fundamental analysis only sometimes provides market entry and exit points in FOREX as a lot of information emerges at regular intervals. Still, only a part of this information is relevant. Therefore, there are only a few agents based on macro-economic and fundamental analysis are implemented, notably:

1. Interest rates - if interest rates are higher in one country than in its neighbors, the currency prices in this country will often strengthen because a higher interest rate attracts more foreign investors.
2. Gross Domestic Product (GDP) is the sum of all goods and services produced/provided by domestic or foreign companies in a given country. Based on GDP, the level of growth (or contraction) of a country's economy can be measured. This indicator has the broadest scope for the change in economic output and production in a given country. The Gross National Product (GNP), in turn, is related to the nationality of capital.
3. Purchasing Manager's Index (PMI) includes data related to new orders, supplier delivery times, production, backlogs, prices, inventories, employment, import and export orders. It is characterized by high correlation with Monetary Policy Decisions and is a valuable tool to track the health of a country's manufacturing sector.
4. Indexes:
   - S&P 500 is treated as a leading indicator of US equities and is meant to reflect the return/risk characteristics of the large cap universe, this index includes 500 stocks chosen on the basis of market size, liquidity industry grouping, and other factors.
   - FTSE 100 is a London Stock Exchange indicator and includes 100 companies characterized by the highest market capitalization on this Exchange.
   - WIG–is a Warsaw Stock Exchange index that includes securities listed on the main market.

To illustrate one of the agents based on macro-economic analysis, there is an agent called *FuzzyNeuralNetIndices*. The agent computes by applying Multilayer Perceptron to the trading decisions on the S&P500 and WIG indices.

This agent is based on the interpretation of the money flow. If WIG20 is rising and the S&P 500 is falling, it can be predicted that investors can exchange their S&P shares for USD, then they can exchange USD for PLN to finally buy WIG shares. Therefore, if they buy PLN for USD, the value of PLN about USD should grow. Other fundamental analysis agents consider information about:

Algorithm 2 The *FuzzyNeuralNetIndices* agent specification.

Input: $w =< w_1, w_2, \ldots, w_M >, v =< v_1, v_2, \ldots, v_M >$
1: /* The vector of S&P 500 index values consisting of M values (M is determined by the user or by genetic algorithm), the vector of WIG index values consisting of M values */
Output: The decision D (fuzzy logic - value range [-1…1]
2: $in_M \leftarrow< w_1, w_2, \ldots, w_M, v_1, v_2, \ldots, v_M >$;
3: $pred_{M+1} \leftarrow Multi-layer Perceptron(in_M)$;
4: $D \leftarrow heuristic\_open\_close(pred_{M+1})$;

- Gold prices ratio: when the price of gold goes down, then the USD often goes up (and vice versa); that means that prices of gold tend to have an inverse relationship to the price of USD and currency traders can take advantage of this relationship.
- Oil price ratio: economies of oil-dependent countries grow (investors buy their currencies as a consequence) as oil prices drop.

4.3 Behavior-based agents

Many experts point out that the currency market is strongly correlated with the expectations of traders and their assessment of these expectations. There is a commonly observed relationship between stock prices and the behavior of traders, notably their perception of risk and benefit. Various prognoses, bulletins, and blogs strongly influence these expectations. An understanding of investor psychology can generate profit opportunities and thus can be extremely valuable for designing trading strategies. Many studies of behavioral models are used in FOREX trading, most based on psychology theories and applying data mining methods [38]. However, to validate these models on real financial markets, detailed information about traders, their experience and knowledge, and their psychological biases is needed.

Considering the limited sources of information on these subjects, in A-Trader only a behavioral time series has been provided and a few behavioral agents have been implemented [20, 39]. The datasets are a broad range of day-by-day indicators (sentiments) provided by Polands MarketPsych Data or INI indicator. The indicators have been computed from millions of articles and posts in the news and on social media. In the experiments, behavioral indicators such as SENTIMENT, OPTIMISM, FEAR, as they relate to specific countries and their currencies (e.g. USD/PLN) are updated every day for countries and currencies and are input directly into A-Trader agents. For example, the SENTIMENT index

Algorithm 3 The specification of the agent working on SENTIMENT index values.

Input: $s\_PLN =< s\_pln_1, s\_pln_2, \ldots, s\_pln_M >$
$s\_USD =< s\_usd_1, s\_usd_2, \ldots, s\_usd_M >$
$w =< w_1, w_2, \ldots, w_M >, threshold\_open, threshold\_close$
/* SENTIMENT values for PLN, SENTIMENT values for USD, USD/PLN quotations, the level of threshold for open long/close short position, the level of threshold for close long/open short position */
Output: The decision D (fuzzy logic - value range [-1…1]
$s\_pln_{M+1} \leftarrow Multi-layer Perceptron(s\_PLN)$;
$s\_usd_{M+1} \leftarrow Multi-layer Perceptron(s\_USD)$;
if $(s\_pln_{M+1} > threshold\_open) \wedge (s\_usd_{M+1} > threshold\_open)$ **then**
  $D \leftarrow heuristic\_open(s\_pln_{M+1}, s\_usd_{M+1}$;
else if $(s\_pln_{M+1} > threshold\_close) \wedge (s\_usd_{M+1} > threshold\_close)$ **then**
  $D \leftarrow heuristic\_close(s\_pln_{M+1}, s\_usd_{M+1})$;
else
  $D \leftarrow heuristic\_do_nothing()$;
end if

indicates the 24-hour rolling average score of references in news and social networks to overall positive references, net of negative references. The OPTIMISM index is a bipolar emotional indicator in the range of -1 to 1. For interpretation purposes, gradual improvement of the SENTIMENT drives the continuation of the trend.

As mentioned above, agents can generate decisions that may be mutually consistent or completely contradictory. In A-Trader, the conflicts between agents are resolved by the Supervisor. This agent receives signals from decision-making agents and evaluates their performance. Through this evaluation, the Supervisor determines the agents for building investment strategies. In this way, the Supervisor can apply various strategies to generate open/close long/short position signals. The following section describes examples of these strategies.

5 Trading strategy construction

The strategies of A-Trader are built on the basis of the following assumptions:

1. Buy/sell decisions generated by a Cloud of Computing Agents form a base for strategy building. Every agent running in this Cloud sends its decision to the NA based on a unique decision method for each agent.
2. The Supervisor Agent builds investment strategies based on buy/sell decisions generated by Cloud Computing Agents (read from NA). These strategies generate the open/close long/short position signals.
3. Users–traders or bots (automatic traders), who invest in FOREX.

The strategies of A-Trader are based on more complex algorithms than algorithms based on technical analysis indicators [40] and, to illustrate the applied concept, four strategies are detailed: *MyStrategy*, *Consensus*, *Evolution-based*, *Deep learning*. These strategies have been chosen, because they were developed on the basis of the deep literature study and based on many experiments (these strategies are in the advanced phase of development in A-Trader, and the remaining strategies are in the preliminary phase of development).

5.1 MyStrategy

The strategy called MyStrategy is built of the basis of the following technical analysis, fundamental analysis, and behavior-based agents' signals:

- FuzzyRSI based on the Relative Strength Index indicator,
- FuzzyROC based on the Rate of Change indicator,
- FuzzyCCI based on the Commodity Channel Index indicator,
- FuzzyMACD based on the Moving Average Convergence Divergence indicator,
- FuzzyBollinger based on the Bollinger Bands indicator,
- FuzzyWilliams based on the Williams %R indicator,
- FuzzyNeuralNetIndices,
- BehavioralAgent.

This strategy is run so that the open / close short / long position signal is generated when the average of fuzzy agent signals is higher / lower than a predefined threshold.

The strategy can be defined as follows:

Algorithm 4 The specification of MyStrategy.

Input: $SignalsAM = \{AM^1, AM^2, \ldots, AM^8\}$,
$threshold_open, threshold_close$
/ * the threshold level for the open long / closed short position, for close long/open short position */
Output: position
/* value 1 – open long and closed short position, value -1 open short and closed long position, value 0 – out of market */
$S \leftarrow 0$;
for $i \leftarrow 1$ to 8 **do**
  $S \leftarrow S + AM^i$;
end for
$S \leftarrow S/8$;
$position \leftarrow 0$;
if $S \geq threshold\_open$ **then**
  $position \leftarrow 1$;
end if
if $S \leq threshold\_close$ **then**
  $position \leftarrow -1$;
end if

5.2 Consensus strategy

The strategy *Consensus*, built on developing a consensus that determines the issues for financial decisions, is described in detail in [41, 42]. The consensus agent, presented in detail in [36], develops a trading strategy based on a set of decisions generated by fuzzy logic agents.

The strategy can be specified as follows:

Algorithm 5 The specification of Consensus.

```
Input:  A = {D^1, D^2, ..., D^M},
threshold_open, threshold_close
/* The profile (set of decisions of M fuzzy logic agents, where M denotes the number of fuzzy logic agents, and D^1, D^2, ..., D^M denotes decisions of particular agents), the threshold level for the open long / closed short position, the level of threshold for close long/open short position */
Output:  position
/* value 1 –open long and closed short position, value -1 open short and closed long position, value 0 – out of market */
CON ← 0; /* consensus */
B ← Sort_Asc(A);
/* B = {B^1, B^2, ..., B^M} – ascending order of the values of profile A */
i ← Floor((M + 1)/2);
j ← Ceil((M + 2/2);
Set CON as any value from interval [B^i, B^j];
position ← 0;
if CON ≥ threshold_open then
    position ← 1;
end if
if CON ≤ threshold_close then
    position ← −1;
end if
```

5.3 Evolution-based strategy

The strategy *Evolution-based* is developed based on work [52]. This strategy determines the best thresholds for open/close long/short positions based on decisions generated by technical analysis agents, fundamental analysis agents, and behavior-based agents. The *Evolution-based* strategy determines which agents should be considered when generating long/closed open/short position signals. It also determines the importance of decisions generated by a specific agent. The evolutionary algorithm indicates the space of agent decisions and weights their importance. The genotype in Fig. 3 consists of the weightings and thresholds for the opening / closing of the short / long position for each agent separately.

In addition to weighting and thresholds, every advisory agent is characterized by 'compulsory' parameters. These parameters mean that the agent's signal value must be open, close, or 'don't care'. The genotype also consists of values such as Profit Taking, Trailing Stop and Stop Loss for long and short positions. The result of this algorithm is a phenotype - a set of decision rules. For example, the open short position rules for the agent at time T_0 can be specified as follows:

$$\begin{gathered} A_1 T_0 * w_{so_1} + \cdots + A_n T_0 * w_{so_n} Th_{so} \\ \wedge \\ (A_1 T_0 * C_{so_1} 0) \vee (C_{so_1} = 0) \\ \wedge \\ \vdots \\ \wedge \\ (A_n T_0 * C_{so_n} 0) \vee (C_{so_n} = 0) \end{gathered} \qquad (1)$$

where:

$A_n T_0$ – value of Agent n signal in time T_0,

$w_s o_n$ – weighting for Agent n short position opening,

$Th_s o$ – threshold for Agent n short position opening,

$C_s o_n$ – compulsory parameter for Agent n short position opening.

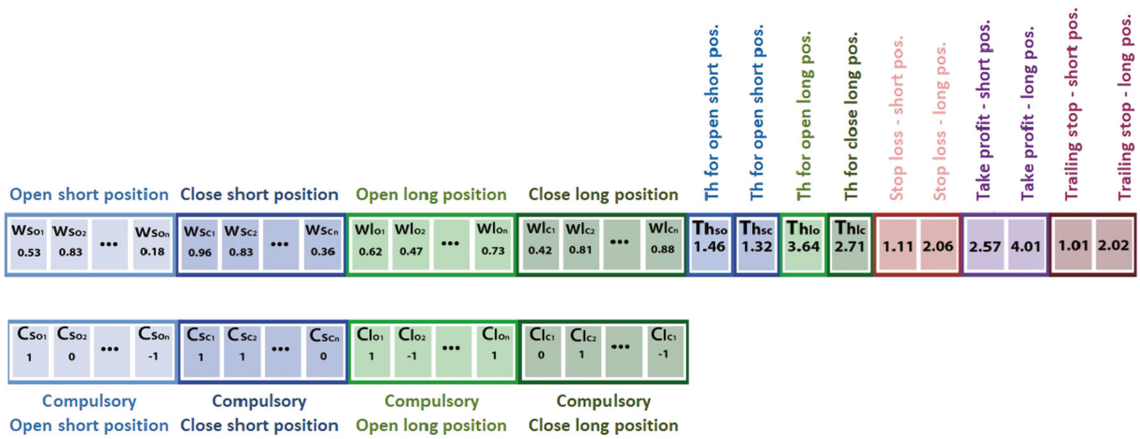
The conditions for the open/close short/long position are divided into two parts. The algorithm checks if a threshold is reached in the first part. The threshold is checked by multiplying the signals of each agent by the corresponding weightings, then all the results are to be summed up. The first part of the condition is met if the sum is higher than the opening short position threshold. The algorithm checks if all the mandatory rules are met in the second part of the condition. If a compulsory parameter of Agent 1 (OSO1) is equal to zero, the algorithm 'does not care' what the value of Agent 1 is. If the parameter is equal to 1, the condition will be fulfilled only when the signal value of Agent 1 is positive. Similarly, in the case where the compulsory parameter is equal to -1, the algorithm expects a negative value of Agent 1. The compulsory parameters are checked for every advising agent. The strategy can be specified as follows.

5.4 Deep learning strategy

The *Deep learning* strategy has been implemented on an open-source H_2O platform [24]. It is a distributed, scalable, and interactive in-memory data analysis and modeling solution. This platform consists of several data analysis models, including the Deep Learning Model, for Big Data exploration. In our approach, H_2O has been integrated with A-Trader.

The *DeepLearningH_2O Agent* is controlled by Supervisor and runs in two modes, cf. Figure 4:

1. Learning mode (continuous) divided into the following steps:
   - Import time series from A-Trader to H_2O platform (H_2O is external module of A-Trader, therefore data are imported indirectly from A-Trader database, Notification Agent signals are not used),

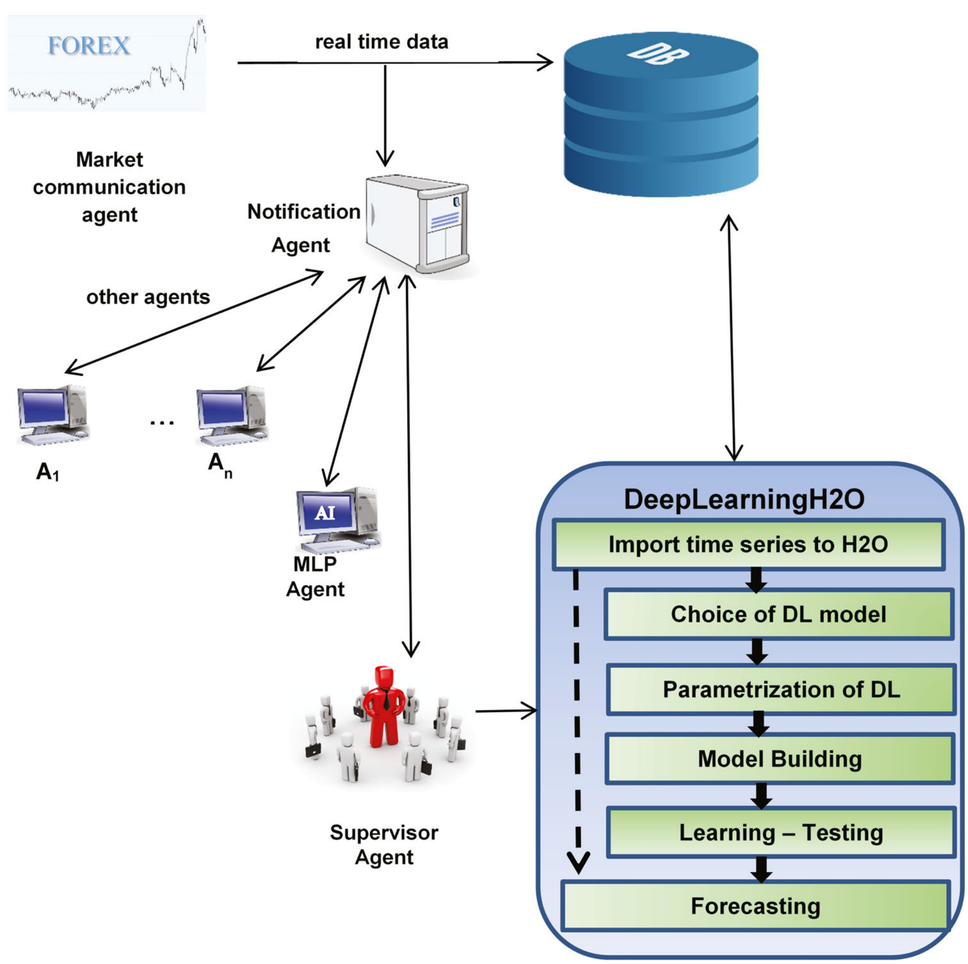


where:
Th. – threshold, o. – opening position, c. – closing position, pos. – position,
Wso_1 – the weighting for opening the short position by the first agent,
Csc_1 – the weighting for the compulsory closing of the long position by the first agent.

Fig. 3 Genotype used in the *Evolution-based* strategy

- Deep Learning (DL) model specification,
- DL model Parametrization – (parameters such as;- number of training epochs, number of hidden layers, stopping rounds, stopping metrics, etc),
- Building of DL model – on the basis of imported data structure and determined parameters,
- Learning and Testing – where the training and the validation datasets are used. Long Short Term Memory

Fig. 4 Schema of Deep Learning strategy



Algorithm 6 The specification of Evolution-based.

Input: $V = \{V^1, V^2, \ldots, V^M\}$,
/* The vector of M decisions of fuzzy logic agents, where M denotes the number of fuzzy logic agents running in the system, $V^1, V^2, \ldots, V^M$ denotes the decisions of particular agents */
Output: position
/* value 1 – open long and closed short position, value -1 open short and close long position, value 0 – out of market – close short/long position*/
if $CheckPerformanceLevel()$ **then**
  $BeginLearningProcess()$;
end if
$position \leftarrow 0$;
if $LongPositionOpened$ **then**
  if $LongPositionOpened$ **then**
    if $CheckClosingLongPositionCondition(V)$ **then**
      $position \leftarrow -1$;
    end if
  end if
  if $ShortPositionOpened$ **then**
    if $CheckClosingShortPositionCondition(V)$ **then**
      $CloseShortPosition()$;
      $position \leftarrow 1$;
    end if
  end if
  if $CheckOpeningLongPositionCondition(V)$ **then**
    $position \leftarrow 1$;
  end if
  if $CheckOpeningShortPositionCondition(V)$ **then**
    $position \leftarrow -1$;
  end if
end if

architecture of the deep neural network was used. The architecture and hyperparameters of the model are as follows: three hidden LSTM layers (16, 8 and 4 units), dropout layer (rate 0.3), RELU activation function for hidden layers and linear activation function for output layer, loss function: mean squared error, optimizer: adam, metrics: mse, mae, mape, msle, number of epochs:100, batch size:32.

2. Forecast mode (continuous) – time series of quotations are continuously imported from the A-Trader database, and the trained model is used for predicting rates of return.

The *DeepLearningH*$_2$*O Agent* is supported also by the following agents [43]:

- Basic Agents - perform time series pre-processing and compute the basic indicators; agents can learn and change their parameters and internal states based on their knowledge.
- Intelligent Agents – running on the basis of artificial intelligence (genetic algorithms, rule-based systems, neural networks including MLP, etc.), text messages analysis-based agents, market behavior-based agents.
- Decisions of Basic Agents and Intelligent Agents are sent to the Supervisor Agent.

Formally, the model used by *DeepLearningH*$_2$*O Agent* is defined as follows:

$$Y^i_{t+1} = DLM\left(x^i, x^p, \ldots, x^q\right) \tag{2}$$

where:
x^i is an input vector of the main quote rate of return, $x^p, \ldots, x^q$ are inputs vectors consist of the rates of return of the quotations correlated with main quotation (e.g., main quotation is EUR/USD and correlated quotations are gold quotations and oil quotations).

This model uses log-return rates, calculated as follows:

$$r^j(t) = \log\left(\frac{S^i_t}{S^i_{t-1}}\right) \tag{3}$$

where S^i_t denotes a price of quotation i at time t.

H_2O normalizes log-return rates and projects them in the range from -1 to 1. Input vector related to main quotation is defined as follows:

$$x^i = \left\{r^j(t), r^j(t-1), \ldots, r^j(t-k)\right\} \tag{4}$$

where k denotes the number of past quotations used as input.

Y_{t+1} values are in the range [-1, 1] (generated as fuzzy logic signals) and predict logarithmic return rates at time $t + 1$ (normalized value).

The training set consists of input vectors x^i and inputs $x^p, \ldots, x^q$ at time t, $t - 1$, etc., and output at time $t + 1$. The learning process is performed on the basis of historical time series; hence the log-return rate at time $t + 1$ is known.

The Supervisor Agent uses different strategies to generate opening/closing positions, on the basis of the output of *DeepLearningH*$_2$*O Agent* using for instance, consensus strategy or a genetic algorithm, whereas a genetic algorithm determines threshold levels for open/closed short/long positions. The Supervisor also determines the mode of *DeepLearningH*$_2$*O Agent* operation. If the performance of *DeepLearningH*$_2$*O Agent* is low (performance measuring issues are presented in the next section), then a learning mode is initiated. If performance is high, a forecasting mode is run using a previously generated model.

The strategies provided by A-Trader can be reused and extended. The user (trader) can add a new agent or source of information by filling out a generic pattern of the agent structure. This is a process of inserting selected agents into your trading strategy.

6 Experiments

The main aim of the experiments is to evaluate the performance of selected trading strategies. The specific aims are as follows:

- running the investment strategies, developed in A-Trader, using real data form FOREX market,
- the assessment of long/short positions results using return-based and risk-based measures,
- comparing the performance of strategies to Buy and Hold benchmark,
- confirming the results using statistical tests.

Back-testing is used to verify that the A-Trader strategies were based on the following.

1. GBP/PLN quotations were selected from randomly selected periods, namely
   - 16-04-2018, 0:00 am to 19-04-2018, 23:59 pm,
   - 23-04-2018, 0:00 am to 26-04-2018, 23:59 pm,
   - 14-05-2018, 0:00 am to 17-05-2018, 23:59 pm.
2. The strategies *MyStrategy*, *Consensus*, *Evolution-based* were used to generate trading signals (open long/close short position equals 1, close long/open short position equals -1). Figure 5 presents an example (with description) of generated signals (the green line denotes the "long position", the red one denotes the "short position").
3. The Buy and Hold (B&H) strategy was used as a benchmark (the B&H strategy relies on opening a position at the beginning of the investment period and closing it at the end of this period).
4. Performance analysis ratios (absolute ratios) were measured in 'pips' (a change in FOREX price of a 'point' is called a pip).
5. The cost of transactions is directly proportional to the number of transactions.
6. It was assumed that in each transaction the investor engages 100% of the capital held where the trader can individually determine the capital management strategy.
7. The following measures (ratios) were used in the performance analysis [44–48]:
   - Rate of return (ratio x_1),
   - Number of transactions,
   - Gross profit (ratio x_2),
   - Gross loss (ratio x_3),
   - Number of profitable transactions (ratio x_4),
   - Number of profitable consecutive transactions (ratio x_5),
   - Number of unprofitable consecutive transactions (ratio x_6),
   - Sharpe ratio (ratio x_7),
   - Average coefficient of variation (ratio x_8),
   - Average rate of return per transaction (ratio x_9), counted as the quotient of the rate of return and the number of transactions.

Fig. 5 Example of strategy visualization

Table 1 Performance analysis results

Ratio	MyStrategy			Consensus			Deep learning			B & H		
	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3	Period 1	Period 2	Period 3
Rate of return [Pips]	-1244	5017	7824	-1149	5257	7698	-968	5412	7736	-1487	4872	7360
The number of transactions	381	59	323	162	47	139	211	59	234	1	1	1
Gross profit [Pips]	103	186	197	146	217	245	182	254	238	0	4872	198
Gross loss [Pips]	126	77	145	157	149	183	143	161	176	-1487	0	0
The number of profitable transactions	204	38	198	74	32	86	127	172	95	0	1	1
The number of profitable consecutive transactions	10	6	11	5	6	12	12	17	10	0	1	1
The number of unprofitable consecutive transactions	7	4	6	4	2	3	4	2	5	1	0	0
Sharpe ratio	0.60	0.52	0.32	1.92	1.24	2.49	0.92	1.17	1.79	0	0	0
The average coefficient of variation	0,79	0.89	0.83	0.62	0.18	0.35	0.70	0.43	0.58	0	0	0
The average rate of return per transaction	4.25	5.18	3.50	-7.09	111.85	55.38	-4.59	91.72	33.05	-1487	4872	7360
Value of evaluation function (y)	0.23	0.32	0.34	0.41	0.36	0.43	0.47	0.44	0.38	0.08	0.26	0.21

8. For comparison of the agent performance, the evaluation function was elaborated, defined as follows:

$$y = (a_1x_1 + a_2x_2 + a_3(1 - x_3) + a_4x_4 + a_5x_5 + a_6(1 - x_6) + a_7x_7 + a_8(1 - x_8) + a_9x_9 \quad (5)$$

where x_i denotes the normalized values of ratios from x_1 to x_9 (mentioned in item 6). For this experiment, coefficients were set as follows: x_1 to $x_9 = 1/9$. However, it is possible to adopt other values for these coefficients. They can be modified using, for example, an evolution-based method, or they can be determined by the trader according to his/her preferences. The functions can be easily modified, and they aggregate many assessment indicators so that users can choose which assessment criteria are most important to them. For example, a trader may be interested in achieving a high rate of return with a high level of risk or a low risk with a low rate of return. Coefficients are needed because the user can arbitrarily classify individual components. The function y returns values from the range $[0 \ldots 1]$, and the agent's performance is assigned proportionally to the function value. This is just one of the evaluation functions, as A-Trader allows a user to build other functions.

Table 1 presents the results of the performance analysis. A wide number of changes in particular ratio values significantly hinder the analysis by the trader and. Consequently, making decisions in time close to real time is very difficult. The results of the experiment allow us to come to the conclusion that the strategy ranking differs in particular periods.

In the first and second periods, *Deep learning* was the best evaluated strategy. In the third period, the best was the *Evolution-based* strategy. *MyStrategy* was evaluated worse than *Deep learning* and *Consensus* and *B&H* was ranked the lowest in all periods.

Considering all periods, it can be stated that the highest rate of return characterized the *Deep learning* strategy, it was ranked highest in two of the three periods. There was a lower value of the evaluation function in the third period than in *Consensus* case, which may result from lower values of ratios such as the average rate of return per transaction and risk measures. The *Consensus* strategy achieved the lowest values for risk measures. It can also be concluded that the low evaluation of *MyStrategy* in all periods is due not only to the level of the rate of return but also to a high risk level and a large number of unprofitable consecutive transactions. The MyStrategy is simple strategy based on decisions generated by particular agents. The results achieved by MyStrategy allow us to draw conclusions that more sophisticated multi-agent-based methods, such as consensus or deep learning, can perform better than simple strategies. The comparison of multi-agent-based methods and stand-alone methods is presented in our earlier research, for example [13, 36, 37].

Table 2 Results of Friedman ANOVA test, POST-HOC (Dunn Bonferroni) for period 1

p-value	MyStrategy	Consensus	Deep learning
MyStrategy		≤ 0.000001	≤ 0.000001
Consensus	≤ 0.000001		0,005493
Deep learning	≤ 0.000001	0,005493	

The evaluation analysis in other trading systems (e.g., Trade Chimp, XTRADE, MetaTrader) is performed "manually" by the investor in most cases, and this is a very time-consuming process during which there is limited working of the system in real time. These systems offer basic performance measures: rate of return, highest profit, highest loss, number of transactions, total profit, number of profitable transactions, number of profitable consecutive transactions, number of unprofitable consecutive transactions. A-Trader calculates additional ratios, such as risk measurements (average coefficient of variation, Sharpe ratio), or the average rate of return for a specific transaction.

The A-Trader evaluation function enables the measurement and evaluation of investment strategies. These operations are performed automatically by the Supervisor Agent (in time close to real time), which may then advise the investor to trade on the basis of the decisions generated by the strategy characterized by the highest performance level. In addition, users can change the parameters a_i and x_i of this function to consider the preferences of the user related to particular performance measures. To confirm the results, statistical tests were performed separately for particular periods using the rate of returns generated by particular transactions in selecting a given strategy as input data. PQStat software [2] was used for this and the following hypothesis was assumed:

1. H_0 – the given strategy was not the best in the given period (the rates of return achieved are not statistically significant).
2. H_1 – the given strategy is the best in the given period (the rates of return achieved are statistically significant).

First, normality tests were performed. Data are characterized by a nonnormal distribution at the 5% significance level; therefore, a Friedman ANOVA test was performed that included POST-HOC (Dunn Bonferroni). The results are presented in Tables 2, 3, and 4.

The calculated p-values between the returns rates generated by particular strategies are less than 0.05 in all periods. The lower probability of the p-value indicates stronger evidence against the null hypothesis. Therefore, the null hypothesis can be rejected and the return rates generated by

[2] *PQStat software*, https://pqstat.pl/

Table 3 Results of Friedman ANOVA test, POST-HOC (Dunn Bonferroni) for period 2

p-value	MyStrategy	Consensus	Deep learning
MyStrategy		$\leq$ 0.000001	$\leq$ 0.000001
Consensus	$\leq$ 0.000001		$\leq$ 0.000001
Deep learning	$\leq$ 0.000001	$\leq$ 0.000001	

all strategies are statistically significant, suggesting that there is a significant difference between strategies. By ranking the strategies according to the performance scores on three series of quotes, the Deep Learning strategy can be rated the highest.

7 Conclusions

The paper delves into several crucial aspects of designing decision support systems for stock traders through the lens of a multi-agent platform. Within the presented A-Trader system, agents autonomously generate buy-sell decisions using various methods and algorithms, which serve as the foundation for crafting investment strategies. Given the diversity of these decisions and strategies, the evaluation process is overseen by a specialized program known as the Supervisor Agent. This agent enables autonomous selection of the most suitable strategy in near-real time, determining when to open or close long and short positions based on the best strategy identified for a given period. The results of the experiments described in this paper and previous experiments (see [36, 49]) highlight that the performance of specific decisions or strategies fluctuates in response to the prevailing conditions in the FOREX market. Through many experiments, it has been clearly demonstrated that no single agent or strategy consistently outperforms others across all periods. The introduction of an evaluation function further enhances this process. A-Trader distinguishes itself with its remarkable flexibility in configuring variables and evaluation functions, providing a dynamic, data-driven platform for user engagement. Investors can assess various strategies regarding returns and risks, allowing for tailored adjustments aligned with their unique requirements. In addition, A-Trader encompasses a broad spectrum of performance measures, including risk-focused metrics, underscoring the critical role of risk management. This emphasis is rooted in the inherent uncertainty and risk associated with financial investments in the FOREX market, influenced by economic cycles, interest rates, government policies, and exchange rates [53]. In contrast to existing platforms, A-Trader harnesses the consensual advice generated by multiple software agents that are proficient in fundamental, technical, and behavioral analyses [18]. Crucially, A-Trader refrains from imposing uniform evaluation strategies or functions on every user. The construction of the investment strategy assessment function remains an open endeavor, acknowledging that a one-size-fits-all solution may not exist. The paper effectively illustrates that the suitability of a linear function is expected. However, adopting a nonlinear function can be intricate and must be more readily understandable to investors, often shrouded in secrecy among financial experts. A-Trader stands as an open system, giving users the tools to fashion their strategies and seamlessly integrate strategies created in other software environments.

Table 4 Results of Friedman ANOVA test, POST-HOC (Dunn Bonferroni) for period 3

p-value	MyStrategy	Consensus	Deep learning
MyStrategy		0,006515	$\leq$ 0.000001
Consensus	0,006515		$\leq$ 0.000001
Deep learning	$\leq$ 0.000001	$\leq$ 0.000001	

This research offers several noteworthy contributions both to the scientific understanding of financial decision support systems and the practical application of these systems in real-world trading scenarios, notably in the areas of:

- Integration of Diverse Decision Sources: A-Trader integrates a wide range of decision sources by enabling multiple agents to generate independent buy-sell decisions. This diversity of sources provides a comprehensive view of market dynamics and contributes to a more holistic decision-making process.
- Agent supervision: The introduction of the Supervisor Agent serves as a pivotal contribution. This agent takes on the task of evaluating the heterogeneity of decisions and strategies. It intelligently selects the best strategy in response to the current market conditions, offering traders a pragmatic solution.
- Dynamic Strategy Selection: The research highlights that no single agent or strategy consistently outperforms others in all market conditions. This observation underscores the need for an adaptive and dynamic approach to strategy selection. Using an evaluation function empowers the Supervisor Agent to automatically identify the best strategy in near-real time, enhancing investment effectiveness and responsiveness to market changes.
- Flexibility in Evaluation Functions: A-Trader allows users to configure variables and evaluation functions, promoting a data-driven approach to user engagement. This flexibility ensures that investors can tailor their strategies based on their unique risk tolerance and performance criteria preferences.

- Risk Management Integration: A-Trader acknowledges the inherent risk and uncertainty associated with financial investments in the FOREX market. By considering a wide range of performance measures, including risk-based metrics, the platform emphasizes the importance of risk management. This is a crucial scientific contribution, as it addresses a key challenge in real-world trading.
- Open System and Interoperability: A-Trader's open system architecture is a scientific breakthrough. It allows users to seamlessly build their strategies and integrate strategies from other software environments. This interoperability enhances the practical utility of the platform, making it adaptable to the diverse needs of traders and investors.

In a pragmatic sense, A-Trader offers traders, investors, and market participants a sophisticated tool that leverages multiple agents for decision support. Provides a more adaptable and responsive approach to trading in the dynamic FOREX market. In addition, it is a pioneering platform that bridges the gap between scientific research and practical trading strategies. The limitation of this approach is the high computational complexity it entails. For example, when A-Trader runs for a month, it processes a substantial amount of data, approximately 1TB. In addition, there is a lack of direct communication between agents, and the Notification Agent acts as an intermediary to transmit signals. Consequently, this agent is a critical component of system performance. The limitation of this research is that we used only one pair of quotations in the experiments. These challenges will be the focus of further research. The ongoing research will include developing a *directional change algorithm*, an evolutionary approach to determine learning parameters, and implementing cognitive agents based on fundamental analysis and expert opinions. Further research on the application of spiking neural networks in a-Trader should also be performed. Overall, the contributions of this work extend beyond the theoretical realm, demonstrating a commitment to addressing the practical challenges traders and investors face in real-time decision making within financial markets. This multidimensional approach to financial decision support promises to enhance investment effectiveness and contribute to the broader field of algorithmic trading.

Author Contributions Conceptualization: Jerzy Korczak, Marcin Hernes; Methodology: Jerzy Korczak, Marcin Hernes; Formal analysis and investigation: Jörg Becker, Dariusz Król; Writing - original draft preparation: Marcin Hernes, Maciej Pondel, Dariusz Król; Writing - review and editing: Jerzy Korczak, Jörg Becker; Funding acquisition: Marcin Hernes; Resources: Marcin Hernes, Maciej Pondel; Supervision: Jerzy Korczak, Marcin Hernes.

Funding This research was founded by the Ministry of Science and Higher Education in Poland under the program "Regional Initiative of Excellence" [No. 015/RID/2018/19].

Data Availability Data will be made available on request.

Declarations

Competing of interest The authors have no competing interests to declare that are relevant to the content of this article.

Ethical and informed consent for data used This article does not involve any studies with human participants or animals performed by any of the authors.

References

1. Rundo F (2019) Deep lstm with reinforcement learning layer for financial trend prediction in fx high frequency trading systems. Appl Sci 9(20). https://doi.org/10.3390/app9204460
2. Chen HJ, Chen SJ, Chen Z, Li F (2019) Empirical investigation of an equity pairs trading strategy. Manage Sci 65(1):370–389
3. Tadelis S (2016) Reputation and feedback systems in online platform markets. Annual Review of Economics 8(1):321–340. https://doi.org/10.1146/annurev-economics-080315-015325
4. Freitas FD, Freitas CD, De Souza AF (2013) System architecture for on-line optimization of automated trading strategies. In: Proceedings of the 6th workshop on high performance computational finance. WHPCF '13. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2535557.2535563
5. Zhou S (2023) Forex automated trading system establishment and optimization analysis. Inform Syst Econ 63–69. https://doi.org/0.23977/infse.2023.040710
6. Roostaee MR, Abin, AA (2023) Forecasting financial signal for automated trading: An interpretable approach. Expert Syst Appl 211:118570. https://doi.org/10.1016/j.eswa.2022.118570
7. Ismail MAH, Yasruddin ML, Husin Z, Tan WK (2022) Automated trading system for forecasting the foreign exchange market using technical analysis indicators and artificial neural network. In: 2022 IEEE 18th International colloquium on signal processing & applications (CSPA), pp 63–68. https://doi.org/10.1109/CSPA55076.2022.9781856
8. Aru O, Okechukwu C (2023) development of an optimized intelligent machine learning approach in forex trading using moving average indicators. LAUTECH Journal of Engineering and Technology 17(2):18–27
9. Thompson JR (2013) Analysis of market returns using multifractal time series and agent-based simulation. PhD thesis. AAI3575853
10. Wah E, Wellman MP (2013) Latency arbitrage, market fragmentation, and efficiency: A two-market model. In: Proceedings of the fourteenth ACM conference on electronic commerce. EC '13, pp.

855–872. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/2482540.2482577
11. Ishikawa K, Nakata K (2021) Online trading models with deep reinforcement learning in the forex market considering transaction costs. arXiv:2106.03035
12. Taleb NN (2018) Election predictions as martingales: an arbitrage approach. Quantitative Finance 18(1):1–5. https://doi.org/10.1080/14697688.2017.1395230
13. Korczak J, Hernes M, Bac M (2017) Collective intelligence supporting trading decisions on forex market. In: Nguyen NT, Papadopoulos GA, Jędrzejowicz P, Trawiński B, Vossen G (eds) Computational Collective Intelligence. Springer, Berlin, Heidelberg, pp 113–122
14. Huang Z, Tanaka F (2022) Mspm: A modularized and scalable multi-agent reinforcement learning-based system for financial portfolio management. PLoS ONE 17(2):1–24. https://doi.org/10.1371/journal.pone.0263689
15. Shavandi A, Khedmati, M (2022) A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Syst Appl 208:118124. https://doi.org/10.1016/j.eswa.2022.118124
16. He F-F, Chen C-T, Huang S-H (2023) A multi-agent virtual market model for generalization in reinforcement learning based trading strategies. Appl Soft Comput 134:109985. https://doi.org/10.1016/j.asoc.2023.109985
17. Vadori N, Ardon L, Ganesh S, Spooner T, Amrouni S, Vann J, Xu M, Zheng Z, Balch T, Veloso M (2024) Towards multi-agent reinforcement learning-driven over-the-counter market simulations. Math Financ 34(2):262–347. https://doi.org/10.1111/mafi.12416
18. Loginov A, Heywood MI (2014) On evolving multi-agent fx traders. In: Esparcia-Alcázar AI, Mora AM (eds) Applications of Evolutionary Computation. Springer, Berlin, Heidelberg, pp 203–214
19. Loginov A, Heywood MI (2013) On the utility of trading criteria based retraining in forex markets. In: Esparcia-Alcázar AI (ed) Applications of evolutionary computation. Springer, Berlin, Heidelberg, pp 192–202
20. Kahneman D, Rosenfield AM, Gandhi L, Blaser T (2016) Reducing noise in decision making. Harv Bus Rev 94(12):18
21. Kočišová J, Horváth D, Kasanický T, Buša J (2012) Prediction of financial markets using agent-based modeling with optimization driven by statistical evaluation of historical data. In: Adam G, Buša J, Hnatič M (eds) Mathematical modeling and computational science. Springer, Berlin, Heidelberg, pp 308–313
22. Sarani D, Rashidi-Khazaee, DP (2024) A Deep Reinforcement Learning Approach for Trading Optimization in the Forex Market with Multi-Agent Asynchronous Distribution. https://arxiv.org/abs/2405.19982
23. Abraham A (2002) Analysis of hybrid soft and hard computing techniques for forex monitoring systems. In: 2002 IEEE World congress on computational intelligence. 2002 IEEE International conference on fuzzy systems. FUZZ-IEEE'02. Proceedings, vol 2, pp 1616–1621. https://doi.org/10.1109/FUZZ.2002.1006749
24. Sher GI (2012) Forex trading using geometry sensitive neural networks. In: Proceedings of the 14th annual conference companion on genetic and evolutionary computation. GECCO '12, pp 1533–1534. ACM, New York. https://doi.org/10.1145/2330784.2331032
25. Khosravi H, Shiri ME, Khosravi H, Iranmanesh E, Davoodi A (2009) Tactic- a multi behavioral agent for trading agent competition. In: Sarbazi-Azad H, Parhami B, Miremadi S-G, Hessabi S (eds) Advances in computer science and engineering. Springer, Berlin, Heidelberg, pp 811–815
26. Mozetic I, Gabrovsek P, Novak PK (2018) Forex trading and twitter: Spam, bots, and reputation manipulation. CoRR **abs/1804.02233**
27. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527
28. Kuremoto T, Kimura S, Kobayashi K, Obayashi M (2014) Time series forecasting using a deep belief network with restricted boltzmann machines. Neurocomputing 137:47–56. https://doi.org/10.1016/j.neucom.2013.03.047
29. Yang S, Chen B (2023) Snib: Improving spike-based machine learning using nonlinear information bottleneck. IEEE Transactions on Systems, Man, and Cybernetics: Systems 53(12):7852–7863. https://doi.org/10.1109/TSMC.2023.3300318
30. Yang CBS (2023) Effective surrogate gradient learning with high-order information bottleneck for spike-based machine intelligence. IEEE Transactions on neural networks and learning systems. https://doi.org/10.1109/TNNLS.2023.3329525
31. Yang S, Pang Y, Wang H, Lei T, Pan J, Wang J, Jin Y (2023) Spike-driven multi-scale learning with hybrid mechanisms of spiking dendrites. Neurocomputing 542:126240. https://doi.org/10.1016/j.neucom.2023.126240
32. Gerlein EA, McGinnity M, Belatreche A, Coleman S (2016) Evaluating machine learning classification for financial trading: An empirical approach. Expert Syst Appl 54:193–207. https://doi.org/10.1016/j.eswa.2016.01.018
33. Talebi H, Hoang W, Gavrilova ML (2014) Multi-scale foreign exchange rates ensemble for classification of trends in forex market. Procedia Computer Science 29:2065–2075. https://doi.org/10.1016/j.procs.2014.05.190
34. Mayo M (2012) Evolutionary data selection for enhancing models of intraday forex time series. In: Di Chio C at al (eds) Applications of evolutionary computation, Springer, Berlin, Heidelberg, pp. 184–193
35. Agarwal S (2011) Toward a push-scalable global internet. In: 2011 IEEE Conference on computer communications workshops (INFOCOM WKSHPS), pp 786–791. https://doi.org/10.1109/INFCOMW.2011.5928918
36. Korczak J, Hernes M, Bac M (2013) Risk avoiding strategy in multi-agent trading system. In: Ganzha M, Maciaszek MPL (ed.) Proceedings of the 2013 federated conference on computer science and information systems, IEEE, Los Alamitos, CA, pp 1119–1126
37. Korczak J, Hernes M, Bac M (2014) Performance evaluation of decision-making agents' in the multi-agent system. In: 2014 Federated Conference on Computer Science and Information Systems, pp 1171–1180. https://doi.org/10.15439/2014F188
38. Longo JM (2014) Trading and Investment Strategies in Behavioral Finance, Wiley Ltd, Hoboken, New Jersey. Chap. 27, pp 495–512. https://doi.org/10.1002/9781118813454.ch27
39. Shiller RJ (2012) Finance and the Good Society. Princeton University Press, New Jersey, United States
40. Lento C (2009) The combined signal approach to technical analysis: A review & commentary. SSRN Electron J. https://doi.org/10.2139/ssrn.1410899
41. Hernes M, Nguyen NT (2007) Deriving consensus for hierarchical incomplete ordered partitions and coverings. J Univ Comput Sci 13(2):317–328
42. Hernes M, Sobieska-Karpińska J (2016) Application of the consensus method in a multiagent financial decision support system. Inf Syst E-bus Manag 14(1):167–185. https://doi.org/10.1007/s10257-015-0280-9
43. Korczak J, Hernes M (2017) Deep learning for financial time series forecasting in a-trader system. In: 2017 Federated conference on computer science and information systems (FedCSIS), pp 905–912. https://doi.org/10.15439/2017F449
44. Hu D, Schwabe G, Li X (2015) Systemic risk management and investment analysis with financial network analytics: research opportunities and challenges. Financ Innov 1(1):2. https://doi.org/10.1186/s40854-015-0001-x

45. Hussain OK, Dillon TS, Hussain FK, Chang EJ (2013) Risk Assessment Phase: Financial Risk Assessment in Business Activities, pp 151–185. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-28690-2_6
46. Lückoff P (2011) Mutual Fund Performance and Performance Persistence. Gabler Verlag, Wiesbaden.https://doi.org/10.1007/978-3-8349-6527-1
47. Qiu Z (2016) Discussion of investment analysis method in the new round of the china stock bull market. In: Li M, Zhang Q, Zhang J, Li Y (eds) Proceedings of 2015 2nd international conference on industrial economics system and industrial security engineering, Springer, Singapore, pp 311–317
48. Yao Y-y, Zhang R-s (2016) Empirical research on efficiency measure of financial investment in education based on se-dea. In: Cao B-Y, Liu Z-L, Zhong Y-B, Mi H-H (eds) Fuzzy systems & operations research and management, Springer, Cham, pp 389–402
49. Korczak J, Hernes M (2018) Performance evaluation of trading strategies in multi-agent systems - case of a-trader. In: 2018 Federated conference on computer science and information systems (FedCSIS), pp 839–844
50. Mancas D, Udristoiu S, Manole E, Lapadat B (2008) A comparison of multi-agents competing for trading agents competition. WSEAS TRANS- ACTIONS on COMPUTERS 7(12):1916–1926
51. Calzi ML, Milone L, Pellizzari P (2010) Progress in Artificial Economics: Computational and Agent-Based Models. Springer, Berlin, Heidelberg
52. Eiben AE, Smith JE (2015) Introduction to Evolutionary Computing, 2nd edn. Springer, Berlin, Heidelberg, Natural Computing Series. https://doi.org/10.1007/978-3-662-44874-8
53. LeBaron B (2011) Active and passive learning in agent-based financial markets. Eastern Economic Journal 37(1):35–43. https://doi.org/10.1057/eej.2010.53

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

Marcin Hernes is an associate professor, head of Department of Process Management, and chair of the Center for Intelligent Management Systems at Wroclaw University of Economics and Business. His research has been concentrated on artificial intelligence, knowledge management, decision support systems, management systems, and cognitive architectures. He has authored over 200 peer-reviewed papers, published in international journals, and presented at conferences. Prof. Hernes is a member of IEEE, The Polish Artificial Intelligence Society, and Scientific Association of Business Informatics. He was awarded the award of the Rector of Wroclaw University of Economics and Business for scientific achievements several times. Marcin Hernes is also a practitioner in management systems, coordination and automation of production processes and multi-agent decision support systems, and the author of dozens of computer applications in industrial companies and public administration entities.

Jerzy Korczak is an expert in data science, computational intelligence, image analysis, and recommendation systems. He currently serves as a professor at the International University of Logistics and Transport in Wrocław, Poland, where he leads research in business machine learning and intelligent management systems. With over 300 publications to his name and numerous research projects, his work has garnered significant recognition, evidenced by an h-index of 22 and over 1,800 citations, according to Google Scholar. Professor Korczak has more than 40 years of academic experience, including teaching and leadership in business-oriented AI projects. He has been an invited speaker at numerous conferences and events across Europe, US and Asia. Previously, he was a professor and head of the Department of Information Technology at the University of Economics in Wrocław from 2006 to 2016. From 1986 to 2006, he was a professor of Informatics at Université Louis Pasteur in Strasbourg, France, where he led the CNRS team on Image Data Mining at the Laboratory of Sciences of Image, Computer Science, and Remote Sensing (LSIIT, URA CNRS 1871). During this time, he also served as the director of the Research Team in Information Technology in Strasbourg.

Dariusz Krol is currently a Professor at Wrocław University of Science and Technology, Vice-Dean of General Affairs of the Faculty of Information and Communication Technology, and Head of the Knowledge Engineering Group in the Department of Applied Informatics. He has held positions as Marie Curie Senior Research Fellow at Bournemouth University and DAAD Research Fellow at the Friedrich-Alexander Universität. His research and teaching interests are oriented toward knowledge engineering applications, data quality, computational intelligence, and cognitive technologies for adaptive and sustainable manufacturing systems. He has published over 150 peer-reviewed papers, co-authored three books, and co-edited four. He was an invited lecturer for many events throughout the UK, Europe, and Asia in the last ten years. Prof. Król has 30 years of academic experience and about 20 years as CEO and Head Leader in R&I projects. He also works as a project evaluator for the European Commission. His prestigious contribution in the field was rewarded by IBM and granted by the European Commission, the Polish National Center for Science, the Polish Ministry of Science and Higher Education, and DAAD. He has a great deal of hands-on experience with EU-funded projects and programs. Moreover, he has been elected to the Polish Accreditation Committee in recognition of their professional standing.

Maciej Pondel is a Doctor of Economics, associate professor, and Head of the Department of Business Intelligence in Management at the Faculty of Management, Wrocław University of Economics. Tech Lead in the Data & AI department at Unity Group. He holds a degree in Computer Science from the Faculty of Computer Science and Management at Wrocław University of Science and Technology. Previously, he worked as a programmer, project manager, consultant, and product manager, where he was responsible, among other things, for the development of the Upsaily system, which uses artificial intelligence techniques to analyze sales data. He has led numerous commercial as well as research and development projects. He has been involved in the IT industry since 2000. His academic and practical expertise focuses on artificial intelligence, machine learning, the application of machine learning in business management (particularly in sales and marketing), as well as databases, data warehouses, and Business Intelligence systems and their role in digital transformation. He is also the lead researcher in R&D projects related to the use of AI techniques in medicine. He is the author of numerous scientific and business articles and has been a speaker at both national and international conferences. A passionate advocate for innovation and the integration of science and business.

Jörg Becker is head of the Department of Information Systems of the University of Münster and of the European Research Center for Information Systems (ERCIS), an institutionalized network of well known European (and non European) Research Institutions in the field of Information Systems and Business Informatics. Jörg is Professor for Information Systems and directs the Chair for Information Systems and Information Management. He holds an honorary professorship at the National Research University - Higher School of Economics (NRU-HSE) in Moscow and is member of the North Rhine-Westphalian Academy of Sciences, Humanities and the Arts. He was granted doctorates honoris causa (Dr. h.c.) from the Universities of Turku (Finland) and Voronesh (Russia). From 2008 to 2016, he served the University of Münster as Pro-Rector for Strategic Planning and Quality Assurance and he was the CIO of the university. He acts as a chairperson of the National E-Government Competence Center NEGZ. He was advisor to former chancellor Angela Merkel concerning all topics of E-Government. His research interests cover Information Modelling including Reference Modelling for Data and Processes, Management Information Systems, Hybrid Value Creation, Business Process Management, E-Government, and Retail Information Systems. Jörg has published in renowned outlets, including *MIS Quarterly (MISQ), European Journal of Information Systems (EJIS), Business & Information Systems Engineering (BISE), Information Systems Frontiers (ISF), Information Systems Journal (ISJ)*, and *Business Process Management Journal (BPMJ)*. He has authored and edited numerous books, including *Retail Information Systems, Process Management, Modernizing Processes in Public Administrations*, and *Reference Modeling*. He is editor-in-chief of *Information Systems and e-Business Management.*