Leveraging the explainability of associative classifiers to support quantitative stock trading

Leveraging the Explainability of Associative Classifiers to Support Quantitative Stock Trading

Giuseppe Attanasio
Politecnico di Torino
Torino, Italy
giuseppe.attanasio@polito.it

Luca Cagliero
Politecnico di Torino
Torino, Italy
luca.cagliero@polito.it

Elena Baralis
Politecnico di Torino
Torino, Italy
elena.baralis@polito.it

ABSTRACT

Forecasting the stock market is particularly challenging due to the presence of a variety of inter-related economic and political factors. In recent years, the application of Machine Learning algorithms in quantitative stock trading systems has become established, as it enables a data-driven approach to investing in the financial markets. However, most professional traders still look for an explanation of automatically generated signals to verify their adherence to technical and fundamental rules.

This paper presents an explainable approach to stock trading. It investigates the use of classification rules, which represent reliable associations between a set of discrete indicator values and the target class, to address next-day stock price prediction. Adopting associative classifiers in short-term stock trading not only provides reliable signals but also allows domain experts to understand the rationale behind signal generation.

The backtesting of a state-of-the-art associative classifier, relying on a lazy pruning strategy, has shown promising performance in terms of equity appreciation and robustness of the trading system to market drawdowns.

CCS CONCEPTS

• Computing methodologies → Machine learning algorithms; **Classification and regression trees**; **• Information systems → Decision support systems**.

KEYWORDS

Explainable AI, Stock Price Forecasting, Associative Classification, Quantitative Trading

1 INTRODUCTION

Predicting the stock markets is one of the most challenging financial problems. Stock markets are extremely complex and dynamic systems, as they are influenced by a large number of factors, among which government policies, economic rules, asset evolution, stakeholder actions, and funds' operations [7]. Traders and investors are particularly interested in forecasting the upcoming stock trends in order to drive their actions. Stock trading can be *discretionary* if a domain expert is primarily in charge of making decisions, or *quantitative* if an automated trading system is set up and appointed to invest in the financial markets.

The evolution of Artificial Intelligence techniques has pushed the limits of quantitative trading systems in terms of performance to those previously unseen. Several research studies have focused on training multi-class classification models on stock-related data to forecast next-day stock directions (e.g., uptrend, downtrend, no variation) [5]. These models are often multivariate, i.e., they incorporate several variables taken from (i) technical analysis [8], (ii) financial and economic indicators from financial reports and rating assessments, (iii) sentiment evaluations conducted on online news and social media [4]. A recent survey of existing approaches is given in [12].

Despite their increasing capabilities, Machine Learning-based trading systems are not fully trusted yet. They may suffer from market instability and could generate an excessive number of signals. Hence, traders are more and more often interested in adopting *Explainable AI models*, i.e., prediction models that are implicitly interpretable by domain experts either *locally* (i.e., explaining why a specific prediction is made) or *globally* (i.e., explaining the rationale behind the predictive model as a whole). Some steps towards this direction have already been made. For example, the trading systems proposed in [2, 11, 13] produce interpretable stock price predictors in the form of neuro-fuzzy rules. Neuro-fuzzy models combine the learning capability of neural networks with the representational power of fuzzy inference systems, thus producing systems that can acquire and represent knowledge from data. Unfortunately, the interpretability of fuzzy knowledge acquired by a neuro-fuzzy system may be heavily compromised by the learning phase of the network. Furthermore, the quality of these rules is known to be sensitive to the initial neural network architecture, the membership function, and the weight setting [9]. In [15] the authors have tried to support news trading experts. They explain the correlations between financial news and stock prices by proposing a dual-layer attention-based neural network, which allocates different weights to different days in terms of their contribution to stock price movements. The solution is again dependent on the learning structure of the neural network.

This paper presents a new explainable approach to stock trading based on associative classification. Associative classifiers discover and exploit classification rules to accomplish the classification task [6]. Classification rules are association rules [1] representing implications in the form $A \rightarrow c$, where, in our context, A is a set of values taken by a subset of categorical descriptors summarizing stock-related information at a daily granularity, whereas the target class indicates the next-day stock price direction. Associative classifiers extract, filter, and rank classification rules denoting strongly reliable correlations between stock descriptors and the prediction target. Since the rules used to generate trading signals are easily interpretable, they can be manually explored, validated, and updated by domain experts. To drive stock trading, they are applied to time-lagged data in order to produce daily trading recommendations. A local explanation of each trading signal produced by an associative model is inherently provided by the rules that are actually applied to forecast the next-day stock direction.

To leverage the explainability of associative classifiers to stock trading, we propose a trading system integrating a state-of-the-art associative algorithm [3] and backtest it on 11-years U.S. market data. The results of the trading simulations show that the associative classifier on average yields stronger returns and more limited drawdown than traditional models, both interpretable and not.

The paper is organized as follows. Section 2 describes the trading system based on associative classification. Section 3 evaluates system performance on historical data. Section 4 draws conclusions and discusses future works.

2 STOCK TRADING BASED ON ASSOCIATIVE CLASSIFICATION

The proposed methodology focuses on applying associative classification in stock trading. Specifically, associative models are first used to forecast next-day stock directions. Then, the rules included in the model are selectively applied as trading signals of a stock trading system. Notably, associative rules are inherently interpretable. Thus, their content can be manually explored and validated by domain experts.

2.1 Data preparation

Historical stock-related data are modelled as relational datasets. The dataset is associated with a given stock and consists of a set of records. Each record corresponds to a distinct trading day and consists of a set of items. *Items* are pairs (descriptor, value), where *descriptor* is the description of a relevant stock-related feature, whereas *value* is the taken value. Hereafter, we will consider the set of descriptors enumerated in Table 1. The aforesaid features describe various properties of the stock-related series, which are commonly used in technical analysis [8]. Notice that, since in this preliminary study we have focused on daily stock price series, we have decided not to include neither descriptors related to stock sentiment (extracted, for instance, from news articles or social news) nor financial or economic indicators (e.g., EBITDA, net income). In fact, the former class of descriptors is mostly useful for driving intra-day trades, whereas the latter is mostly used for planning long-term investment strategy [12]. However, the proposed methodology can be tailored to stock data with arbitrary granularity levels and including an arbitrary set of features. Descriptor values are discretized according to domain knowledge provided by technical analysts [8]. A synthetic description of the semantic aggregations used in th performed experiments is given in Table 1.

2.2 Prediction of next-day stock directions

Forecasting the stock market entails predicting future stock trends. The forecast target is the daily stock price variation, which can be modeled as a discrete class label: (i) *Up*: uptrend, assigned when the variation between the closing price on the next day (t+1) and the closing price of the current day (t) is above 1%. (ii) *Down*: downtrend, assigned when the variation between the closing stock price on the next day (t+1) and the closing price of the current day (t) is below -1%. (iii) *Hold*: stationary trend, assigned when the variation between the closing stock price on the next day (t+1) with respect to the current day (t) is between -1% and 1%.

Classification entails generating a predictive model from a set of training records for which the class value is known and then applying it to a set of (unlabeled) test records. For each stock, we build a separate training dataset including past stock-related data, i.e., records corresponding to past trading days for which the next-day stock direction is known. On day t the per-stock models are used to predict stocks directions on the next day t+1.

2.3 Associative classifiers

Associative classifiers represent scalable yet accurate models based on association rules [1]. They have been first proposed by [6].

Association rules represent reliable correlations between two sets of descriptors values. Let $\mathcal{R}$ be a relational dataset. An *itemset* I is a set of items, i.e., a set of pairs (descriptor, value) related to distinct descriptors. An *association rule* is represented in the form $AR : X \rightarrow Y$, where X and Y are itemsets including distinct descriptors (also denoted as *rule body* and *head*, respectively).

Association rule extraction is driven by two main quality indices: support and confidence [1]. The support of the rule AR, denoted as $sup(AR)$, indicates the frequency of occurrence in $\mathcal{R}$ of the implication in the source dataset, i.e., the number of records in the dataset including both the descriptor values in X and Y. The rule confidence $conf(AR)$ is the conditional probability of occurrence in $\mathcal{R}$ of itemset Y given itemset X, i.e., $conf(AR) = \frac{sup(X \cup Y)}{sup(X)}$.

Rule mining is constrained by a minimum support threshold ms, to consider only the most frequent associations, and by a minimum confidence threshold mc, to filter out unreliable implications.

To tackle the classification problem using association rules, in [6] the authors propose to focus on a particular kind of rules, namely the *classification rules*.. A classification rule is an association rule whose head is a class label (see Section 2.2).

Once extracted, classification rules are filtered and ranked to build the classification model. The idea behind is to shortlist high-quality rules and use them to automatically assign the class labels. In this preliminary work we have considered a state-of-the-art rule filtering and ranking strategy, first proposed by [3].

In [3], rule reliability is validated on training data. Specifically, for each training record, the top-ranked rule matching the record is used to predict the class label (assumed to be unknown). According to validation outcomes, rules are classified as (i) *Level-I rules*: rules that correctly classify at least one training record. (ii) *Level-II rules*: rules that have not classified any record (neither correctly nor incorrectly). (iii) *Harmful rules*: rules that only wrongly classify training records. The classification model consists of the ranked lists of level-I rules (considered first) and level-II rules.

2.4 Rule exploration

Classification rules are inherently interpretable. Thus, they can be easily explored by domain experts to verify their adherence to domain knowledge coming from technical or fundamental analyses. If need be, the data-driven models can be easily updated on-the-fly, e.g., by removing rules that are deemed as misleading.

Table 2 shows four examples of level-I rules extracted from a real stock dataset (related to the U.S. Standard&Poor 500 index).

Table 1: Stock descriptors and the corresponding discretization ranges.

Feature	Description	Discretization range
SMA5-20	Relative difference between SMA(5) and SMA(20)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
SMA8-15	Relative difference between SMA(8) and SMA(15)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
SMA20-50	Relative difference between SMA(20) and SMA(50)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
EMA5-20	Relative difference between EMA(5) and EMA(20)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
EMA8-15	Relative difference between EMA(8) and EMA(15)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
EMA20-50	Relative difference between EMA(20) and EMA(50)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
MACD	Moving Average Convergence/Divergence	(-inf, -5], (-5, -2], (-2, 0], (0, 2], (2, 5], (5, +inf)
AO14	Aroon Oscillator (14 periods)	[-100, -50], (-50, 0], (0, 50], (50, 100]
ADX14	Average Directional Index (14 periods)	(-inf, 20], (20, 25], (25, 40], (40, +inf)
WD14	Difference between Positive Directional Index (DI+) and Negative Directional Index (DI-) (14 periods)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
PPO12-26	Percentage Price Oscillator (12 and 26 periods)	(-inf, -5], (-5, 0], (0, 5], (5, +inf)
RSI14	Relative Strength Index (14 periods)	[0, 15], (15, 30], (30, 50], (50, 70], (70, 85], (85, 100]
MFI14	Money Flow Index (14 periods)	[0, 15], (15, 30], (30, 50], (50, 70], (70, 85], (85, 100]
TSI	True Strength Index	(-inf, -25], (-25, 0], (0, 25], (25, +inf)
SO14	Stochastic Oscillator (14 periods)	[0, 10], (10, 20], (20, 50], (50, 80], (80, 90], (90, 100]
CMO14	Chande Momentum Oscillator (14 periods)	[-100, -75], (-75, -50], (-50, 0], (0, 50], (50, 75], (75, 100]
ATRP14	Average True Range Percentage: ratio, in percentage, between Average True Range and the closing price (14 periods)	[0, 10], (10, 30], (30, 40], (40, 100]
PVO12-26	Percentage Volume Oscillator (14 and 26 periods)	[-100, -40], (-40, -20], (-20, 0], (0, 20], (20, 40], (40, 100]
ADL	Accumulation Distribution Line	(-inf, -1e9], (-1e9, 0], (0, 1e9], (1e9, +inf)
OBV	On Balance Volume	(-inf, -1e9], (-1e9, 0], (0, 1e9], (1e9, +inf)
FI13	Force Index (13 periods)	(-inf, -1e7], (-1e7, 0], (0, 1e7], (1e7, +inf)
FI50	Force Index (50 periods)	(-inf, -1e7], (-1e7, 0], (0, 1e7], (1e7, +inf)

Algorithm 1 Trading strategy

```
1:  procedure TRADE(stocks, initialEquity, stopLoss, maxOpPerDay, maxOpTot)
2:      equity ← initialEquity
3:      positions ← []
4:      for all days do
5:          positions ← close(positions, stocks.Predictions, stopLoss)
6:          profitableStocks ← filterStocks(stocks.Predictions)
7:          sortedStocks ← sort(profitableStocks, stocks.Volume)
8:          positions ← open(sortedStocks, maxOpPerDay, maxOpTot)
9:          equity ← updateEquity(positions)
10:     end for
11: end procedure
```

According to their confidence level, all these rules are deemed as strongly reliable (confidence=100%). For example, the rule related to the AMZN stock highlights a decreasing trend, which is supported by the relative difference between the moving average values (indicators SMA and EMA) and by an increase of the average exchanging volume (according to the ADL indicator). The class indicates *Down*, adhering the fundamentals of technical analysis [8]. Conversely, EBAY has apparently an increasing trend according to the moving averages, but the average exchanging volume is rapidly increasing. Despite the rule recommends *Up*, the decreasing volume trend seems to be contrasting (indicator PVO12-26). In the latter case, thanks to model explainability, domain experts could take appropriate decisions, eventually ignoring the automatic recommendation or updating the rule if they do not trust it.

2.5 Trading strategy and portfolio management

The per-stock predictions of the next-day price direction are deemed as potential short-term trading signals: an *Up* prediction could trigger the opening of a long-selling position, a *Down* prediction could trigger the opening of a short-selling position, whereas a *Hold* could result in a hint of keeping a multi-day position open.

The trading strategy (summarized by Algorithm 1) operates on the stocks deemed as most profitable according to the per-stock daily predictions (see line 7). Money management is based on a fixed percentage strategy. Opening multiple positions over the same stock at the same time is forbidden. To limit the impact of transaction fees on the equity (hereafter approximated to 0.15% in the trading simulations), an overall and a per-day maximum number of open positions are enforced. Among the candidate trading signals, the trading strategy favors those coherent with the per-stock volume trend. (i.e., the top picks defined at line 8). Specifically, the system sorts the stocks by decreasing value of average exchange volume in the last 5 days and opens as much operations picking the signals from the top of the ranking (see line 8).

A short-term position (either long- or short-selling) is opened at closing time and is kept open for at most three trading days. To preserve the equity, at the closing time of each intermediate day each open position is reconsidered and closed in the following cases: (i) A stop loss level (1%, in our experiments) was reached in the current day. (ii) The predicted direction for the same stock on the next day is *Up* for short-selling positions or *Down* for long-selling positions.

3 EXPERIMENTAL RESULTS

We extensively explored the performance of the proposed stock trading system on historical data acquired in different years and market conditions. In the trading simulations we analyzed, separately for each period, the temporal variations of the equity (assuming an initial equity of 100,000 USD), the maximum return and drawdown, the number of opened positions, the average return per position, and the number of positions closed at stop loss level.

To accomplish the stock direction forecast, beyond the L^3 associative classifier, we considered the following renowned supervised classifiers: Random Forest (RF), Support Vector Machines (SVC), Multi-Layer Perceptron (MLP), and k-Nearest Neighbors (KNN). For the L^3 classifier we used a Python library built on top of the

Table 2: Examples of level-I classification rules.

Stock	Antecedent	Class	Support	Confidence
AMZN	SMA20-50:(-5:0], EMA5-20:(-5:0], EMA8-15:(-5:0], RSI14:(30:50], AO14:(-50:0], CMO14:(-50:0], WD14:(-5:0], ADL:(1e9:inf), OBV:(0:1e9]	*Down*	4%	100%
MSFT	SMA5-20:(-5:0], EMA5-20:(-5:0], EMA8-15:(-5:0], MACD12-26:(-2:0], ADX14:(-inf:20], PPO12_26:(-5:0], CMO14:(-50:0], ATRP14:[0:10], PVO12-26:(-40:-20], OBV:(-inf:-1e9]	*Up*	2%	100%
EBAY	SMA20-50:(0:5], EMA20-50:(0:5], ATRP14:[0:10], ADX14:(20:25], PVO12-26:(-40:-20], ADL:(1e9:inf), OBV:(0:1e9], FI50:(0:1e7]	*Up*	2%	100%
EBAY	EMA20-50:(-5:0], RSI14:(50:70], ATRP14:[0:10], ADX14:(20:25], WD14:(-5:0], ADL:(1e9:inf), OBV:(0:1e9], FI13:(0:1e7]	*Down*	1.33%	100%

authors' implementation[1], whereas for the competitors we used the implementations available in the scikit-learn library [10].

In the validation phase, for the associative classifier we varied the support threshold *ms* between 0.5% and 10% and the confidence threshold between 25% and 75%. For MLP we tested both a one- and a two-layer configurations varying the size (between 10 and 30), the activation function (relu, logistic, tanh), the solver (lbfgs, sgd, adam), the learning rate (constant, invscaling) and its initialization level (between 0.0001 and 0.1). For SVC we varied the kernel (rbf, poly, linear), the degree of the polynomial kernel (3, 4, 5) and the regularization parameter C (between 0.001 and 50). For KNN we tested the number of neighbors K (between 3 and 7), the way to compute the neighbors (ball_tree, kd_tree) and the weighting schema (uniform, inverse of the distance). For RF we varied the split criterion (gini or entropy), the minimum number of samples required to split an internal node (from 2 to 10) and the minimum number of samples required to be at a leaf node (from 1 to 5).

We carried out the trading simulations by considering all the equities listed in the main U.S. stock market index (Standard&Poor 500). We backtested the trading systems on 11-year historical data, i.e., from January 1, 2007 to December 31, 2017.

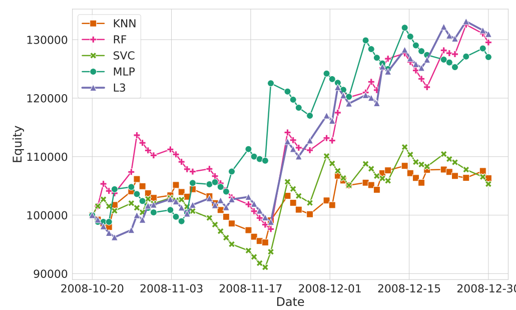
To account for different market conditions, we run simulations separately for each year. We split 1-year data as follows: the first 60% of the year was used for training, the following 20% for validation, and the remaining 20% (approximately 50 trading days per year) for testing. The average number of candidate stocks to trade per year was around 480.

We run the simulations on a local workstation with 16 GB of RAM and an Intel© Core™ i7-8700 CPU. The average time spent in training and validation by the associative classifier for each trading simulation was around 3,500s and was mainly due to association rule extraction and I/O operations (not required by scikit-learn implementations). For the other methods the average execution times were 900s (MLP), 250s (KNN), 1300s (RF), and 430 (SVC).

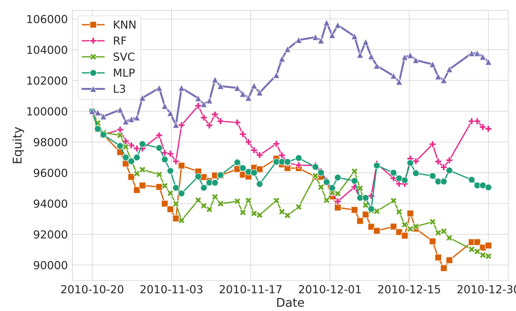
3.1 Performance analysis

Figure 1 shows the equity lines achieved using different algorithms in three representative periods: (i) Year 2008, mostly characterized by a bearish market condition due to the outbreak of the financial crisis. (iii) Year 2010, characterized by a mixed trend. (ii) Year 2014, mostly characterized by a bullish trend due to the economic recovery after the crisis.

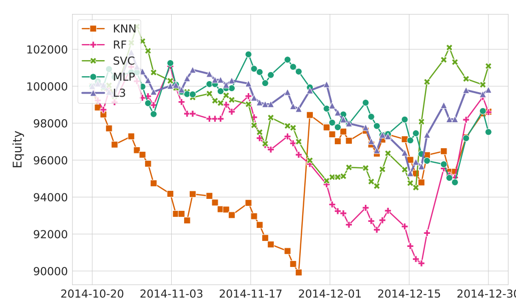
In 2008, all the classifiers overcame the initial equity. However, only the strategies based on L^3 (+31%), RF (+29.5%), and MLP (+27%) led to significant profits at the end of the year. In 2010, due to the fluctuating market conditions, the classifiers struggled to provide profitable signals. L^3 (+3.2%) was the only one achieving a positive overall percentage return. In 2014, despite a pronounced recovery of the U.S. markets, classifiers failed to generate profitable signals. After an unsuccessful start, SVC (+1.1%) overcame the initial equity,



(a) Year 2008 (bearish market).



(b) Year 2010 (mixed trend).



(c) Year 2014 (bullish market).

Figure 1: Equity lines comparison.

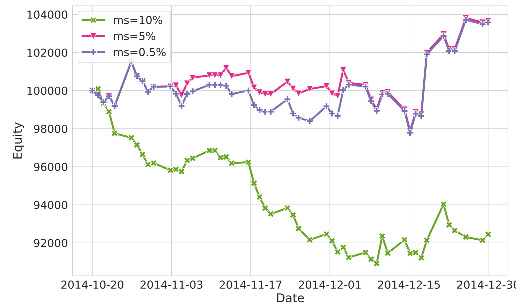
[1] l3wrapper: https://github.com/g8a9/l3wrapper

whereas L^3 yielded similar results (-1.4%), but with a more limited drawdown.

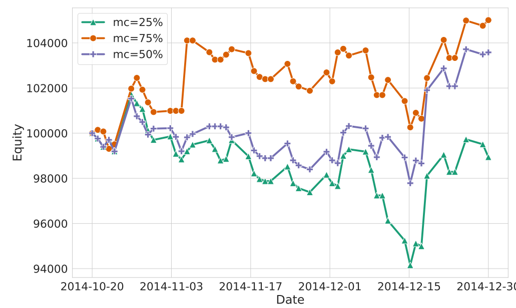
To gain insight into the performance of the systems from 2007 and 2017, in Table 3 we report more detailed performance statistics. L^3 yielded the lowest maximum drawdown (MD) among all the tested strategies (always below 7%, with a mean of 3.6%) thus making temporary losses psychologically sustainable. In 7 years out of 11 L^3 placed either first or second in terms of MD. The average number of operations closed at the stop loss level using L^3 is slightly lower than those achieved by to other classifiers, and the mean of the average return per position is lightly higher. In 8 years out of 11 L^3 placed either first or second in terms of overall return. These empirical results indicate that L^3 rules provide reliable forecasts in the short-term.

To further compare the performance of L^3 against the other tested classifiers, we adopted the Borda count voting system and the Wilcoxon statistical test [14]. To apply the Borda count voting procedure, separately for each year we ranked classifiers by decreasing value of a specific metric of interest and assigned 5 points to the first placed classifier, 4 to the runner-up, and so on. Then, we summed up the yearly score. The classifier achieving the highest score over all the 11 years is the winner. According to the Borda count outcomes, the L^3 classifier performed best in terms of both overall return and maximum drawdown rankings.

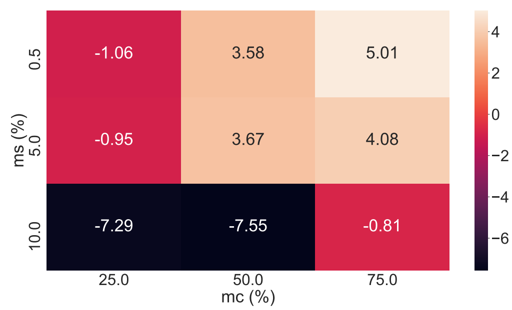
The Wilcoxon signed-rank test is an established, non-parametric statistical hypothesis test adopted to compare the mean ranks of two populations. In our context, we consider as samples the values of a metric of interest achieved by the classifiers each year. We compared L^3 with each of the other classifiers running one-sided tests. The null hypothesis indicates that the median of the difference is negative (i.e. the median performance of L^3 is worse than its competitor). Considering the difference in terms of overall return, we rejected the null-hypothesis with a confidence level of 64% (L^3 vs RF) in the worst case, and with a confidence level of 92% in the best case (L^3 vs KNN). Considering the difference in the minimum equity value (i.e. the maximum drawdown), we rejected the null-hypothesis with a confidence level of 86% (L^3 vs RF and MLP) in the worst case, and with level 92% in the best case (L^3 vs KNN).



(a) Effect of minimum support *ms*: equity lines (*mc*=50%)



(b) Effect of minimum confidence *mc*: equity lines (*ms*=0.5%)



(c) Overall percentage returns

Figure 2: Effect of thresholds *ms* and *mc*. L^3. Year 2014.

3.2 Analysis of L^3 settings

We tested various configuration settings of the associative classifier. First, we varied the granularity level of the discretization ranges reported in Table 1 by using coarser and fine-grained intervals. The results, not reported here due to the lack of space, showed that coarser discretization levels yield less specialized rules thus degrading prediction performance (e.g., overall return in 2011 37.19% with fine-grained ranges vs. 25.49% with coarser ranges). Conversely, further decreasing the granularity level is likely to be not beneficial because the probability of occurrence of the corresponding items would become rather low.

Next, we separately analyzed the impact of the support and confidence thresholds. Figures 1 shows the results on a sample year. Specifically, Figure 2a shows the equity lines achieved using different support thresholds and a fixed confidence threshold value. The higher the threshold value the less specialized the extracted rules. Generally speaking, less specialized rules yield lower-quality trading performance at the cost of a lower computational power. However, in this particular case, setting *ms* to 5% yielded returns similar to *ms*=0.5%. Hence, the best trade-off was achieved setting *ms* to 5%. Figure 2b compares the equity lines achieved using different confidence thresholds. Setting low confidence thresholds (e.g., 25%) could degrade prediction quality since the extracted rules are on average less reliable. Oppositely, too high confidence thresholds may produce data overfitting. As summarized by the heatmap in Figure 2c, in our experiments the best compromise was to set medium support and confidence thresholds (5% and 50%, respectively).

4 CONCLUSIONS AND FUTURE WORKS

The paper has explored the use of associative classification model in stock trading systems. The models based on association rules were trained on historical data at daily granularity and used to drive trading signals generation. Thanks to the interpretability of

Table 3: Detailed statistics per year

	Overall Return (%)					Maximum Drawdown (%)					Positions Closed at Stop Loss Level (%)				
Year	KNN	L^3	MLP	RF	SVC	KNN	L^3	MLP	RF	SVC	KNN	L^3	MLP	RF	SVC
2007	-7.611	24.104	1.133	-1.195	1.361	11.983	0.000	0.000	7.448	2.331	74.150	62.585	66.667	64.626	68.027
2008	6.306	30.968	27.024	29.531	5.317	4.671	3.779	1.150	2.384	8.914	82.313	79.592	82.313	80.272	85.714
2009	12.151	3.574	2.091	0.662	2.206	1.150	2.353	2.158	4.296	4.867	59.184	63.946	61.905	62.585	63.946
2010	-8.732	3.203	-4.953	-1.145	-9.422	10.205	0.894	6.354	5.868	9.422	64.626	55.782	60.544	55.782	65.306
2011	1.580	25.486	7.964	27.629	15.911	4.837	2.574	3.343	3.420	1.908	67.347	57.143	62.585	61.905	65.306
2012	-4.700	2.987	2.591	4.619	-8.140	7.728	6.647	6.751	6.815	9.590	63.889	61.806	61.806	56.944	71.528
2013	-7.018	5.560	-3.782	-3.307	6.523	7.510	0.014	4.534	5.241	3.968	53.741	48.299	59.864	59.184	57.143
2014	-1.374	-0.193	-2.477	-1.406	1.091	10.080	4.708	5.201	9.585	5.485	59.184	51.701	55.102	59.184	63.265
2015	-2.577	-5.653	2.778	10.372	7.389	4.417	6.568	7.636	0.930	5.306	64.626	61.905	65.306	61.905	62.585
2016	9.736	9.251	19.311	-6.286	6.920	1.795	2.875	3.588	10.943	3.449	57.823	68.707	65.306	71.429	72.109
2017	0.955	-3.942	6.444	2.345	2.367	6.558	6.497	3.371	3.256	2.979	70.748	54.422	63.265	62.585	62.585
Avg	-0.117	**8.668**	5.284	5.620	2.866	6.449	**3.355**	4.008	5.471	5.293	65.239	**60.535**	64.060	63.309	67.047
Std	7.068	12.486	9.829	12.157	7.104	3.469	2.517	2.378	3.057	2.821	8.178	8.652	6.825	7.011	7.503

	Average Return Per Position (%)					Weighted Accuracy					Weighted F-score				
Year	KNN	L^3	MLP	RF	SVC	KNN	L^3	MLP	RF	SVC	KNN	L^3	MLP	RF	SVC
2007	-0.070	0.635	0.147	-0.025	0.214	0.344	0.340	0.340	0.351	0.345	0.345	0.339	0.318	0.329	0.320
2008	0.242	0.704	0.526	0.631	0.060	0.348	0.345	0.339	0.354	0.339	0.351	0.338	0.310	0.347	0.308
2009	0.309	0.132	0.090	0.138	0.050	0.346	0.341	0.343	0.347	0.344	0.378	0.366	0.359	0.361	0.347
2010	-0.305	-0.001	-0.230	0.037	-0.260	0.341	0.342	0.343	0.348	0.339	0.464	0.469	0.451	0.472	0.451
2011	0.124	0.632	0.283	0.700	0.407	0.344	0.338	0.336	0.344	0.351	0.339	0.342	0.308	0.322	0.326
2012	-0.070	-0.030	0.155	0.155	-0.297	0.334	0.335	0.335	0.343	0.338	0.465	0.459	0.460	0.468	0.462
2013	-0.202	0.071	-0.044	-0.103	0.177	0.335	0.343	0.339	0.340	0.336	0.528	0.532	0.531	0.544	0.540
2014	-0.024	-0.031	0.069	0.136	0.011	0.343	0.346	0.340	0.345	0.341	0.499	0.503	0.485	0.495	0.486
2015	-0.029	-0.090	0.010	0.274	0.181	0.343	0.341	0.337	0.346	0.344	0.409	0.413	0.390	0.405	0.393
2016	0.238	0.242	0.465	-0.177	0.117	0.347	0.340	0.337	0.346	0.345	0.474	0.466	0.455	0.464	0.457
2017	-0.046	0.053	0.147	-0.001	0.048	0.337	0.340	0.334	0.338	0.337	0.547	0.537	0.531	0.545	0.542
Avg	0.015	**0.211**	0.147	0.160	0.064	0.342	0.341	0.338	**0.346**	0.342	**0.436**	0.433	0.418	0.432	0.421
Std	0.193	0.301	0.217	0.280	0.202	0.005	0.003	0.003	0.005	0.004	0.075	0.077	0.085	0.083	0.087

the rule-based model, the expert could closely monitor on-the-fly the quantitative trading process.

As future extension of the present work, we plan to elaborate more on the characteristics of the rule-based model. Specifically, we would like to identify the rule-based patterns that are most likely to be used in forecasting the next-day stock direction, to verify the adherence of the automatically generated rules to technical analysis fundamentals, and to design an adaptive decision support system able to get feedbacks from the domain experts and on-the-fly update the associative model accordingly.

REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. 1993. Mining Association Rules between Sets of Items in Large Databases. In *ACM SIGMOD Record*, Vol. 22. ACM, New York, 207–216.

[2] K. K. Ang and C. Quek. 2006. Stock Trading Using RSPOP: A Novel Rough Set-Based Neuro-Fuzzy Approach. *IEEE Transactions on Neural Networks* 17, 5 (2006), 1301–1315.

[3] Elena Baralis, Silvia Chiusano, and Paolo Garza. 2008. A Lazy Approach to Associative Classification. *IEEE Trans. Knowl. Data Eng.* 20, 2 (2008), 156–171. https://doi.org/10.1109/TKDE.2007.190677

[4] Tai-liang Chen and Feng-yu Chen. 2016. An intelligent pattern recognition model for supporting investment decisions in stock market. *Information Sciences* 346 (02 2016).

[5] David Enke and Suraphan Thawornwong. 2005. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications* 29, 4 (2005), 927 – 940.

[6] Bing Liu, Wynne Hsu, and Yiming Ma. 1998. Integrating Classification and Association Rule Mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*. AAAI Press, 80–86.

[7] Harry M. Markowitz. 1991. *Portfolio Selection: Efficient Diversification of Investments* (2 ed.). Wiley.

[8] J.J. Murphy. 1999. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York Institute of Finance.

[9] E. I. Papageorgiou and J. L. Salmeron. 2013. A Review of Fuzzy Cognitive Maps Research During the Last Decade. *IEEE Trans. on Fuzzy Syst.* 21, 1 (2013), 66–79.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[11] Sharifa Rajab and Vinod Sharma. 2019. An Interpretable Neuro-Fuzzy Approach to Stock Price Forecasting. *Soft Comput.* 23, 3 (Feb. 2019), 921–936. https://doi.org/10.1007/s00500-017-2800-7

[12] Rundo, Trenta, di Stallo, and Battiato. 2019. Machine Learning for Quantitative Finance Applications: A Survey. *Applied Sciences* 9, 24 (Dec 2019), 5574. https://doi.org/10.3390/app9245574

[13] Vlasenko, Vlasenko, Vynokurova, Bodyanskiy, and Peleshko. 2019. A Novel Ensemble Neuro-Fuzzy Model for Financial Time Series Forecasting. *Data* 4, 3 (Aug 2019), 126. https://doi.org/10.3390/data4030126

[14] Frank Wilcoxon. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin* 1, 6 (1945), 80–83. http://www.jstor.org/stable/3001968

[15] L. Yang, Z. Zhang, S. Xiong, L. Wei, J. Ng, L. Xu, and R. Dong. 2018. Explainable Text-Driven Neural Network for Stock Prediction. In *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*. 441–445.