

INTERPRETABLE MACHINE LEARNING FOR FINANCIAL APPLICATIONS

Boris Kovalerchuk

Dept. of Computer Science, Central Washington University, USA

Evgenii Vityaev

Sobolev Institute of Mathematics, Russian Academy of Sciences, Russia

Alexander Demin

Ershov Institute of Informatics, Russian Academy of Sciences, Russia

Antoni Wilinski

Dept. of Electronics and Computer Science, Koszalin University of Technology, Poland

Abstract. This chapter describes machine learning (ML) for financial applications with focus on interpretable relational methods. It presents financial tasks, methodologies and techniques in this ML area. It includes time dependence, data selection, forecast horizon, measures of success, quality of patterns, hypothesis evaluation, problem ID, method profile, attribute-based and interpretable relational methodologies. The second part of the chapter presents ML models and practice in finance. It covers the use of ML in portfolio management, design of interpretable trading rules, discovering money-laundering schemes using the machine learning methodology.

Keywords: finance time series, relational machine learning, visual knowledge discovery, decision tree, neural network, deep neural network, success measure, portfolio management, stock market, trading rules.

October. This is one of the peculiarly dangerous months to speculate in stocks in. The others are July, January, September, April, November, May, March, June, December, August and February. Mark Twain, 1894

1. INTRODUCTION: FINANCIAL TASKS

Forecasting stock market, currency exchange rate, bank bankruptcies, understanding and managing financial risk, trading futures, credit rating, loan management, bank customer profiling, and money laundering analyses are core financial tasks for machine learning [1, 26, 55]. Stock market forecasting includes uncovering market trends, planning investment strategies, identifying the best time to purchase the stocks and what stocks to purchase. Financial institutions produce huge datasets that build a foundation for approaching these enormously complex and dynamic problems with machine learning tools. Potential significant benefits of solving these problems have motivated extensive research for years.

Almost every computational method has been explored and used for *financial modeling for a long time*. We will name just a few studies: Monte-Carlo simulation of option pricing, finite-difference approach to interest rate derivatives, and fast Fourier transform for derivative pricing [6,10,14,34]. Such developments augment traditional technical analysis of stock market curves [54].

Machine Learning (ML) as a process of *discovering useful patterns, correlations* has its own niche in financial modeling. Similarly, almost every ML method has been used in financial modeling. An incomplete list includes linear and non-linear models, multi-layer neural networks, k-means and hierarchical clustering; k-nearest neighbors, decision tree analysis, regression, ARIMA, principal component analysis, Bayesian learning; and most recently deep learning for price formation in financial markets, directional movements prediction of S&P500 index, financial sentiment analysis and others [16, 32, 34, 38, 58, 57, 59].

Less traditional relational deterministic and probabilistic methods based on the First Order Logic (FOL) are growing in many domains including finance [19,20,31, 39, 40-42, 50-53,54, 60-64, 66] due to current surge of interest in interpretable machine learning and AI.

Bootstrapping and other evaluation techniques have been extensively used for improving ML results. Specifics of financial series analyses with ARIMA, neural networks, relational methods, support vector machines and traditional technical analysis is discussed in [3,40,49,54].

The naive approach to ML in finance assumes a cookbook instruction on “how to achieve the best result”. Some publications continue to foster this unjustified belief. In fact, the only realistic approach proven to be successful is providing comparisons between different methods showing their strengths and weaknesses relative to problem characteristics (problem ID) conceptually, and leaving for user the selection of the method, which likely fits the specific user’s problem. This means clear understanding that ML is still more art than hard science. Fortunately, now there is growing number of books that discuss the issues of matching tasks and methods in a regular way [21, 40]. For instance, understanding the power of FOL If-Then rules over the decision trees can significantly change and improve ML design. In comparison with other fields such as geology or medicine, where test of the forecast is expensive, difficult, and even dangerous, a trading forecast can be tested next hour or day, without cost and capital risk involved in real trading.

Attribute-based learning methods such as neural networks, the nearest neighbor’s method, and decision trees dominate in ML financial applications. They are relatively simple, efficient, and handle noisy data, but are limited in using background knowledge and complex relations.

Relational machine learning techniques based on the first order logic, which includes Inductive Logic Programming (ILP) [50-53, 24, 40] intend to overcome these limitations. Previously these methods have been relatively computationally inefficient and had rather limited facilities for handling numerical data [12]. These methods are enhanced in both aspects [40] and are expanded to several domains from bioinformatics to robotics. We expect that the applications of these methods will grow due to current demand for interpretable ML. Various publications explored the

use of ML methods like hybrid architectures of neural networks with genetic algorithms, chaos theory, and fuzzy logic in finance. Many proprietary financial ML applications exist but rarely reported.

2. SPECIFICS OF MACHINE LEARNING IN FINANCE

Specifics of machine learning in finance are coming from the need to:

- forecast multidimensional time series with high level of noise;
- accommodate specific efficiency criteria (e.g., the max of profit);
- make coordinated multiresolution forecast (minutes, days, and so on);
- incorporate text stream as input data (e.g., Enron case and September 11);
- *explain the forecast* and the *forecasting model* (“black box” models have limited interest and future for significant investment decisions);
- discover very *subtle patterns* with a *short lifetime*; and
- incorporate the impact of market players on market regularities.

The *efficient market theory/hypothesis* discourages attempt to discover long-term stable trading rules/regularities with significant profit. This theory is based on the idea that if such regularities exist, they would be discovered and used by most of the market players. This would make rules less profitable and eventually useless or even damaging. The market efficiency theory does not exclude that hidden *short-term local conditional regularities* may exist. These regularities cannot work “forever,” they should be corrected *frequently*. It has been shown that the financial data are not random and that the efficient market hypothesis is merely a subset of a larger *chaotic market hypothesis* [23]. This hypothesis does not exclude successful short-term forecasting models for prediction of chaotic time series [13]. ML does not try to accept or reject the efficient market theory. It creates *tools* for discovering subtle short-term conditional patterns in financial data. Thus, retraining should be a permanent part of ML in finance and any claim that a silver bullet trading has been found should be treated similarly to claims that a *perpetuum mobile* has been discovered.

The impact of market players on market regularities stimulated a surge of attempts to use ideas of *statistical physics* in finance [11]. If an observer is a large marketplace player then such observer can potentially change regularities of the marketplace dynamically. Attempts to forecast in such dynamic environment with thousands active agents lead to much more complex models than traditional ML models designed for. Therefore, such interactions are modeled using ideas from statistical physics. The *physics approach* in finance [35, 46] is also known as “econophysics” and “physics of finance”. The major difference from the ML approach is that the physics approach is deeper integrated into the finance subject matter. ML approach covers empirical models and regularities derived directly from data and almost only from data with little domain knowledge explicitly involved. Historically, deep field-specific theories emerge after accumulating enough empirical regularities. We see that the future of ML in finance would be to generate more regularities that are empirical and combine them with domain knowledge via generic analytical ML approach [48]. First attempts in this direction are presented in [40] that exploit power of relational machine learning as a mechanism that permits to encode domain knowledge in the first order logic language.

Data selection and forecast horizon. The selection of data for ML in finance is tightly connected to the selection of the target variable. There are several options for target variable y : $y=T(k+1)$, $y=T(k+2)$, ..., $y=T(k+n)$, where $y=T(k+1)$ represents forecast for the next time moment, and $y=T(k+n)$ represents forecast for n moments ahead. Selection of dataset T and its size for a specific desired forecast horizon n is a significant challenge. For stationary stochastic processes, the answer is well known -- a better model can be built for longer training duration. For financial time series such as S&P500 index, this is not the case [47]. Longer training duration may produce many contradictory profit patterns that reflect bear and bull market periods. Models built using too short durations may suffer from overfitting, and hardly be applicable to the situations, where market is moving from the bull period to the bear period. The standard ML assumes that the model quality does not depend on *frequency* of its use. In finance frequency of trading a model parameter, because *the model quality* includes both the accuracy of prediction and its profitability. The frequency of trading affects the profit, the trading rules and strategy.

Measures of success. Traditionally the quality of forecasting models is measured by the standard deviation between forecast and actual values on training and testing data. For trading tasks two models with the same standard deviation can provide very different trading return [40]. A more specific success measure in financial ML are Average Monthly Excess Return (AMER):

$$AMER_j = R_{ij} - \beta_i R500_j - \sum_{j=1:12} (R_{ij} - \beta_i R500_j) / 12$$

where R_{ij} is the average return for the S&P500 index in industry i and month j and $R500_j$ is the average return of the S&P 500 in month j . The β_i values adjust the AMER for the index's sensitivity to the overall market. A second measure of return is Potential Trading Profits (PTP) that shows investor's trading profit versus the alternative investment based on the broader S&P 500 index.

Quality of patterns and hypothesis evaluation. A typical approach is the testing of the null hypothesis H that pattern P is not statistically significant at level α . A meaningful statistical test requires that pattern parameters such as the month(s) of the year and the relevant sectoral index in a trading rule pattern P have been chosen *randomly* [30]. In many tasks, this is not the case.

Greenstone and Oyer argue that in the “summer swoon” trading rule, the parameters are not selected randomly. This means that rigorous test would require testing a different null hypothesis not only about one “significant” combination, but also about the “*family*” of combinations. A bootstrapping method was used to evaluate the statistical significance of such a hypotheses. Greenstone and Oyer [30] suggest a simple computational method – combining individual *t-test* results by using the Bonferroni inequality that given any set of events A_1, A_2, \dots, A_n , the probability of their union is smaller than or equal to the sum of their probabilities:

$$P(A_1 \& A_2 \& \dots \& A_k) \leq \sum_{i=1:k} P(A_i).$$

3. ASPECTS OF ML METHODOLOGY IN FINANCE

ML in finance typically follows a set of general for any ML task steps such as problem understanding, data collection and refining, building a model, model evaluation and deployment. An important step is adding expert-based rules in ML loop when dealing with absent or insufficient data.

“Expert mining” is a valuable additional source of regularities. However, in finance, expert-based learning systems respond slowly to the market changes [18]. A technique for efficiently mining regularities from an *expert’s perspective* has been offered in [40]. Such techniques need to be integrated into financial ML loop similar to what was done for medical ML applications [41].

Attribute-based and relational methodologies. Several parameters characterize machine-learning methodologies for financial forecasting. Data categories and mathematical algorithms are most important among them. The first data type is represented by *attributes* of objects, that is each object x is given by a set of values $A_1(x), A_2(x), \dots, A_n(x)$. The common ML methodology assumes this type of data, known as an *attribute-based* or *attribute-value methodology*. It covers a wide range of statistical and connectionist (neural network) methods.

The *relational data type* is a second type, where objects are represented by their relations with other objects, for instance, $x > y$, $y < z$, $x > z$. Here, we may not know that $x=3$, $y=1$ and $z=2$. Thus, attributes of objects are not known, but their relations are known. Objects may have different attributes (e.g., $x=5$, $y=2$, and $z=4$), but still have the same relations. The *relational methodology* is based on such a relational data type.

Another data characteristic important for financial modeling methodology is an actual *set of attributes* involved. A fundamental analysis uses all available attributes, but technical analysis uses only time series such as stock price and parameters derived from it. Most popular time series are index value at open, index value at close, highest index value, lowest index value and trading volume and lagged returns from the time series of interest. Fundamental factors include the price of gold, retail sales index, industrial production indices, and foreign currency exchange rates. Technical factors include variables that are derived from time series such as moving averages. The next characteristic is a form of the relationship between objects. Many ML methods assume its *functional form*, e.g., linearity of the border that discriminates between two classes that is often hard to justify. Relational ML does not assume a functional form, but *learns symbolic relations* on numerical data of financial data.

Attribute-based relational methodologies. Below we discuss a combination of attribute-based and relational methodologies to mitigate their difficulties. Historically, relational ML was associated with *Inductive Logic Programming* (ILP), which is a deterministic technique in its purest form. The typical claim about relational ML is that it cannot handle large data assuming that input data are relations, which take more space than individual attributes. Computing relations from attribute-based data on demand resolves this issue. For instance, to explore a relation, $Stock(t) > Stock(t+k)$ for k days ahead we can compute it for every pair of stock data as needed. Multiple studies had shown that relational ML is most suitable for applications, where *structure can be extracted from the instances*.

Problem ID and method profile. Selection of a method for discovering regularities in financial time series is a very complex task. Uncertainty of problem descriptions and method capabilities are among the most obvious difficulties in this process. To deal with this issue a unified vocabulary and a framework for matching problems and methods have been proposed in [21]. A problem is described using a set of *desirable values* (problem ID profile) and a method is described using its *capabilities* in the same terms. Use of unified terms (*dimensions*) for problems and methods enhances capabilities of comparing alternative methods. Introducing dimensions also accelerates their clarification. Next, users should not be forced to spend time determining a method's capabilities (values of dimensions for the method). This is a task for developers, but users should be able to identify desirable values of dimensions using natural language terms [21]. Along these lines, Table 4 indicates advantages and shortcomings of neural networks for stock price forecasting related to explainability, usage of logical relations and tolerance for sparse data. The strength of neural networks includes modeling complex functions and using a high number of fundamental and technical factors.

Table 4. Comparison of model quality and resources for neural networks.		
<i>Dimension</i>	<i>Desirable value for stock price forecast problem</i>	<i>Capability of a neural network method</i>
Accuracy	Moderate	High
Explainability	Moderate to High	low to Moderate

Response speed	Moderate	High
Ease to use logical relations	High	Low
Ease to use numerical attributes	High	High
Tolerance for noise in data	High	Moderate to high
Tolerance for sparse data	High	Low
Tolerance for complexity	High	High
Independence from experts	Moderate	High

Relational machine learning in finance. Decision trees are very popular in ML applications. They provide human readable, consistent rules, but discovering small trees for complex problems can be a significant challenge in finance [40]. In addition, DT rules fail to compare two attribute values, while it is possible with relational methods. Several publications strengthened that relational ML area is moving toward probabilistic first-order rules to avoid the limitations of deterministic systems. Relational methods in finance such as Machine Method for Discovering Regularities (MMDR) [40] are equipped with probabilistic mechanism, which is necessary for time series with high level of noise. MMDR is well suited to financial applications given its ability to handle numerical data with high levels of noise [18]. In computational experiments, trading strategies developed based on MMDR consistently outperform trading strategies developed based on other ML methods, and the buy-and-hold strategy.

4. ML MODELS AND PRACTICE IN FINANCE

Prediction tasks in finance typically are posed as: (1) *straight prediction* of the market numeric characteristic, e.g., stock return or exchange rate, and (2) the prediction whether the market characteristic will *increase or decrease* no less than some threshold. Thus, the difference between ML methods for (1) or (2) can be less obvious, because (2) may require numeric forecast. Another type of task assessment of investing risk [8]. It used a DT technique C5.0 and neural networks to a dataset of 27 variables for 52 countries whose investing risk category was assessed in a Wall Street Journal survey of international experts.

4.1 Portfolio management and neural networks

Historically, the neural network most commonly used by financial institutions was a multilayer perceptron (MLP) with a single hidden layer of nodes for time series prediction. The peak of such research activities was in mid 1990s [2, 27] that covered MLP and recurrent neural networks.

Other neural networks used in prediction are time delay networks, Elman networks, Jordan networks, GMDH, multi-recurrent networks [28]. Later we review the most recent work on deep neural networks (DNN).

Below we present typical steps of *portfolio management* using the neural network forecast of return values.

1. Collect 30- 40 historical fundamental and technical factors for stock S_1 , say for 10-20 years.
2. Build a neural network NN_1 for predicting the return values for stock S_1 .
3. Repeat steps 1 and 2 for every stock S_i , that is monitored by the investor. Say 3000 stocks are monitored and 3000 networks, NN_i are generated.
4. Forecast stock return $S_i(t+k)$ for each stock i and k days ahead (say a week, seven days) by computing $NN_i(S_i(t))=S(t+k)$.
5. Select n highest $S_i(t+k)$ values of predicted stock return.
6. Compute a total forecasted return of selected stocks, T and compute $S_i(t+k)/T$. Invest to each stock proportionally to $S_i(t+k)/T$.
7. Recompute NN_i model for each stock i every k days adding new arrived data to the training set. Repeat all steps for the next portfolio adjustment.

These steps show why neural networks became so popular in finance. Potentially all steps above can be done automatically including actual investment. Even institutional investors may have no resources to analyze manually the 3000 stocks and their 3000 neural networks every week. If investment decisions are made more often, say every day, then the motivation to use neural networks with their high adaptability is even more evident.

This consideration shows challenges of ML in finance – the need to build models that can be very quickly evaluated in both accuracy and *interpretability*. Because NN are difficult to interpret even without time limitation steps 1-6 have been adjusted by adding more steps after step 3 that include *extracting interpretable rules* from the trained neural networks, and improving prediction accuracy using rules, e.g., [28] and more recently for DNN.

It is likely that extracting rules from the neural network is a *temporary solution*. It would be better to extract rules directly from data without introducing neural network artifacts to rules and potentially overlooking some better rules because of this. A growing number of computational experiments support this claim, e.g., [40] on experiments with S&P500, where first order rules built directly from data outperformed backpropagation neural networks.

The logic of using ML in trading futures is like portfolio management. The most significant difference is that it is possible to substitute the numeric forecast of actual return for the categorical forecast, will it be profitable to buy or sell the stock at a price $S(t)$ on date t . This corresponds to long and short terms used in the stock market, where *Long* stands for buying the stock and *Short* stands for sell the stock.

Recently, due to the success of Deep Learning methods in various fields, the interest in using neural networks in financial forecasting problems has returned. Researchers actively use new neural network architectures and learning algorithms such as Deep Neural Network Classifier (DNNC), Long-Short Term Memory (LSTM), Gated Recurring Unit (GRU) and Convolutional Neural Networks (CNN) to solve financial tasks, including in conjunction with reinforcement learning to develop optimal solutions for the purchase and sale of assets [4,16,17,37,56]. However, in general, analyzing the results, we can conclude that the use of Deep Neural Networks (DNN) does not provide an obvious advantage compared to simpler neural network models.

4.2 Interpretable trading rules and relational ML

Comparison of approaches. Below we present the categories of rules, which can be discovered by different techniques. *Categorical rules* predict categorical attributes, such as increase/decrease, buy/sell. A typical example of a *monadic categorical rule* is the following rule:

If $S_i(t) < \text{Value}_1$ and $S_i(t-2) < \text{Value}_2$ then $S_i(t+1)$ will increase.

Here, $S_i(t)$ is a continuous variable, e.g., stock price at the moment t . If $S_i(t)$ is a discrete variable, then Value_1 and Value_2 are taken from m discrete values. This rule is called monadic, because it compared a *single attribute* value with a *constant*. Such rules can be discovered from trained decision trees by tracing their branches to the terminal nodes. Unfortunately, decision trees produce only such rules.

The technical analysis rule below is a *relational categorical rule*, because it *compares* values of 5 and 15 day moving averages (ME_5 and ME_{15}) and derivatives of moving averages for 10 and 30 days (DerivativeME_{10} , DerivativeME_{30}):

$$\text{If } \text{ME}_5(t) = \text{ME}_{15}(t) \ \& \ \text{DerivativeME}_{10}(t) > 0$$

$$\text{DerivativeME}_{30}(t) > 0 \text{ then Buy stock at moment } (t+1).$$

This rule can be read as: If moving averages for 5 and 15 days are equal and derivatives for moving averages for 10 and 30 days are positive, then buy stock on the next day. Thus, classical for stock market technical analysis is superior to decision trees. This rule is written in a *first order logic* form. Typically, technical analysis rules are not discovered in this form, but relational ML technique does.

Classical categorical rules assume crisp relations such as $S_i(t) < \text{Value}_1$ and $\text{ME}_5(t) = \text{ME}_{15}(t)$. More realistic is to assume that $\text{ME}_5(t)$ and $\text{ME}_{15}(t)$ are equal only approximately and Value_1 is not exact. Fuzzy logic and rough sets rules are used in finance to work with “*soft*” relations [9, 40]. The logic of using “*soft*” trading rules includes the conversion of time series to soft objects, discovering a temporal “soft” rule from stock market data, discovering a temporal “soft” rule from experts (“expert mining”), testing consistency of expert rules and rules extracted from data, and finally using rules for forecasting and trading.

Unlike neural networks, below we use a probabilistic logic network (PLN) approach that is related to Probabilistic logic network [29]. The main distinct characteristics of PNL are as follows.

1. It uses first-order logic to record patterns, which makes it quite capable to discover a wide range of patterns.
2. These patterns are readable and understandable and belong to the field of Explainable Artificial Intelligence [22].
3. The inductive-statistical inference (I-S inference) is used to derive the

predictions.

4. The disadvantage of I-S inference is its statistical ambiguity, which was resolved recently [60-62].

In PLN we define the maximally specific rules, the I-S derivation by which is logically consistent [62]. Inductive inference of rules is carried out by a special semantic probabilistic inference, during which the conditional probability of rules strictly grows, which distinguishes it from probabilistic logical programming. The most specific rules are obtained in the training process that uses semantic probabilistic inference which is implemented in the Discovery software system [63, 64, 66].

4.3 Relational machine learning for trading

This section presents the application of first order logic relational approach to develop a trading system for the S&P500 index.

Let $c(t_1), \dots, c(t_n)$ be values of S&P500 close at time moments t_1, t_2, \dots, t_n . The goal is to predict the direction of S&P500 (up or down) 5 days ahead. Thus, it is to predict the truth of the predicates $c(t_i) < c(t_i + 5)$ and $c(t_i) > c(t_i + 5)$.

First, we formulate the hypotheses to be tested on time series. The experience of classical technical analysis shows that before changing the direction of their movement, prices form the so-called figures of technical analysis [Murphy, 1999]. Based on this idea, a class of hypotheses and predicates were developed. The following predicates were used for this:

Predicate $c(t_i) < c(t_j)$, which compares the time series value at t_i and t_j ;

Predicate $ext(t_i) = \delta_i$, where δ_i is -1 or 1 and $ext(t_i) = -1$ means a local minimum of $c(t)$ is at time t_i .

$$(ext(t_i) = -1) \Leftrightarrow (c(t_i) < c(t_i - 1)) \& (c(t_i) < c(t_i + 1)),$$

$ext(t_i) = 1$ means a local maximum of $c(t)$ is at time t_i .

$$(ext(t_i) = 1) \Leftrightarrow (c(t_i - 1) < c(t_i)) \& (c(t_i + 1) < c(t_i)).$$

Below we use notation $ext(t_1, \dots, t_n) = \langle \delta_1, \dots, \delta_n \rangle$, which is equivalent to $(ext(t_1) = \delta_1) \& \dots \& (ext(t_n) = \delta_n)$.

The hypotheses will test whether a certain combination of points of local minima and maxima has been formed in the past of the time series. If such a combination is found, a forecast is made five days ahead. Here is an example rule:

$$\forall t_1 \exists t_2, t_3, t_4, t_5, t_6 : (ext(t_1, t_2, t_3, t_4, t_5, t_6) = \langle -l, l, -l, l, -l, l \rangle) \& (c(t_1) < c(t_2)) \& (c(t_2) < c(t_4)) \& (c(t_3) < c(t_2)) \& (c(t_5) < c(t_6)) \& (c(t_6) < c(t_4)) \rightarrow (c(t_1) > c(t_1 + 5))$$

(1)

Figure 1 shows a general view of the figure described by this rule. It says that if the time series formed a figure described by this rule, then the value of the series in five days will become less than the value of the series on the current day. This figure resembles the well-known figure “head-shoulders” of technical analysis.

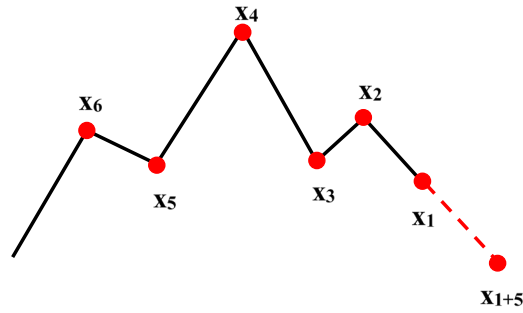


Fig. 1. Figure described by rule (1).

Since several different rules may work on the same trading day, we may have several forecasts, with different probability, predicting the direction of price movement. Moreover, since the rules are probabilistic, the forecasts may contradict each other. The next step is determining the trading strategy, which deals with contradictory predictions.

In this experiment, the final forecast is based on a comparison of the *maximum probabilities* of forecasts of an increase or decrease in price after five days. The *trading strategy* is to buy if the maximum probability that the price rises in five days is more than the maximum probability that it will fall and sell otherwise. A buy signal means to open a buy position for five days (buy and hold for five days, then close), if at the same time a

sell position is already open, then it must be closed, If $Signal(t)=0$, then do not trade on this day.

Testing. We used for testing the method of moving control on the 9-year time interval. This interval includes 2065 trading days. A training window of size of 500 trading days and a testing interval of 100 days were used. Thus, for the entire testing period, the system passed 16 training and testing cycles, and the total testing interval was 1565 trading days. The quality of the trading system during testing was evaluated by modeling the real trading by assessing the potential profitability of the system. It requires ensuring that the system has a stable positive expected value:

$$P_{Win}Trade_{Win} + P_{Loss}Trade_{Loss} > 0, \quad (2)$$

where P_{Win} is the probability of gain, $Trade_{Win}$ is the average gain, P_{Loss} is the probability of loss, and $Trade_{Loss}$ is the average loss. The system trades with one unit of the contract and does not use stop-loss orders. While it is not optimal, it allows fairly objectively evaluate the expected value (2).

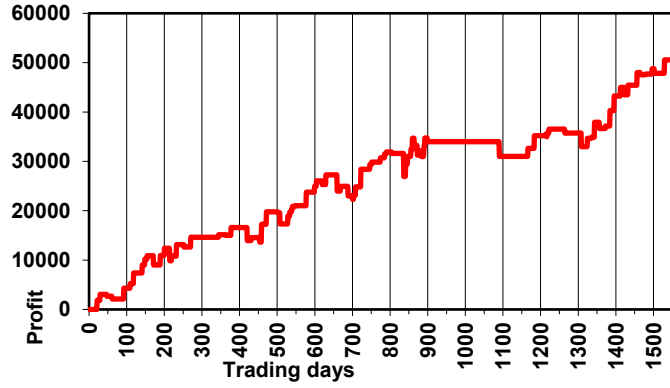


Fig. 2. Results of testing of the trading system.

Figure 2 shows test results with the dynamics of capital growth over the entire test period (1565 days) where trading days are numbered from 1 to

1565. Table 1 presents the indicators characterizing this trading system. Annual rate of return in relation to the maximum drawdown of the account characterizes the rate of return per unit of risk, where the value of the maximum drawdown of the account acts as a measure of risk. It is calculated by the formula:

$$\text{Annual Rate of return} = \frac{(\text{Profit})}{(\text{Maximum drawdown})} \cdot \frac{365}{(\text{Number of trading days})}.$$

Table 1. Characteristics of the trading system.

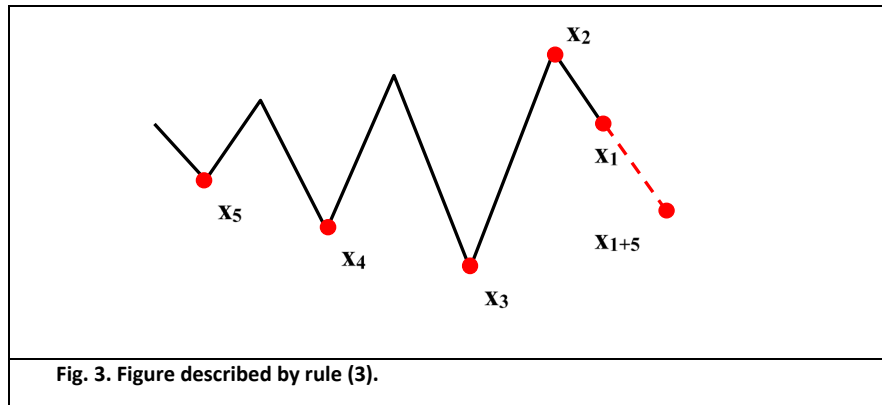
Indicator	Value
Profit	50563
Maximum drawdown	-4960
Expected value, see (3)	568
Percentage of profitable trades	69 %
Annual rate of return in relation to the maximum drawdown of the account	238 %

As the test results show, this trading system has fairly stable positive expected values. Figure 2 shows that the system provides a steady growth of capital over the entire test period. The use of capital management methods and risk limitation would significantly improve this trading system; however, this is a topic for a separate study.

Here is an example of the rules that were found during training:

$$\begin{aligned} &\forall t_1 \exists t_2, t_3, t_4, t_5 : (ext(t_1, t_2, t_3, t_4, t_5) = \langle -1, 1, -1, -1, -1 \rangle) \& (c(t_5) < c(t_2)) \& \\ &(c(t_4) < c(t_1)) \& (c(t_4) < c(t_2)) \& (c(t_3) < c(t_4)) \& \& (c(t_1) < c(t_5)) \rightarrow (c(t_1) > c(t_1 + 5)) \end{aligned} \quad (3)$$

This rule predicts a price reduction in five days. Figure 3 shows the general view of the figure described by this rule. The probability of this rule on the test set is 0.71.



For this rule, results of on a test set are shown in Figure 4 and Table 2.

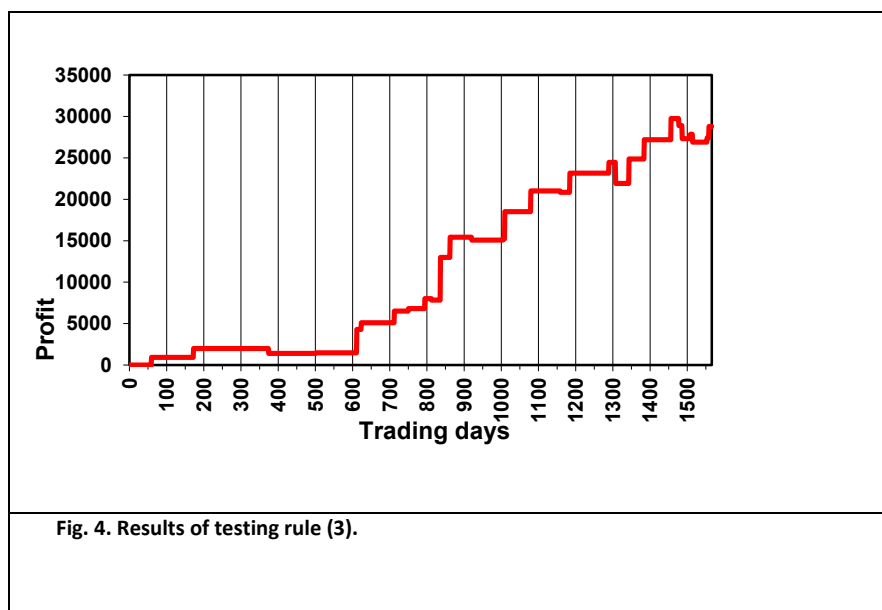


Table 2. Characteristics of rule (3).

Indicator	Value
Profit	28786
Maximum drawdown	-2846
Expected value, see (3)	993
Percentage of profitable trades	72 %
Annual rate of return in relation to the maximum drawdown of the account	236 %

Comparison with other methods. In order to evaluate the efficiency of the Discovery system, we compared this system with the sliding linear regression method and neural networks. To compare the quality of various methods, it is necessary to choose a method for comparing the results of various methods. As applied to financial problems, such a comparison method can be a comparison of the financial performance indicators of the trading systems based on these methods. Obviously, the method, whose forecast allows extracting more profit with less risk, has an advantage.

Sliding linear regression. For an arbitrary function $c(t)$ represented by its values $c(t_k)$ at time moments t_k , a moving linear regression is constructed as a function $y(t) = a_n t + b_n$ using the last n values of the time series. Based on this function, the following value is predicted by the formula $y'(t+1) = a_n(t+1) + b_n$. The trading strategy based on sliding linear regression is defined as follows:

$$Signal(t) = \begin{cases} 1, & \text{if } y(t)' < y(t+1)' \\ -1, & \text{if } y'(t+1) < y'(t) \\ 0, & \text{otherwise} \end{cases}$$

$Signal(t) = 1$ means to open a buy position, if at the same time a sell position is already open, then it must be closed. $Signal(t) = -1$ means to open a sell position, if at the same time a buy position is already open, then it must be closed. $Signal(t)=0$ means to keep open positions. The accuracy of the linear regression forecast largely depends on the choice of the size of the linear regression window. Linear regression was used for

comparison, the window size of which was optimized, in order to obtain the best financial performance indicators of the trading system.

Neural network. For comparison, we used multilayer neural networks with direct connections, trained by back propagation of errors. A greater influence on the quality of forecasts of neural networks is provided by the method of presenting input information [25]. Here, there is a large correlation between successive values - the most probable course value at the next moment is equal to its previous value: $c(t) = c(t-1) + \Delta c(t) \approx c(t-1)$.

At the same time, to improve the quality of training, one should strive for statistical independence of inputs, i.e. to the absence of such correlations. Therefore, the most significant values for prediction are not the values themselves, but their changes $\Delta c(t)$ as the most statistically independent values [25]. Based on these considerations, the initial series $c(t)$ were converted into a series of relative increments $d(t)$, $d(t) = \Delta c(t) / c(t)$.

Here the input features of neural networks are k last values of the series: $d(t), d(t-1), \dots, d(t-k+1)$ and the output value is $(c(t+5) - c(t)) / c(t)$, i.e. neural networks were trained to predict the relative change of c in five days.

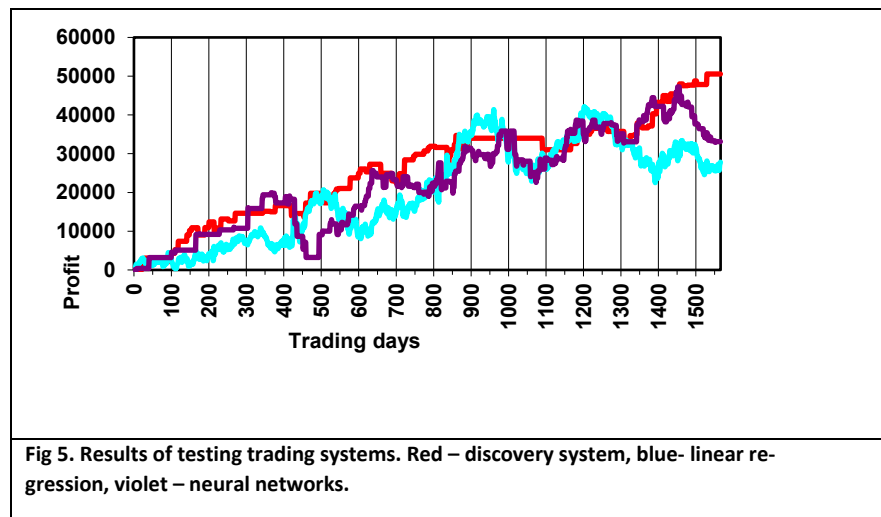
Neural network training contains uncertainty associated with a random selection of initial weights. It leads to instability of neural networks for highly noisy financial time series. To increase the reliability of prediction, it is recommended to use a committee of neural networks [7]. We used a committee of 12 neural networks with different architectures and the following trading strategy:

$$Signal(t) = \begin{cases} 1, & \text{if } Out(t) > 0 \\ -1, & \text{if } Out(t) < 0 \\ 0, & \text{otherwise} \end{cases},$$

where $Out(t)$ is output of the set of neural networks to time t , $Signal(t)$ is a signal to trade on day t . $Signal(t)=1$ means to open a position to buy for 5 days, and if the position is already opened it need to be closed. $Signal(t)=-1$ means to open a position to sell for 5 days, and if the position is already opened it need to be closed. $Signal(t)=0$ means no trading on day

t. Testing of this trading system was also carried out by moving control, with the same parameters.

Summary of the comparison. All methods were tested on the same time interval. Each trading system always entered the market with only one unit of the contract, and did not use stop-loss orders. Table 3 and Figure 5 present the results of comparing the Discovery system with other methods. The graph in Figure 5 shows the dynamics of capital growth over time for all three systems. The table 3 shows that the "Discovery" system surpasses other methods, in the percentage of correct forecasts and the indicators of financial efficiency.



Comparing the profitability charts of trading systems with the closing price chart of S&P500 shows that the trading system based on moving linear regression works well in areas with a noticeable trend. However, in places of a trend change and, especially, in the areas with very slight fluctuations of the price, it shows big losses. This is easily explained by the fact that linear regression at each moment of time tries to approximate the time series by a straight line, and this is acceptable only if there is a strong linear trend in the market.

Table 3. Comparison of methods.

Indicator	Discovery system	Linear regression	Neural networks
Profit	50563	27805	33117
Maximum drawdown	-4960	-19059	-16700
Expected value, see (3)	568	66	106
Percentage of profitable trades	69 %	41 %	57 %
Annual rate of return in relation to the maximum drawdown of the account	238 %	34 %	46 %

Compared to linear regression, the neural network trading system works slightly better in the areas with very slight fluctuations of the price, but a trend change still has a catastrophic effect on the shape of the yield curve, causing large dips. Moreover, the careful analysis of the graphs shows that neural networks work well only in those areas where the overall price dynamics coincides with the price dynamics of the area in which they were trained. This is because neural networks try to approximate the entire training set, trying to minimize the average error, so they first find the most general patterns that are satisfied for most examples from the training set. Thus, neural networks basically “catch” the most typical dynamics of the time series and cease to work when the trend changes.

The relational system, in contrast to neural networks, can find highly probable statistically significant patterns preceding a directional price movement. Most of these patterns continue to work successfully with trend changes.

4.4 Visual knowledge discovery for currency exchange trading

This section presents a visual knowledge discovery approach to find an investment strategy in multidimensional space of financial time series [67]. Visualization based on the lossless Collocated Paired Coordinates (CPC) [43] plays an important role in this investment approach. The dedicated

CPC subspaces constructed for EUR/USD foreign exchange market time series include characteristics of moving averages, differences between moving averages, changes in volume, adjusted moving averages (Bollinger band), etc.

In this study the profit is analyzed in normalized units: price interest points (pip) and PPC (Profit per candle). A pip indicates the change in the exchange rate for a currency pair, where one pip is 0.0001 USD in the pair EURUSD that is used as a measurement unit of change. Profit per candle (PPC) is the difference between the cumulative profit at the end and the start of the period divided by the number of candles in the period,

$$PPC = (\text{Profit}_{\text{end}} - \text{Profit}_{\text{begin}}) / (i_{\text{end}} - i_{\text{begin}}),$$

where i_{end} and i_{begin} are the numbers of considered candles, e.g., one-hour candles.

Effective relations were found for one-hour EURUSD pair in 2-D and 3-D visualization spaces, which lead to a profitable investment decision (long, short position or nothing) that can be used in algotrading mode.

Below we summarize steps of CPC:

- (1) Representing a normalized to $[0,1]$ n-D point $\mathbf{x} = (x_1, x_2, \dots, x_{n-1}, x_n)$, as a set of pairs $(x_1, x_2), \dots, (x_i, x_{i+1}), \dots, (x_{n-1}, x_n)$;
- (2) Drawing 2-D orthogonal Cartesian coordinates $(X_1, X_2), \dots, (X_{n-1}, X_n)$ with all odd coordinates *collocated* on a single horizontal axis X and all even coordinates *collocated* on a single vertical axis Y;
- (3) Drawing each pair (x_i, x_{i+1}) in (X_i, X_{i+1}) ;
- (4) Connecting pairs by arrows to form a graph \mathbf{x}^* :
 $(x_1, x_2) \rightarrow (x_3, x_4) \rightarrow \dots \rightarrow (x_{n-1}, x_n)$.

This graph \mathbf{x}^* represents n-D point \mathbf{x} in 2-D losslessly, i.e., all values of \mathbf{x} can be restored. Thus, this visualization is reversible representing all n-D data without loss of them. For the odd n the last pair can be (x_n, x_n) or $(x_n, 0)$. For 3-D visualization, the pairs are substituted by the triples $(x_1, x_2, x_3), \dots, (x_i, x_{i+1}, x_{i+2}), \dots, (x_{n-2}, x_{n-1}, x_n)$. For an arbitrary n , some coordinates are repeated to get n divisible by 3.

For time series, pairs of variables (x_i, x_{i+1}) can be sequential pairs of values at time t and $t+1$. In this way, a 4-D point \mathbf{x} is formed as $(v_t, y_t, v_{t+1}, y_{t+1})$, where v is volume and y is profit at two consecutive times t and $t+1$. Respectively, a 4-D point $(v_t, y_t, v_{t+1}, y_{t+1})$ is represented in 2-D as an arrow from 2-D point (v_t, y_t) to 2-D point (v_{t+1}, y_{t+1}) . This simple graph fully represents 4-D data. It has a clear and simple meaning, for instance, the arrow going up and to the right indicates the growth in both profit and volume from time t to $t+1$. To observe better the beginnings and ends of events the time pairs starting from odd time t are visualized separately from time pairs starting from even time t .

Similarly, a 6-D point \mathbf{x} can be $(v_t, d_{MA_t}, y_t, d_{MA_{t+1}}, v_{t+1}, y_{t+1})$, where d_{MA} is the difference between the moving averages for some windows. We represent it in 3-D as an arrow from 3-D point (v_t, y_t, d_{MA_t}) to 3-D point $(v_{t+1}, d_{MA_{t+1}}, y_{t+1})$. This simple graph fully represents 6-D data point with a clear meaning — the arrow going up and to the right indicates the growth in all three attributes: profit, d_{MA} , and volume from time t to $t+1$.

To shorten notation for the spaces we will use notation like (Y_r, V_r) instead of $(Y_{rt}, V_{rt}, Y_{r,t+1}, V_{r,t+1})$, and (Y_r, d_{MA_r}, V_r) , instead of $(Y_{rt}, V_{rt}, d_{MA_{rt}}, Y_{r,t+1}, d_{MA_{r,t+1}}, V_{r,t+1})$, where index r stands for normalized profit, volume and d_{MA} . Thus, 2-D and 3-D notations (Y_r, V_r) and (Y_r, d_{MA_r}, V_r) represent 4-D and 6-D spaces, respectively. In figures below, pins represent the arrows in the graphs, where circles indicate the beginnings of the arrows with filled circles for the long position and empty circles for the short positions.

The CPC visualization gives an idea of how to create an investment strategy -- to learn and discover places in 2-D/3-D CPC representation where asymmetry between number of suggestions to open long position (filled circles) and short positions (empty circles) is high. A rectangle or a square in 2D space and a cube or a cuboid in 3D space have been used in the experiments. These areas are changing and need to be learned and updated regularly.

The learning mode includes finding the optimal size rectangles and cuboids. Figure 6 shows the 2 colored cubes, with the largest asymmetry factor found by the learning algorithm. The proper positions can be opened when subsequent events are located in the cubes. It provided the positive

prediction with 0.619 accuracy for long positions, 0.686 for short positions and a threshold $T_{min}=10$ on the number of pin circles required in the cube. The cumulative profit of this strategy for 5000 candles in learning period and 1700 candles during testing period is not too rewarding because of small number of positive events with a large period without trade. It indicates the need for a more dynamic strategy.

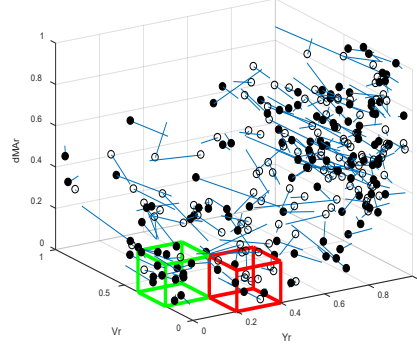


Fig. 6. Two discovered cubes in Y_r - $dMAr$ - V_r space with the maximum asymmetry between long and short positions.

Strategy based on quality of events in the cubes. The next approach uses the sum of returns Y_r accumulated in the learning period. A new hypothesis is that the more vertical arrows (pins) lead to higher profit. Consider a cube indexed by (k_1, k_2, k_3) in a 3D grid, where $k_1, k_2, k_3 = 1, 2, \dots, K$. We are interested in maximum of criterion C_l for long positions and C_s for short positions:

$$C_l(k_1, k_2, k_3) = \sum (Y_{r,i}(k_1, k_2, k_3) - Y_{r,i-1}(k_1, k_2, k_3))$$

when $(Y_{r,i} - Y_{r,i-1}) > 0$ for all i that belong to a learning period and for all cubes (k_1, k_2, k_3) and

$$C_s(k_1, k_2, k_3) = \sum (Y_{r,i}(k_1, k_2, k_3) - Y_{r,i-1}(k_1, k_2, k_3))$$

when $(Y_{r,i} - Y_{r,i-1}) < 0$ for all i that belong to learning period and for all cubes (k_1, k_2, k_3) .

Recall that the beginning of the arrow ($Y_{r,i-1}$) belongs to (k_1, k_2, k_3) -cube belongs, not its head. For each learning period, the sums $C_l(k_1, k_2, k_3)$

and $C_s(k_1, k_2, k_3)$ are computed in every (k_1, k_2, k_3) -cube. A corresponding investment strategy is -- if C_l dominates, then open a long position, else open a short position:

$$C_l(k_1, k_2, k_3) > C_s(k_1, k_2, k_3) + d_c \text{ then open long position,}$$

$$C_s(k_1, k_2, k_3) > C_l(k_1, k_2, k_3) + d_c \text{ then open short position,}$$

where d_c is additional difference for C_l and C_s to make the difference more distinct.

Figure 7a shows the bars that represent the criteria C_l and C_s (by their length) in every cube (in grid $4 \times 4 \times 4$). The bars have different lengths. A visual strategy based on this difference is one of the bars is longer than the other one, then open a proper position. It was checked for different values of periods of learning and testing, i.e., bars have been generated for the test data and compared with bars for the training data.

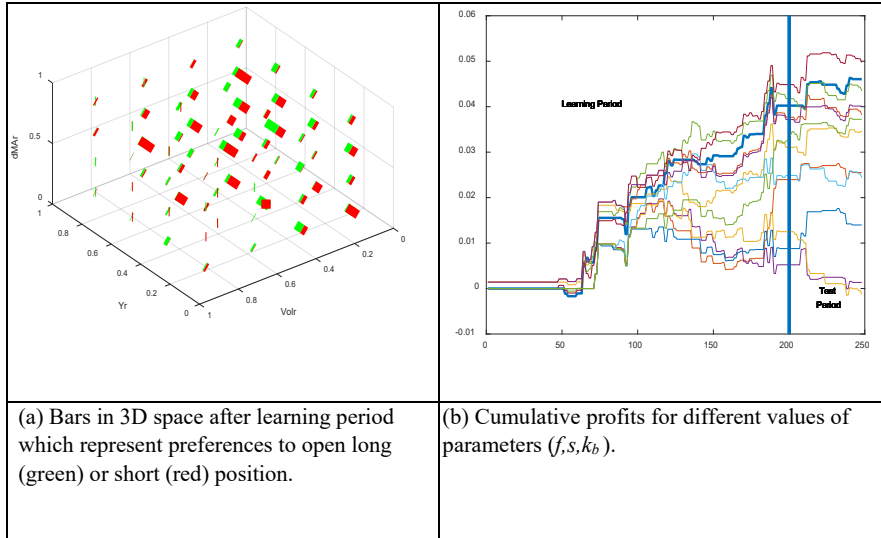


Fig. 7. Bars in 3D after learning period which represent preferences to open long (green) or short (red) position and cumulative profits for different values of model parameters.

The thicker line in Figure 7b is the best one, relative to the value of a linear combination of the cumulative profits, for the learning period, and Calmar ratio in the same learning period. For selecting the promising curve, the following criterion is used:

$$C = w_c \cdot \text{Calmar}_{\text{learn}} + w_p \cdot \text{profit}_{\text{learn}}$$

where w_c is a weight of Calmar component (in these experiments $w_c=0.3$); w_p is a weight of profit component (in these experiments $w_p=100$ for 500 hours of learning period); $Calmar_{learn}$ is the Calmar ratio at the end of learning period; and $profit_{learn}$ is a cumulative profit at the end of the learning period measured as a change in EURUSD rate.

Here Calmar ratio, $CR = (\text{average of return over time } \Delta t) / \text{max drawdown over time } \Delta t$, which shows the quality of trading. In the literature the Calmar ratio of more than 5 is viewed as excellent, 2 – 5 as very good, and 1 – 2 as just good [67].

The curve with a significant profit and a small variance (captured by larger Calmar ratio for evaluating risk) is used in criterion C as a measure of success of the algorithm along with comparison with the result to typical benchmarks.

The main idea in constructing the criterion C is to balance risk and profit contributions with different pairs of weights (w_c, w_p) .

Figure 7b shows that while the thicker line (best in criterion C) provides one of the top profits, it does not provide the best cumulative profits at the end of learning period due to the weighting nature of criterion C that also takes into account the risk in the form of Calmar ratio.

Figures 8 and 9 show cumulative profits for different learning and testing periods as a function of 20 shifted training and testing datasets (cycles). First the best values of parameters (f, s, k_b) were found by optimizing them on training data and then these values were used in the test period which follows the learning period. Figures 8 and 9 show very efficient results for 1 hour and 1 day taking into account the profit-risk relation. Profit is represented by its pips value. These figures show the general direction of changes in cumulative profit, but y-axes have different scales and should not be compared directly. Practitioners consider the profit at the level of 10-20 pips per day as very high or even unrealistic to be stable [67]. The general conclusion is that the chosen space (Y_r, V_r, d_{MAr}) is very efficient.

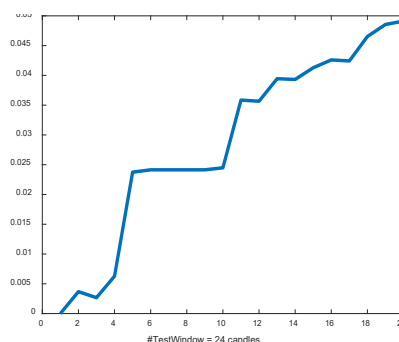


Fig. 8. Cumulative profit for main strategy with learning windows of 100 1h-candles and test windows of 24 1h-candles
(Calmar ratio ~18, PPC=0.87).

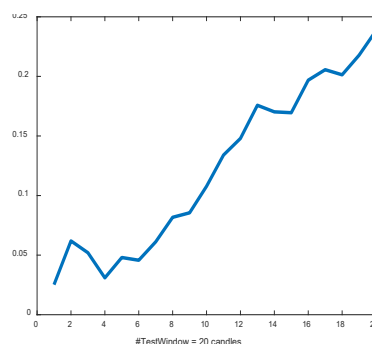


Fig. 9. Cumulative profit for the strategy in 1d EURUSD time series with 5-days testing windows and 30-days learning windows in 20 periods, Calmar ratio=7.69; PPC=23.83 pips; delta=0.05.

The learning period in Figure 9 is about one month and the testing period is a one-week period (five trading days). These are very convenient for automatic and manual trading. For model detail, other experiments and positive comparison with Buy&Hold benchmark see [67]. The CPC concept can be applied more generally for financial time series. Besides CPC, the whole class of General Line Coordinates [43] opens multiple opportunities to represent n-D financial data visually and discovering patterns in these data. See a chapter on explainable machine learning and visual knowledge discovery in this handbook [44]. This section illustrates the potential of an emerging joint area of visual knowledge discovery and investment strategies, to boost the creativity of both scientists and practitioners.

4.5 Discovering money laundering and attribute-based relational machine learning

Problem statement. Forensic accounting is a field that deals with possible illegal and fraudulent financial transactions. One of the tasks in this field is the analysis of funding mechanisms for terrorism (Prentice, 2002) where *clean money* (e.g., *charity money*) and *laundered money* are both *used* for a variety of activities including acquisition and production of weapons and their precursors. In contrast, traditional illegal businesses and drug trafficking *make dirty money appear clean*.

The specific tasks in automated forensic accounting related to machine learning are the identification of suspicious and unusual electronic transactions and the reduction in the number of 'false positive' suspicious transactions. Inexpensive, simple rule-based systems, customer profiling, statistical techniques, neural networks, fuzzy logic and genetic algorithms are considered as appropriate tools (Prentice, 2002).

There are many indicators of possible suspicious (abnormal) transactions in traditional illegal business. These include: (1) the use of several related and/or unrelated accounts before money is moved offshore, (2) a lack of account holder concern with commissions and fees, (3) correspondent banking transactions to offshore shell banks, (4) transferor insolvency after the transfer or insolvency at the time of transfer, (5) wire transfers to new places [15], (6) transactions without identifiable business purposes, and (7) transfers for less than reasonably equivalent value.

Some of these indicators can be easily implemented as simple flags in software. However, indicators such as wire transfers to new places produce a large number of 'false positive' suspicious transactions. Thus, the goal is to develop more sophisticated mechanisms, based on interrelations of many indicators.

Machine learning can assist in discovering patterns of fraudulent activities that are closely related to terrorism, such as transactions without identifiable business purposes. The problem is that often an individual transaction does not reveal that it has no identifiable business purpose, or that it was done for no reasonably equivalent value. Thus, machine-learning techniques can search for suspicious patterns in the form of more complex combinations of transactions and other evidence using background knowledge. This means that the training data are formed not by transactions themselves but combination of two, three or more transactions. This implies that the number of training objects exploded. The percentage of suspicion records in the set of all transactions is very small, but the percentage of suspicious combinations in the set of combinations is minuscule. This is a typical task of *discovering rare patterns*. Traditional machine learning methods and approaches are ill equipped to deal this such problems. Relational machine learning methods open new opportunities for solving these tasks by discovering "negated patterns" described below based on [42].

Approach and method. Consider a transactions dataset with attributes such as seller, buyer, item sold, item type, amount, cost, date, company name, type, company type. We will denote each record in this dataset as $\langle S \rangle, \langle B \rangle, \langle I \rangle$, where $\langle S \rangle, \langle B \rangle$, and $\langle I \rangle$ are sets of attributes about the seller, buyer, and item, respectively. We may have two linked records $R_1 = (\langle S_1 \rangle, \langle B_1 \rangle, \langle I_1 \rangle)$ and $R_2 = (\langle S_2 \rangle, \langle B_2 \rangle, \langle I_2 \rangle)$, such that the first buyer B_1 is also a seller S_2 , $B_1 = S_2$. It is also possible that the item sold in both records is the same $I_1 = I_2$. We create a new dataset of pairs of linked records $\{\langle R_1, R_2 \rangle\}$. ML methods will work in this dataset to discover suspicious records if samples or definitions of normal and suspicious patterns provided. Below we list such patterns:

- a normal pattern (NP) – a Manufacturer Buys a Precursor & Sells the Result of manufacturing (MBPSR);
- a suspicious (abnormal) pattern (SP) – a Manufacturer Buys a Precursor & Sells the same Precursor (MBPSP);
- suspicious pattern (SP) – a Trading Co. Buys a Precursor and Sells the same Precursor Cheaper (TBPSPC);
- a normal pattern (NP) – a Conglomerate Buys a Precursor & Sells the Result of manufacturing (CBPSR).

A machine learning algorithm A analyzes pairs of records $\{\langle R_1, R_2 \rangle\}$ with say 18 attributes total and can match a pair (#5,#6) with a normal pattern MBPSR, $A(\#5,\#6) = \text{MBPSR}$, while another pair (#1,#3) can be matched with a suspicious pattern, $A(\#1,\#3) = \text{MBPSP}$.

If the definitions of suspicious patterns are given, then finding suspicious records is a matter of a computationally efficient search in a database, which can be distributed. This is not the major challenge. The *automatic generation of patterns/hypotheses descriptions* is a major challenge. One can ask: “Why do we need to discover these definitions (rules) automatically?” A manual way can work if the number of types of suspicious patterns is small, and an expert is available. For multistage money-laundering transactions, this is difficult to accomplish manually. Creative criminals and terrorists permanently invent new and more sophisticated money laundering schemes. There are no statistics for such new schemes, to learn as it is done in traditional machine learning approaches. An approach based on the idea of “*negated patterns*” can uncover such

unique schemes. According to this approach, *highly probable patterns* are discovered and then *negated*. It is assumed that a highly probable pattern should be *normal*. More formally, the *main hypothesis (MH)* of this approach is:

If Q is a highly probable pattern (>0.9) then Q constitutes a normal pattern and $\text{not}(Q)$ can constitute a suspicious (abnormal) pattern

Below we outline an algorithm, based on this hypothesis, to find suspicious patterns. Computational experiments with two synthesized databases and few suspicious transactions schemes permitted us to discover transactions. The actual relational machine-learning algorithm used was algorithm MMRD (Machine Method for Discovery Regularities). Previous research has shown that MMDR based on first-order logic and probabilistic semantic inference is computationally efficient and complete for statistically significant patterns (Kovalerchuk and Vityaev, 2000).

The algorithm finding suspicious patterns based on the main hypothesis (MH) consists of four steps:

1. *Discover* patterns, compute probability of each pattern, select patterns with probabilities above a threshold, say 0.9. To be able to compute conditional probabilities patterns should have a rule form: IF A then B. Such patterns can be extracted using decision tree methods for relatively simple rules and using relational machine learning for discovering more complex rules. Neural Network (NN) and regression methods typically have no if-part. With additional effort, rules can be extracted from the NN and regression equations.
2. *Negate* patterns and compute *probability* of each negated pattern,
3. Find records database that satisfy negated patterns and analyze these records for possible *false alarm* (records maybe normal not suspicious).
4. Remove false alarm records and provide detailed analysis of suspicious records.

The details can be found in [42]

5. CONCLUSION

To be successful a machine learning project should be driven by the application needs and results should be tested quickly. Financial applications provide a unique environment where efficiency of the methods can be tested instantly, by not only using traditional training and testing data, but making real stock forecast and testing it the same day or week. This process can be repeated daily for several months collecting quality estimates. This chapter highlighted problems of ML in finance and specific requirements for the ML methods including in making interpretations, incorporating relations, and probabilistic learning.

The relational ML outlined in this chapter advances pattern discovery methods that deal with complex numeric and non-numeric data, involve structured objects, text and data in a variety of discrete and continuous scales (nominal, order, absolute and so on). The chapter shows benefits of using such *interpretable* methods for stock market forecast and forensic accounting, which includes uncovering money laundering schemes. The technique combines first-order logic and probabilistic semantic inference. The approach has been illustrated with examples of discovery of suspicious patterns in forensic accounting, and stock market trading.

The success of machine learning exercises on automated trading systems has been reported in literature extensively., e.g., [33, 67]. For instance, machine-learning methods achieved better performance than traditional statistical methods in predicting bankruptcy and credit ratings, e.g., [5, 34].

The next direction is developing decision support software tools specific for financial tasks to adjust efficiently the financial ML models to a new data stream.

In the field of ML in finance, we expect an extensive growth of *hybrid methods* that combine different analytical and visual models [67, 43] to provide a better performance than individual methods. In the integrative approach, individual models can serve as *trained artificial “experts”*. Therefore, their combinations can be organized similar to a consultation of real *human experts*. Moreover, these artificial experts can be combined with real experts. These artificial experts can be built as autonomous *intelligent software agents*. Thus, “experts” to be combined can be machine learning models, real financial experts, trader and virtual *experts* (software

intelligent agents) that runs trading rules extracted from real experts. This requires “*expert mining*” for extracting knowledge from human experts to enhance virtual experts [41].

We expect that ML in finance will be shaped as a distinct field, which blends knowledge from finance and machine learning, similar to what we see now in bioinformatics where integration of field specifics and machine learning is close to maturity. We also expect that the blending with ideas from the theory of dynamic systems, chaos theory, and physics of finance will deepen.

References

- [1] Alexander L, Das SR, Ives Z, Jagadish HV, Monteleoni C. Research challenges in financial data modeling and analysis. *Big data*. 2017 Sep 1;5(3):177-88.
- [2] Azoff, E., *Neural networks time series forecasting of financial markets*, Wiley, 1994.
- [3] Back, A., Weigend, A., A first application of independent component analysis to extracting structure from stock returns. *Int. J. on Neural Systems*, 8(4):473– 484, 1998.
- [4] Balaji, A. Ram D., Nair B., Applicability of Deep Learning Models for Stock Price Forecasting An Empirical Study on BANKEX Data, *Procedia Computer Science*, Volume 143, 2018, Pages 947-953.
- [5] Barboza F, Kimura H, Altman E. Machine learning models and bankruptcy prediction. *Expert Systems with Applications*. 2017 Oct 15;83:405-17.
- [6] Baumann C, Elliott G, Burton S. Modeling customer satisfaction and loyalty: survey data versus data mining. *Journal of services marketing*. 2012 May 18;26(3):148-57.
- [7] Baxt W. G. Improving the accuracy of an artificial neural network using multiple differently trained networks, *Neural Computation*, 4(5):, 772-780. 1992
- [8] Becerra-Fernandez, I., Zanakakis, S. Walczak, S., Knowledge discovery techniques for predicting country investment risk, *Computers and Industrial Engineering* Vol. 43 , Issue 4:787 – 800, 2002.
- [9] Bojadziev G. *Fuzzy logic for business, finance, and management*. World Scientific; 2007.
- [10] Bormetti G, Callegaro G, Livieri G, Pallavicini A. A backward Monte Carlo approach to exotic option pricing. *European J. of Applied Mathematics*. 2018, 29(1):146-87.
- [11] Bouchaud, J., Potters, M., *Theory of Financial Risks: From Statistical Physics to Risk Management*, 2000, Cambridge Univ. Press, Cambridge, UK.

- [12] Bratko, I., Muggleton, S., Applications of Inductive Logic Programming. Communications of ACM, 38(11): 65-70, 1995.
- [13] Casdagli, M., Eubank S., (Eds). Nonlinear modeling and forecasting, Addison Wesley, 1992.
- [14] Cao J, Lian G, Roslan TR. Pricing variance swaps under stochastic volatility and stochastic interest rate. Applied Mathematics and Computation. 2016;277:72-81.
- [15] Chabrow, E. Tracking the terrorists, Information week, Jan. 14, 2002.
- [16] Chen Y, He K, Tso GK. Forecasting crude oil prices: a deep learning based model. Procedia computer science. 2017, vol. 1;122:300-307.
- [17] Gonçalves R, Ribeiro VM, Pereira FL, Rocha AP. Deep Learning in Exchange Markets. Information Economics and Policy, Vol. 47, 2019, 38-51
- [18] Cowan, A., Book review: Data Mining in Finance, International journal of forecasting, Vol.18, Issue 1, 155-156, Jan-March 2002.
- [19] De Raedt L. Logical and relational learning. Springer, 2008.
- [20] De Raedt L., Kersting K, Natarajan S, Poole D. Statistical relational artificial intelligence: Logic, probability, and computation. Synthesis Lectures on Artificial Intelligence and Machine Learning. 2016 Mar 24;10(2):1-89.
- [21] Dhar, V., Stein,R., Intelligent decision support methods, Prentice Hall, 1997.
- [22] Došilović FK, Brčić M, Hlupić N. Explainable artificial intelligence: A survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO) 2018 (pp. 0210-0215). IEEE.
- [23] Drake, K., Kim Y., Abductive information modeling applied to financial time series forecasting, In: Nonlinear financial forecasting, Finance and Technology, 1997, 95-109.
- [24] Dzeroski S. , Relational data mining, in: Data mining and knowledge discovery Handbook. 2nd ed., O.Maimon, L. Rokach (Eds), pp. 887-911, Springer, 2010.
- [25] Ezhov, A., Shumski, S., Neurocomputing and its applications in economics and business, MFTI, Moscow,1998 (in Russian).
- [26] Fang B, Zhang P. Big data in finance. InBig data concepts, theories, and applications 2016, 391-412, Springer.
- [27] Freedman R., Klein R., Lederman J., Artificial intelligence in the capital markets, Irwin, Chicago, 1995.
- [28] Giles, G., Lawrence S., Tshoi, A. Rule inference for financial prediction using recurrent neural networks, In: Proc. of IEEE/IAAFE Conference on Computational Intelligence for financial Engineering, IEEE, NJ, 1997, 253-259.
- [29] Goertzel B, Iklé M, Goertzel IF, Heljakka A. Probabilistic logic networks: A comprehensive framework for uncertain inference. Springer, 2008.

- [30] Greenstone, M., Oyer, P., Are There Sectoral Anomalies Too? The Pitfalls of Unreported Multiple Hypothesis Testing and a Simple Solution, *Review of Quantitative Finance and Accounting*, 15, 2000: 37-55.
- [31] Gulwani S, Hernández-Orallo J, Kitzelmann E, Muggleton SH, Schmid U, Zorn B. Inductive programming meets the real world. *Communications of the ACM*. 2015;58(11):90-9.
- [32] Hajizadeh E, Ardakani HD, Shahrabi J. Application of data mining techniques in stock markets: A survey. *Journal of Economics and International Finance*. 2010 Jul 31;2(7):109-18.
- [33] Huang B, Huan Y, Xu LD, Zheng L, Zou Z. Automated trading systems statistical and machine learning methods and hardware implementation: a survey. *Enterprise Information Systems*. 2019 Jan 2;13(1):132-44.
- [34] Huang Z, Chen H, Hsu CJ, Chen WH, Wu S. Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision support systems*. 2004;37(4):543-58.
- [35] Ilinski, K., *Physics of Finance: Gauge Modeling in Non-Equilibrium Pricing*, Wiley, 2001
- [36] Jadhav I., He H., Jenkins K., an academic review: applications of data mining techniques in finance industry, *International Journal of Soft Computing and Artificial Intelligence*, ISSN: 2321-404X, Volume-4, Issue-1, May-2016
- [37] Jeong G., Kim H., Improving financial trading decisions using deep Q-learning: Predicting the number of shares, action strategies, and transfer learning, *Expert Systems with Applications*, Vol. 117, 2019, 125-138.
- [38] Kingdon, J., *Intelligent systems and financial forecasting*. Springer, 1997.
- [39] Koller D, Friedman N, Džeroski S, Sutton C, McCallum A, Pfeffer A, Abbeel P, Wong MF, Heckerman D, Meek C, Neville J. *Introduction to statistical relational learning*. MIT press; 2007.
- [40] Kovalerchuk, B., Vityaev, E., *Data Mining in Finance: Advances in Relational and Hybrid Methods*, Kluwer, 2000.
- [41] Kovalerchuk, B., Vityaev E., Ruiz J.F., Consistent and Complete Data and "Expert Mining" in Medicine, In: *Medical Data Mining and Knowledge Discovery*, Springer, 2001, 238-280.
- [42] Kovalerchuk B, Vityaev E, Holtfreter R. Correlation of complex evidence in forensic accounting using data mining. *Journal of Forensic Accounting*. 2007;8(1).

Kovalerchuk B., Vityaev E., *Symbolic Methodology for Numeric Data Mining. Intelligent Data Analysis*. v.12(2), IOS Press, 2008, pp.165-188.
- [43] Kovalerchuk, B. *Visual Knowledge Discovery and Machine Learning*, 2018, Springer.
- [44] Kovalerchuk, B., *Explainable Machine Learning Boosted by Visual Means*, In: *Handbook of Machine Learning for Data Science (this volume)*, 2020, Springer.

- [45] .Lavrač N, Vavpetič A. Relational and semantic data mining. In: International Conference on Logic Programming and Nonmonotonic Reasoning 2015, 20-31, Springer.
- [46] Mantegna, R., Stanley, H., An Introduction to Econophysics: Correlations and Complexity in Finance, Cambridge Univ. Press, Cambridge, UK, 2000
- [47] Mehta, K., Bhattacharyya S., Adequacy of Training Data for Evolutionary Mining of Trading Rules, Decision support systems, 37(4):461-474, 2004
- [48] Mitchell, T., Machine learning. 1997, McGraw Hill.
- [49] Muller, K.-R., Smola, A., Rtsch, G., Schlkopf, B., Kohlmorgen, J., & Vapnik, V., 1997. Using support vector machines for time series prediction, In: Advances in Kernel Methods – Support Vector Learning, MIT Press, 1997.
- [50] Muggleton S., Scientific Knowledge Discovery Using Inductive Logic Programming. Communications of ACM, 42(11), 1999, 42-46.
- [51] Muggleton SH, Lin D, Tamaddoni-Nezhad A. Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. Machine Learning. 2015 Jul 1;100(1):49-73.
- [52] Muggleton SH, Schmid U, Zeller C, Tamaddoni-Nezhad A, Besold T. Ultra-Strong Machine Learning: comprehensibility of programs learned with ILP. Machine Learning. 2018 Jul 1;107(7):1119-40.
- [53] Muggleton, S., Learning Structure and Parameters of Stochastic Logic Programs, 12th International Conference, ILP 2002, Sydney, Australia, July 9-11, 2002. Lecture Notes in Computer Science 2583 Springer 2003, 198-206.
- [54] Murphy JJ. Technical analysis of the financial markets: A comprehensive guide to trading methods and applications. Penguin; 1999.
- [55] Saxena A, Sharma N, Saxena K, Parikh SM. Financial Data Mining: Appropriate Selection of Tools, Techniques and Algorithms. In: Intern. Conf. on Smart Trends for Information Technology and Computer Communications 2017, 244-251. Springer.
- [56] Selvin S, Vinayakumar R, Gopalakrishnan EA, Menon VK, Soman KP. Stock price prediction using LSTM, RNN and CNN-sliding window model. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2017, 1643-1647, IEEE
- [57] Sirignano J, Cont R. Universal features of price formation in financial markets: perspectives from deep learning. Quantitative Finance. 2019 Sep 2;19(9):1449-59.
- [58] Sohagrir S, Wang D, Pomeranets A, Khoshgoftaar TM. Big Data: Deep Learning for financial sentiment analysis. Journal of Big Data. 2018 Dec;5(1):3.
- [59] Vargas MR, De Lima BS, Evsukoff AG. Deep learning for stock market prediction from financial news articles. In: 2017 IEEE Intern. Conf. on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, 2017, 60-65, IEEE.

- [60] Vityaev E. The logic of prediction. In: Mathematical logic in Asia 2006, 263-276, World Scientific Publ.
- [61] E. Vityaev, S. Smerdov. On the Problem of Prediction, K.E. Wolff et al. (Eds.): KONT/KPP 2007, LNAI 6581, 2011, 280-296, Springer.
- [62] Vityaev, E., Odintsov, S. How to predict consistently? In: Trends in Mathematics and Computational Intelligence. M. Cornejo (ed), 2019, 35-41, Springer.
- [63] E. Vityaev, B. Kovalerchuk. Ontological Data Mining. Uncertainty Modeling, Studies in Computational Intelligence 683, V. Kreinovich (ed.), Springer, 2017, pp. 277-292.
- [64] Vityaev E., Kovalerchuk B., Relational Methodology for Data Mining and Knowledge Discovery. Intelligent Data Analysis, v.12(2), 2008, 189-210, IOS Press.
- [65] Vityaev E., Kovalerchuk B., Empirical Theories Discovery based on the Measurement Theory. *Mind and Machine*, v.14, #4, 551-573, 2004.
- [66] Vityaev E., Perlovsky L., Kovalerchuk B., Speransky S., Probabilistic dynamic logic of cognition. Biologically Inspired Cognitive Architectures. 2013, 1;6:159-168.
- [67] Wilinski A, Kovalerchuk B. Visual knowledge discovery and machine learning for investment strategy. Cognitive Systems Research. 2017, 1;44:100-114.