ABSTRACT Decision support systems using Artificial Intelligence in the context of financial services include different application ranging from investment advice to financial trading. The analysis of order flow provides many challenges that can be addressed by Machine Learning (ML) techniques in order to determine an optimal dynamic trading strategy. The first step in this direction is represented by the outcome analysis of order flow: the model should identify strong predictors that determine a positive/negative outcome. The aim of this work is the proposal of a closed-loop ML approach based on decision tree (DT) model to perform outcome analysis on financial trading data. The overall approach is integrated in a Decision Support System for Outcome Analysis (DSS-OA). Taking into account the model complexity, the DT algorithm enables to generate explanations that allow the user to understand (i) how this outcome is reached (decision rules) and (ii) the most discriminative outcome predictors (feature importance). The closed-loop approach allows the users to interact directly with the proposed DSS-OA by retraining the algorithm with the goal to a finer-grained outcome analysis. The experimental results and comparisons demonstrated high-interpretability and predictive performance of the proposed DSS-OA by providing a valid and fast system for outcome analysis on financial trading data. Moreover, the Proof of Concept evaluation demonstrated the impact of the proposed DSS-OA in the outcome analysis scenario.

INDEX TERMS Finance, decision support systems, financial management, machine learning, decision trees, outcome analysis.

I. INTRODUCTION
The exponential growth of big data analytics in recent years has also had a consistent impact on the financial world. Exploiting Artificial Intelligence (AI) methodologies, it is possible to analyze a huge amount of data in order to discover hidden patterns within it which may help investors in their profit operations on financial markets [1], [2]. In particular, this data analysis procedure may allow to identify anomalies and unusual events, to mitigate critical risks related to human error and to predict the outcome of a future event by analyzing backward-looking data [3].

The application of AI in financial analytics is also affecting the practice of automated trading [4]. Markets and trades generally comprised a huge amount of well-structured data that lies the foundation from the application of a data-driven model rather than a knowledge-based technique [5]. Experts and decision support systems (DSS) using a portion of AI are widely spread in different domains [6]–[9]. In the context of financial services DSS are often used for different application ranging from investment advice, credit granting, online portfolio selection [10] and service to trading [5]. The analysis of order flow provides many challenges that can be addressed by Machine Learning (ML) techniques in order to determine an optimal or sub-optimal dynamic trading strategy. This means that ML models have the objective to find solutions that can maximize profitability over time taking into account risk as the main factor of this task. The first step in this direction can conventionally be represented by the outcome analysis of order flow: given as input a historical series of market operations (e.g. 1 month), the model should identify strong predictors that determine a positive/negative outcome (i.e. IOC order fill/cancel). This approach aims i) to steer data analysis for improving trading strategies and ii) to provide
information on relative benchmarks (e.g. best execution) to brokers.

In the context of financial prediction literature, knowledge-based approaches have been proposed for time series forecasting problem and financial risk management using respectively fuzzy candlestick patterns [11] and pragmatic approach to the design of knowledge repositories [12]. However, considering the increasing amount of longitudinal data and available features, expert systems based on the knowledge-based technique in financial services tend to tackle only quite narrow, routine tasks with required knowledge already mapped out. This fact may limit the automation of the prediction process as well as the global comprehension of the task. In this context data-driven model may provide a valuable solution to achieve a reliable forecast provided one can understand how the prediction was achieved and which feature determines the predicted outcome (i.e. outcome analysis). Both these conditions may provide the acceptance of the financial community [5].

Although complex models such as standard Neural Networks (NN) and other Deep Learning techniques have been applied to financial trading [13], [14] and stock prediction [15], Decision Tree (DT) based ML models are capable to find a multivariate relationship between input and output variables, also dealing with a dataset consisting of thousands of observations and a large heterogeneous feature set. At the same time, DT models allow providing a direct interpretation of the most discriminative predictors [16], [17].

For what concerns the financial forecasting different ML and DL algorithms were proposed for learning in the presence of sequence data. These approaches range from standard Deep Neural Network (DNN) [18], [19] to recurrent neural network (i.e. Long-short-term memory) [20]–[22]. However, the potential of DL approaches may be limited by the interpretability of the model [23], which does not always allow to retrieve the feature importance and represents a crucial aspect in order to perform an outcome analysis in financial data. On the other hand, there is an increasing interest in the introduction of clustering approaches to extract useful knowledge from existing collected data to help make reasonable decisions for new customer requests (e.g. user credit category, the confidence of expected return). However, starting from the motivation of the outcome analysis (i.e. the model should identify strong predictors that determine a positive/negative outcome) we have decided to apply a supervised learning strategy (i.e., DT) to improve at the same time prediction and interpretability results [24]. At the same time, the DT exploits the Gini index measurements by encouraging the creation of uniform regions. The Gini and Information Gain index are closely related together, they can be exploited for feature selection and outcome analysis as a univariate filter-based approach. However, the employment of Decision Tree allows performing an in-deep, non-linear multivariate feature selection by discovering also the relation (i.e. decision rule) among different predictors [25]–[27].

Our objective is in line with the recent Ethics guidelines for trustworthy AI (https://ec.europa.eu/digital-single-market/en/news/ethics-guidelines-trustworthy-ai) where the mission in the capital market scenario is also to ensure key requirements such as transparency and accountability. The principle of transparency and interpretability are closely related together. The AI model should ensure high-interpretabiltiy by providing outcomes that are understand-able and relevant to the involved stakeholder. Accordingly, the high-transparency should ensure a high level of support (i.e. decision support system) while providing the information about the system’s capabilities and limitations for the involved users/customers. The principle of accountability plays a central role, especially in critical applications such as the capital market. The assessment of the outcome is clearly provided for each processing and algorithm step: the user/customers can actively interact with the AI model, ensuring that the objectives are achieved. These guidelines are the foundations that are taken into account in the design of the proposed Decision Support System that lead us towards the introduction of a simple and interpretable ML model that are functional models for achieving these purposes in the capital market scenario.

The aim of this work is the proposal of a closed-loop machine learning approach based on features selection and classification/regression algorithm (i.e., Decision Tree [DT]) to perform outcome analysis on financial trading data. The overall approach is integrated as a Decision Support System for Outcome Analysis (DSS-OA) for improving the outcome analysis of order flow. Taking into account the model complexity (i.e. depth of the tree) [28], the DT algorithm enables to generate explanations that allow the user to understand (i) how this outcome is reached (i.e decision rules) and (ii) the most discriminative outcome predictors (i.e. feature importance). Feature importance gives a score for each predictor, the higher the score more important or relevant is the feature towards your output variable. This information may be salient to support the outcome analysis of the trading analytics. In this scenario, the closed-loop approach allows the user to (i) visualize the decision rules and the feature importance (ii) uncheck features that are not interesting for the outcome analysis and (iii) retraining the model for discovering relevant features that are not easily correlated with the outcome.

This work contributes to the knowledge and data engineering field for the following reasons:

- it introduces a closed-loop machine learning approach based on DT algorithm as the main core of a DSS-OA for performing outcome analysis on financial trading data;
- it seeks to detect the most discriminative trading predictors which are not easily correlated with the target variable (outcome analysis). Such analysis and interpretation could support the short and long term forecast of the order flow for financial trading prediction;
- it allows the user to interact directly with the proposed DSS-OA by visualizing the learned decision rules and feature importance. Based on these outcomes the users
FIGURE 1. Flow-chart of the proposed DSS-OA approach: decision Support System for improving the outcome analysis of
order flow. The overall framework consists of 4 steps: pre-processing, feature selection, feature encoding and decision tree.
The closed-loop machine learning approach allows the user to (i) visualize the decision rules and the feature importance (ii)
uncheck features that are not interesting for the outcome analysis and (iii) retraining the model for discovering relevant
features that are not easily correlated with the outcome. Two use-cases have been tested: post and pre-trade procedure.

TABLE 1. Description of the dataset: features types and domain.

| Domain                  | Categorical | Ordinal | Continuous |
|-------------------------|-------------|---------|------------|
| Order status            | 112         | 15      | 24         |
| Instrument definition   | 31          | 4       | 5          |
| Client ID               | 19          | 4       | -          |
| Instrument trading data | 23          | 6       | 6          |
| Others                  | 1           | -       | 28         |
| Total                   | 186         | 29      | 63         |

can uncheck features that are not interesting for the outcome analysis, and retraining the algorithm with the
goal to a finer-grained outcome analysis.

II. MATERIALS AND METHOD

Figure 1 shows the flow-chart of the proposed approach. The overall framework comprised of 4 steps: pre-processing, feature
selection, feature encoding and decision tree. The user can interact with the overall framework by visualizing the
decision rule and the feature importance. Taking into account this information and the fact that the user has full knowledge
about the domain, he/she can decide to exclude further features that are not interesting for the outcome analysis, before
retraining the DT model.

A. DATASET

The dataset Trading History employed in this study consists of an order flow of 5 months equal to 324530 trades observations
and a feature set of 278 variables (trade features) (see Table 1). The trade features are shown in terms of domain
(order status, instrument definition, client ID, Instrument trading data and Others) and types (categorical, ordinal and
continuous). Trading History data are anonymous in terms of customers and instrument and their use, detention and conserv-
ation are regulated by an agreement between ATS company, Università Politecnica delle Marche and data owners. The
dataset refers to a private dataset from ATS company and it represents a recent year’s operations in several markets.
The dataset was used because (i) it is an example of how trading operations data can be made available from a trading
platform and (ii) it contains a good variety of operations on different markets/asset types over a consistent period.
These motivations support the implementation and the testing of the proposed DSS-OA for solving the outcome analysis
task using this dataset.

An order sent to the market can trigger 0 or more trades. In the most common case, one order produces one trade, but
the order could expire or be canceled by the user without being filled (i.e without producing any trade), or the order
could be executed in multiple steps, trading fractions of the order quantity in multiple trades until it gets filled. According
to the ATS company, four target variables have been identi-
ified on the Trading History dataset:
- TGT_LASTORDSTATUS (binary target), the last state of the order (i.e. “Canceled” or “Filled”);
- TGT_EXECQTYPERC (continuous target), the executed quantity in relation to the quantity specified in the
  order (in percentage);
- TGT_LIMPRICE_DIFF_PERC (continuous target), the percentage of deviation of the executed price from
  the limit price;
- TGT_COMMISSION_WEIGHT_PERC (continuous target), the percentage weight of commissions in relation to
  the counter value.

Our goal is to perform individually the outcome analysis of these 4 tasks. The features set is represented by the trade
features while the response is the related target variable (i.e. TGT_LASTORDSTATUS, TGT_EXECQTYPERC,
TGT_LIMPRICE_DIFF_PERC and TGT_COMMISSION_WEIGHT_PERC). Based on the types of target variables we
have one classification and three regression tasks.

The target variable TGT_LASTORDSTATUS consists of 107859 samples targeted with the “Canceled” class and
211809 targeted with the “Filled” class. Figure 2 shows the histograms of the occurrences of the target variables
TGT_EXECQTYPERC (see Figure 2a), TGT_LIMPRICE_DIFF_PERC (see Figure 2b) and TGT_COMMISSION_WEIGHT_PERC
(see Figure 2c). Notice that the histogram was built with varying bin width by taking into account the unbalanced nature of these target variables.

B. PRE-PROCESSING

As a first step, only orders with TGT_LASTORDSTATUS equal to “Canceled” or “Filled” are maintained.
The remaining observations, corresponding to 0.3% of the entire dataset, are removed reducing the dataset to 319668 rows. Besides date/time-related features have been standardized to the ISO 8601 format. The pre-processing step aims also to identify the features as categorical, ordinal and continuous. Additionally, in this stage, we convert absolute feature related to specific field (i.e. price) into variations and deviations (i.e. price deviation) from average or baseline value. This step allows to minimize possible bias originating from different orders.

C. FEATURE SELECTION

The outcome analysis aims to discover unseen discriminative predictors related to the selected target variable. Hence, the feature selection procedure is implemented in order to discard the easily correlated features which do not provide additional value for the outcome analysis task. This procedure is different from the standard feature selection procedure for improving the generalization performance of a ML model, where only the most discriminative features are retained.

1) REMOVAL OF FEATURES WITH MISSING VALUES (NaN)
Starting from the hypothesis that the mechanism which governs the missing data occurrences is complex and completely random (i.e. the frequency of missing values is different for each feature), we supposed that the probability that features were missing may be dependent (i.e., informative) or independent (i.e., non-informative) from the target variable [29]. Hence, to explore how the mechanism of missing values is informative, we performed an extra-value imputation (i.e. values of feature columns are converted in 0 [Nan] or 1 [all other values]) and we measure the correlation between the feature column and the target variable.

Based on the different nature of target variable, two correlation metrics were used: Matthews Correlation for the binary target (TGT_LASTORDSTATUS) and Point Biserial Correlation for continuous targets (TGT_EXECQTYPERC, TGT_LIMPRICE_DIFF_PERC, TGT_COMMISSION _WEIGHT_PERC). The statistical significance of the correlation tests was set at the 5% significance level. Features with a \( p-value < 0.05 \) were discarded: this condition reflects the fact that the missing value mechanism of the related feature is highly informative and easily correlated with the target variable. This condition does not provide any added value to the user.

2) REMOVAL OF FEATURES EASILY CORRELATED WITH THE TARGET VARIABLE

Based on the different natures of dependent and independent variables, four different correlation metrics were employed to explore the correlation between each feature and target variables (see Table 2). Features with a \( p-value < 0.05 \) were discarded: this condition reflects the fact that the related feature is highly informative and easily correlated with the target variable. This condition does not provide any added value to the user.

| Methodology               | Dependent Variable | Independent Variable |
|---------------------------|--------------------|----------------------|
| Point Biserial Correlation| Continuous         | Binary               |
| Cramer’s V                | Categorical        | Binary               |
| Pearson Correlation       | Continuous         | Continuous           |
| Point Biserial Correlation| Categorical        | Continuous           |

D. FEATURE ENCODING

The encoding procedure for not-ordered and ordered categorical features is a salient step to capture information according to the nature of the independent variables [30]. For not-ordered categorical features (i.e. features related to ID, codes), label encoding was performed converting each value in the feature column to a number in order of appearance. Although this technique could insert some kind of sorting relationship between values, it has been preferred to the One-Hot encoding procedure which is not suitable due to the high amount (i.e. 186 categorical features) and high cardinality of each categorical features (i.e. the average cardinality of the categorical features is 7285). On the other hand ordered categorical features, where values have a natural order (e.g. date, timestamp), were encoded with the ordinal encoding approach, by maintaining the true order to the classes themselves. Continuous features were kept unchanged. Before performing features encoding, the remaining missing values were treated with a technique of extra-value data imputation by replacing NaN with value 999.

E. DECISION TREE

Decision Tree (DT) is a non-parametric supervised learning algorithm used to predict the value of a target variable by learning simple decision rules inferred from the data features/predictors [31]. The learning process is performed by selecting the predictor that maximizes the splitting criteria gain over all possible splits of all predictors. The DT model is conceived as a CART model to handle both categorical and continuous features. A DT consists of nodes (which are tests for the value of a certain attribute), branches (which correspond to the outcome of a test and connect to the next node or leaf) and leaf nodes (which are terminal nodes representing the prediction of the outcome, i.e. class labels or class distribution). DTs were used to perform both classification (TGT_LASTORDSTATUS) and regression tasks (TGT_EXECQTYPERC, TGT_LIMPRICE_DIFF_PERC and TGT_COMMISSION _WEIGHT_PERC). For the DT classifier, we employed the Gini’s diversity index: this measure encourages the creation of uniform regions. Differently from the misclassification rate, the Gini index is more sensitive to node probabilities and is differentiable and thus more suited to gradient-based optimization approaches [32]. Accordingly, for the DT regressor, we used the sum-of-squares error as a splitting criterion.

The Gini’s diversity index was used to evaluate the feature importance of the classification task: the more relevant is the feature the greater the Gini’s index value. Accordingly, the sum-of-squares error was used to evaluate the feature
importance of the regression tasks: the more relevant is the feature the lower the sum-of-squares error value.

F. EXPERIMENTAL PROCEDURE
The max-depth validation of the DT model was performed in order to improve the interpretability of the overall DSS-OA approach (see Section II-F1). The proposed DSS-OA was tested on two real-use case, i.e. post and pre-trade analysis (see Section II-F2) by evaluating the related metrics (see Section II-F3).

1) MAX-DEPTH VALIDATION
The interpretability/complexity of the DT model was encouraged and controlled by constraining the model size (i.e. depth of the tree) of the learned trees [28]. The depth of the tree is the maximum number of tests that have to be made for a single example to be classified [33]. Thus, the maximum depth of the tree was considered as the hyperparameter that regulates the complexity of the DT algorithm. We decided to explore the maximum depth in a range between 2 and 10 to encourage the interpretability of DT. Accordingly, the optimization of the maximum depth of the tree was performed by the implementation of a grid search and the optimization of the macro-f1 and $R^2$ score in a stratified Cross-Validation procedure according to [34]. This approach allows increasing both the discriminative power (high performance) and the interpretability of the DT model.

2) POST AND PRE-TRADE ANALYSIS
Two use-cases have been identified: Post-trade and Pre-trade outcome analysis. The Post-trade analysis process consists of recording all data from previous trading periods (i.e. trade timing, prices and other details about order flow) and evaluating performance in order to make better trading decisions while monitoring future transactions. Pre-trade analytic is the process of taking known parameters of a planned trade to predict the potential future outcome of an order and the expected market impact. In our case, the post-trade analysis is carried out considering the whole dataset. Instead, the pre-trade analysis is performed excluding all those features that are not available a-priori when the order is placed on the market (e.g. LASTORDSTATUPDTIME, indicating the time of the last order status change, and ORD_TOTAL_LIFE, which reports the time elapsed since the order was created and its last update).

3) METRICS
The proposed DSS-OA was evaluated by considering the following metrics for the classification task:
- accuracy: the percentage of correct predictions;
- Macro-precision (Precision): the Precision is calculated for each class and then take the unweighted mean;
- Macro-recall (Recall): the Recall is calculated for each class and then take the unweighted mean;
- macro-F1 (F1): the harmonic mean of precision and recall averaged over all output categories;
- confusion matrix: the square matrix that shows the type of error in a supervised paradigm;
and the following metrics for the regression tasks:
- $R^2$ score (coefficient of determination): it is a proportion between the variability of the data and the correctness of the model used. It varies in range: $[-\infty; 1]$ [35];
- Mean Absolute Error (MAE): it measures the difference between two continuous variables and represents the average of the absolute error $e_i = |y_i - x_i|$ where $y_i$ represents the prediction and $x_i$ the ground-truth. It varies in range: $[0;+\infty]$;
- Pearson correlation coefficient: it measures the linear relationship between two continuous variables. It varies in range: $[-1; 1]$, where 0 implies that there is no correlation.
- Spearman correlation coefficient: it measures the non-linear relationship between two continuous variables. It varies in range: $[-1; 1]$, where 0 implies that there is no correlation.

To test for significant difference between random classification, an independent one-sided t-test was performed comparing the F1 distribution over folds with respect to chance level (.5) Accordingly, the statistical significance of the correlation tests was set at the 5% significance level.

III. RESULTS
We first show the results of the first-loop iteration of the proposed DSS-OA (see Section III-A). In this phase,
FIGURE 3. Example of feature selection: removal of features with missing values and removal of features easily correlated with the target variable.

(a) Removal of features with missing values: Matthews correlation coefficient of the most correlated imputed categorical features (DAYTOTALCOMMISSION, TOTALCOMMISSION, DAYTOTALFEE, TOTALFEE, MYOCRIPD, ORDER_TEXT) ($p$-value < .05) (left side) with respect to the binary independent variable TGT_LASTORDSTATUS. Focus on TOTALCOMMISSION (right side): contingency table with respect to TGT_LASTORDSTATUS.

(b) Removal of features easily correlated with the target variable. Cramer correlation of the most correlated categorical features with respect to the binary independent variable TGT_LASTORDSTATUS. Focus on ORDER_TARGETSUBID (right side): contingency table with respect to TGT_LASTORDSTATUS.

TABLE 3. Predictive performance of the DSS-OA model for all the target variables (TGT_LASTORDSTATUS, TGT_EXECQTPERC, TGT_LIMPRICE_DIFF_PERC and TGT_COMMISSION_WEIGHT_PERC) post and pre-trade on first-loop (FL) and closed-loop (CL) iteration.

|                      | FL, post-trade | FL, pre-trade | CL, post-trade | CL, pre-trade |
|----------------------|----------------|---------------|---------------|---------------|
| **Target**           |                |               |               |               |
| TGT_LASTORDSTATUS    | Max-Depth      | Accuracy      | Precision     | Recall        |
| 6                    | 0.87           | 0.92          | 0.88          | 0.90          |
| TGT_EXECQTPERC       | 6              | 0.60 ± 0.06   | 17.85 ± 2.68  | 0.77          | 0.71          |
| TGT_LIMPRICE_DIFF_PERC | 6            | 0.29 ± 0.43   | 0.58 ± 0.90   | 0.68          | 0.48          |
| TGT_COMMISSION_WEIGHT_PERC | 2          | 0.05 ± 0.16   | 63.14 ± 12.69 | 0.27          | 0.05          |

|                      | FL, post-trade | FL, pre-trade | CL, post-trade | CL, pre-trade |
|----------------------|----------------|---------------|---------------|---------------|
| **Target**           |                |               |               |               |
| TGT_LASTORDSTATUS    | Max-Depth      | Accuracy      | Precision     | Recall        |
| 6                    | 0.76           | 0.84          | 0.80          | 0.81          |
| TGT_EXECQTPERC       | 6              | 0.33 ± 0.03   | 29.34 ± 1.16  | 0.58          | 0.56          |
| TGT_LIMPRICE_DIFF_PERC | 6            | −210.34 ± 63.94 | 0.61 ± 0.89 | 0.71         | 0.48          |
| TGT_COMMISSION_WEIGHT_PERC | 2          | 0.05 ± 0.16   | 63.14 ± 12.69 | 0.27          | 0.05          |

|                      | FL, post-trade | FL, pre-trade | CL, post-trade | CL, pre-trade |
|----------------------|----------------|---------------|---------------|---------------|
| **Target**           |                |               |               |               |
| TGT_LASTORDSTATUS    | Max-Depth      | Accuracy      | Precision     | Recall        |
| 6                    | 0.88           | 0.93          | 0.90          | 0.91          |
| TGT_EXECQTPERC       | 6              | 0.60 ± 0.06   | 17.78 ± 2.65  | 0.78          | 0.71          |
| TGT_LIMPRICE_DIFF_PERC | 5            | 0.04 ± 0.28   | 0.78 ± 0.85   | 0.46          | 0.36          |
| TGT_COMMISSION_WEIGHT_PERC | 3          | −0.43 ± 1.60  | 63.08 ± 16.57 | 0.11          | 0.01          |

|                      | FL, post-trade | FL, pre-trade | CL, post-trade | CL, pre-trade |
|----------------------|----------------|---------------|---------------|---------------|
| **Target**           |                |               |               |               |
| TGT_LASTORDSTATUS    | Max-Depth      | Accuracy      | Precision     | Recall        |
| 6                    | 0.79           | 0.83          | 0.85          | 0.84          |
| TGT_EXECQTPERC       | 6              | 0.34 ± 0.03   | 29.39 ± 1.16  | 0.58          | 0.56          |
| TGT_LIMPRICE_DIFF_PERC | 2            | −8.55 ± 11.13 | 0.89 ± 0.80   | 0.41          | −0.01         |
| TGT_COMMISSION_WEIGHT_PERC | 3          | −0.61 ± 1.63  | 65.49 ± 19.22 | 0.15          | 0.02          |

the user has not yet interacted with the overall framework. Then, we provide the results of the closed-loop approach (see Section III-B). In this phase, the user has interacted with the overall framework by (i) visualizing the decision rules and the feature importance (ii) eventually unchecking features that are not interesting for the outcome analysis and (iii) retraining the model for discovering relevant features that are not easily correlated with the outcome. We summarized the predictive performance for all classification (TGT_LASTORDSTATUS) and regression tasks (TGT_EXECQTPERC, TGT_LIMPRICE_DIFF_PERC and TGT_COMMISSION_WEIGHT_PERC), while we show the results related to the outcome analysis (i.e. feature importance and decision rules) for solving the classification task (TGT_LASTORDSTATUS).

A. FIRST-LOOP

Here we show the results of the first-loop iteration related to the feature selection (see Section III-A1) and classification model. The results are presented in terms of predictive performance (see Section III-A2) and outcome analysis (feature importance) (see Section III-A3) for post and pre-trade procedures.

1) FEATURE SELECTION

Figure 3 shows an example of the feature selection procedure for removing features with missing values and features easily correlated with the target variable. In particular, we show the most correlated imputed categorical features ($p$-value < .05) with respect to the binary independent variable TGT_LASTORDSTATUS (see Figure 3a).
FIGURE 4. Feature importance for the TGT_LASTORDSTATUS: post and pre-trade on first-loop and closed-loop iteration. We show the features which disclosed a feature importance greater than the 3%. The remaining features are marked as OTHER.

Then, we have discarded these features, because the missing value occurrences are easily correlated with the TGT_LASTORDSTATUS. Accordingly, we have discarded the most correlated features with respect to the target variable TGT_LASTORDSTATUS (see Figure 3b).

2) PREDICTIVE PERFORMANCE
The predictive performance of the DSS-OA model for the first-loop iteration is shown in Table 3 for the post and pre-trade procedure. F1 distribution over folds is significantly higher \((p-value < .05)\) than chance level (i.e. \(F1 = 0.5\)) for classifying TGT_LASTORDSTATUS for both post \((F1 = 0.90)\) and pre-trade \((F1 = 0.81)\) procedure. The Pearson correlation between the predicted TGT_EXECQTYPERC/TGT_LIMPRICE_DIFF_PERC and the ground-truth is significantly \((p-value < .05)\) different from 0 for both post-trade (0.77 and 0.58 respectively) and pre-trade (0.68 and 0.71 respectively) procedures.

Table 4 (top row) shows the confusion matrices for predicting TGT_LASTORDSTATUS for both post and pre-trade procedures. In the post-trade analysis both the true negative (TN) and true positive (TP) rates are high and balanced, while in the pre-trade procedure, although the TN is over chance level, it is lower than TP.

3) FEATURE IMPORTANCE
Figure 4a and Figure 4b show the feature importance computed for solving the classification task TGT_
discard a total of 77 features. One of these features (i.e., PRICE) resulted as high discriminative for solving the post and pre-trade tasks in the first-loop iteration. The DT model was retrained according to the new feature set.

1) PREDICTIVE PERFORMANCE

The new predictive performance of the DSS-OA model for the closed-loop iteration is shown in Table 3 for the post and pre-trade procedures. F1 distribution over folds is significantly higher ($p < .05$) than chance level (i.e. $F_1 = 0.5$) for classifying TGT_LASTORDSTATUS for both post ($F_1 = 0.91$) and pre-trade ($F_1 = 0.84$) procedures. The Pearson correlation between the predicted TGT_EXECQTYPERC/ TGT_LIMPRICE_DIFF_PERC and the ground-truth is significantly ($p < .05$) different from 0 for both post-trade (0.78 and 0.58 respectively) and pre-trade (0.46 and 0.41 respectively) procedures.

2) FEATURE IMPORTANCE AND FINAL DECISION RULES

Figure 4c and Figure 4d show the feature importance computed for solving the classification task TGT_LASTORDSTATUS in the closed-loop iteration. The most discriminative features remained respectively LASTORDSTATUPTIME and DIFF_CURR_LIM_PRICE for post and pre-trade procedures. As a result of the user interaction with the first iteration of the DSS-OA approach, the new feature importance analysis unveils new outcomes (e.g. ORDTYPE and ORDER_ORDERSOURCE) that were not present in the first-loop step. These outcomes may provide salient knowledge for deeply understanding, discriminative unseen predictors that are strongly relevant for the outcome analysis task.

As example, we show the first three layers of the DT decision rules for solving the TGT_LASTORDSTATUS for the post-trade procedure during the closed-loop iteration (see Figure 5). The decision rules of DT together with the features importance represent the salient outcomes of the DSS-OA that the user can visualize both in the first loop and closed-loop iterations.

C. COMPARISONS WITH RESPECT TO ML AND DL APPROACHES

We have performed a comparison with respect to ML and DL models used in a similar financial scenario. In particular, we have tested Multi-layer perceptron (MLP) which was used for financial forecasting [18], [19]. Additionally, we have performed a further comparison with a XGBoost methodology that was applied in a different domain within the financial scenario (i.e. credit scoring task) [36]. The provided experimental comparisons (see Table 5 and Table 6) with respect to XGBoost and state-of-the-art DL approaches evidenced how the DT is a trade-off between interpretability, model complexity and predictive accuracy for solving the outcome analysis task.

### Table 5. Predictive performance of the DSS-OA model for the binary target variable (TGT_LASTORDSTATUS: post and pre-trade on closed-loop (CL) iteration. Comparisons with respect to XGBoost and Multi layer perceptron (MLP).

| Methodology | Accuracy | Precision | Recall | F1   |
|-------------|----------|-----------|--------|------|
| Decision Tree | 0.88     | 0.93      | 0.90   | 0.91 |
| XGBoost     | 0.93     | 0.95      | 0.94   | 0.95 |
| MLP         | 0.80     | 0.81      | 0.94   | 0.87 |

### Table 6. Confusion matrices (rows are the true classes) for the TGT_LASTORDSTATUS (C: Canceled, F: Filled): post and pre-trade on first-loop and closed-loop iteration. Comparisons with respect to XGBoost and Multi layer perceptron (MLP).

|          | C       | F       | C       | F       |
|----------|---------|---------|---------|---------|
| Decision Tree | 0.86   | 0.14   | 0.90   | 0.10   |
| XGBoost   | 0.10   | 0.90   | 0.05   | 0.95   |
| MLP       | 0.68   | 0.71   | 0.89   | 0.79   |

D. PROOF OF CONCEPT OF THE PROPOSED DSS-OA APPROACH

The idea as the basis of the overall project originated from specific software vendor demands. As a Proof of Concept (POC) of the proposed approach we have provided a wireframe example that has led the development of the POC in terms of Front-end design (FE) (see Figures 6).

IV. DISCUSSION

Starting from the software vendor demands the term outcome analysis is defined as a measure of the result of trading activity which is defined by the user himself. This definition supports the integration of the ML approach in a DSS to support different users’ profiles. Thus, we developed a framework with the dual purpose of facilitating the post-trade analysis of user-defined outcomes (with high explainability algorithms) and, as a result of this analysis, to provide prediction tools of the same user-defined outcomes in the pre-trade phase (with a lower explainability requirement).

Given a historical series of order flow, the main goal of the outcome analysis task is to determine the strong predictors that determine a negative/positive outcome (e.g. order-specific characterization). These results may provide a relevant impact in order to provide benchmarks for brokers (e.g. best execution) and improving trading or market making strategies supported by a ML-based procedure. The proposed DSS-OA represents a salient solution for achieving these objectives by supporting the users/customers during the overall outcome analysis procedure. The proposed DT is conceived as the main core of the DSS-OA and the learned feature importance and decision rules provide a direct impact on customers/users. In particular, we maximized both accuracy...
and interpretability of the ML model: its internal behavior can be directly understood by users/customers (interpretability) and explanations (justifications) can be provided for the main factors that led to its output (outcome analysis). Furthermore,
we maximized the interaction between the users and the DSS-OA by allowing the user to interact actively with the overall system. Based on the outcome of the first loop analysis the user can decide to discard features that are not interesting for the outcome analysis and retraining the algorithm with the goal to a finer-grained outcome analysis. In fact, this procedure allows the user to discover novel relevant predictors (see Figure 4) in the closed-loop iteration (e.g. ORDTYPE and ORDER_ORDERSOURCE) that were not fully detected in the first loop iteration.

Accordingly, the outcome analysis results were supported by the high discriminative performance of the ML algorithm for both post and pre-trade procedure in the first-loop and closed-loop iteration (see Table 3). In particular, the experimental results (see Table 3) demonstrated how the learned DT/RT is discriminative (high performance) and at the same time interpretable (i.e., the maximum number of tests [decision rules] that have to be made for a single example to be classified is less than 10. Nevertheless, the difficulty of the predictive task increased from the post-trade to the pre-trade procedure (see Table 3 and Table 4). This outcome reflects the more challenging market use-case where the objective is to predict the future relevant outcomes based on past available features (i.e. excluding all those features that are not available a-priori when the order is placed on the market).

Another advantage of the proposed DSS-OA is the low computation effort of the overall procedure. The training time of DT model is approximately less than (averaged across task) 5 minutes. Thus, the DT model ensures a timely outcome analysis for each iteration of the DSS-OA. Additionally the low computation effort makes the proposed DSS-OA directly adaptable in program trading systems (i.e. algo trading and high-frequency trading): the effort required for predictive outcome computation is compatible with the low-latency provided by the proposed approach. In a software-engineered solution the computational effort of the proposed solution (i.e. testing time of the DT) is in the order of tens or a few hundred microseconds, which is compatible with software low-latency trading platform. Integrating pre-trade outcome prediction is such a cost-effective solution tier that can help to spread AI-based services to a wider user range, making it feasible for smaller brokerage firms or institutions to have limited IT and quant team resources. As a future direction, the model could be easily generalized to take into account a new set of orders placed in the markets. It would be interesting to evolve our DSS-OA by learning in presence of sequential data (i.e. a sequence of time series in the past) in order to predict the most relevant outcome in the future. In this scenario, future development could be addressed to improve the generalization performance of the algorithm (i.e. by evolving the DT into a Gradient Boosting) and speed-up the re-training procedure (i.e. by exploiting incremental learning strategies [37]).

The proposed DSS-OA may suffer from the unbalanced setting of the classification (TGT_LASTORDSTATUS) (34% canceled vs 66% filled) and regression (TGTExecqtyperc, TgtLimprice_diff_perc and TGT_commission_weight_perc) tasks. In particular, the DSS-OA is more sensitive to the unbalanced setting of TGT_LASTORDSTATUS in the more challenging pre-trade procedure (see Table 4). Accordingly, the Pearson correlation between the predicted TGT_COMMISSION_WEIGHT_PERC and the ground truth is not significantly different from 0 (see Table 3). Future works may be devoted to explore advanced oversampling strategies by including synthetic oversampling technique [38] and cost-sensitive DT model [39].

Another interesting future direction would be to extend the methodology into a multi-task regression/classification approach. This would involve modeling, discriminating and localizing the most relevant predictors by exploiting the intrinsic similarity across tasks. This strategy could be effective especially when we have limited knowledge and few observations of a single task (e.g. TGT_COMMISSION_WEIGHT_PERC) and we aim to improve the learning of a model for this specific task by using the knowledge contained in all or some of the related tasks [40].

V. CONCLUSION

As demonstrated by the high-interpretability and predictive performance provided by the experimental results and comparisons, the proposed DSS-OA represents a valid tool for performing outcome analysis on financial trading data. Moreover, the POC evaluation demonstrated the impact of the proposed DSS-OA in the outcome analysis scenario. The proposed approach is an example of decision support system based on ML algorithm for supporting the capital markets analysis.

REFERENCES

[1] S.-C. Huang, “A big data analysis system for financial trading,” in Proc. WCBM, 2017, p. 62.
[2] G. Ritter, “Machine learning for trading,” Tech. Rep., 2017.
[3] EBA Report on Big Data and Advanced Analytics, EBA, London, U.K., 2020.
[4] Y. Li, W. Zheng, and Z. Zheng, “Deep robust reinforcement learning for practical algorithmic trading,” IEEE Access, vol. 7, pp. 108014–108022, 2019.
[5] L. F. Pau, “Artificial intelligence and financial services,” IEEE Trans. Knowl. Data Eng., vol. 3, no. 2, pp. 137–148, Jun. 1991.
[6] H. Jiang, “A decision support framework for robust R&D budget allocation using machine learning and optimization,” Decis. Support Syst., vol. 121, pp. 1–12, Jun. 2019.
[7] J. Siryani, B. Tanju, and T. J. Eveleigh, “A machine learning decision-support system improves the Internet of Things smart meter operations,” IEEE Internet Things J., vol. 4, no. 6, pp. 1056–1066, Aug, 2017.
[8] L. Romeo, J. Loncarski, M. Paolanti, G. Bocchini, A. Mancini, and E. Frontoni, “Machine-learning-based design support system for the prediction of heterogeneous machine parameters in industry 4.0,” Expert Syst. Appl., vol. 140, Feb. 2020, Art. no. 112869.
[9] M. Calabrese, M. Cimmino, F. Fiume, M. Manfrin, L. Romeo, S. Ceccacci, M. Paolanti, G. Toscano, G. Ciandrini, A. Carrotta, M. Mengoni, E. Frontoni, and D. Kapeitis, “SOPHIA: An event-based IoT and machine learning architecture for predictive maintenance in industry 4.0,” Information, vol. 11, no. 4, p. 202, Apr, 2020, doi: 10.3390/info11040202.
[10] D.-J. Huang, J. Zhou, B. Li, S. C. H. Hoi, and S. Zhou, “Robust median reversion strategy for online portfolio selection,” IEEE Trans. Knowl. Data Eng., vol. 28, no. 9, pp. 2480–2493, Sep. 2016.
[11] C.-H. L. Lee, A. Liu, and W.-S. Chen, “Pattern discovery of fuzzy time series for financial prediction,” IEEE Trans. Knowl. Data Eng., vol. 18, no. 5, pp. 613–625, May 2006.

[12] M. Benarroch, “Toward the notion of a knowledge repository for financial risk management,” IEEE Trans. Knowl. Data Eng., vol. 9, no. 1, pp. 161–167, Jan. 1997.

[13] E. A. Gerlein, M. McGinnity, A. Belatreche, and S. Coleman, “Evaluating machine learning classification for financial trading: An empirical approach,” Expert Syst. Appl., vol. 54, pp. 193–207, Jul. 2016.

[14] D. Ferguson and L. Chao, “An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features,” IEEE Access, vol. 8, pp. 63666–63678, 2020.

[15] L. Shi, Z. Teng, L. Wang, Y. Zhang, and A. Binder, “DeepClue: Visual interpretation of text-based deep stock prediction,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 6, pp. 1094–1108, Jun. 2019.

[16] R. G. Leiva, A. F. Anta, V. Mancuso, and P. Casari, “A novel hyperparameter-free approach to decision tree construction that avoids overfitting by design,” IEEE Access, vol. 7, pp. 99978–99987, 2019.

[17] S. Hwang, H. G. Yeo, and J.-S. Hong, “A new splitting criterion for better interpretable trees,” IEEE Access, vol. 8, pp. 62762–62774, 2020.

[18] A. Namdar and Z. S. Li, “Integrating fundamental and technical analysis of stock market through multi-layer perceptron,” in Proc. IEEE Technol. Eng. Manage. Conf. (TEMSCON), Jun. 2018, pp. 1–6.

[19] V.-E. Seagoe, A.-D. Ciotec, and G.-S. Cucu, “Deep convolutional neural networks versus multilayer perceptron for financial prediction,” in Proc. Int. Conf. Commun. (COMCOM), Jun. 2018, pp. 201–206.

[20] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” PLoS ONE, vol. 12, no. 7, Jul. 2017. Art. no. e0180944.

[21] M. R. Vargas, B. S. L. P. de Lima, and A. G. Evsukoff, “Deep learning for stock market prediction from financial news articles,” in Proc. IEEE Int. Conf. Comput. Intell. Virtual Environ. Meas. Syst. Appl. (CIVEMSA), Jun. 2017, pp. 60–65.

[22] A. Bhowmick, A. Rahman, and R. M. Rahman, “Performance analysis of different recurrent neural network architectures and classical statistical model for financial forecasting: A case study on Dhaka stock exchange,” in Proc. Comput. Sci. Line Conf. Cham, Switzerland: Springer, 2019, pp. 277–286.

[23] Z. C. Lipton, “The mythos of model interpretability,” Queue, vol. 16, no. 3, pp. 51–57, 2018.

[24] F. Cai, N.-A. Le-Khac, and T. Kechadi, “Clustering approaches for financial time series using stacked autoencoders and long-short term memory,” PLoS ONE, vol. 107, nos. 8–10, Apr. 2016.

[25] V. Sugumaran, V. Muralidharan, and K. I. Ramachandran, “Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing,” Mech. Syst. Signal Process., vol. 21, no. 2, pp. 930–942, Feb. 2007.

[26] Y. Jeong and S.-H. Myaeng, “Feature selection using a semantic hierarchy to biomedical applications and affective computing and motion analysis,” in Proc. 6th Int. Joint Conf. Natural Lang. Process., 2013, pp. 136–144.

[27] J. Novakovic, P. Strbac, and D. Bulatovic, “Toward optimal feature selection using ranking methods and classification algorithms,” Yugoslav J. Oper. Res., vol. 21, no. 1, pp. 119–135, 2011.

[28] C. Molnar, G. Casalicchio, and B. Bischl, “Quantifying model complexity via functional decomposition for better post-hoc interpretability,” 2019, arXiv:1904.03867. [Online]. Available: http://arxiv.org/abs/1904.03867

[29] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, “Pattern classification with missing data: A review,” Neural Comput. Appl., vol. 19, no. 2, pp. 263–282, Mar. 2010.

[30] P. Cerda, G. Varoquaux, and B. Kégl, “Similarity encoding for learning with dirty categorical variables,” Mach. Learn., vol. 107, nos. 8–10, pp. 1477–1494, Sep. 2018.

[31] L. Breiman, J. Friedman, R. Olshen, and C. Stone, Classification and Regression Trees. Monterey, CA, USA: Wadsworth and Brooks, 1984.

[32] C. B. M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.

[33] S. Rüping, “Learning interpretable models,” Ph.D. dissertation, Univ. Dortmund Fachbereich Informatik, Dortmund, Germany, 2006.

[34] G. C. Cavley and N. L. Talbot, “On over-fitting in model selection and subsequent selection bias in performance evaluation,” J. Mach. Learn. Res., vol. 11, pp. 2079–2107, Jul. 2010.

[35] A. Di Buccianico, “Coefficient of determination (R^2),” in Encyclopedia of Statistics in Quality and Reliability, vol. 1, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470061572.eqv173

[36] Y. Xia, C. Liu, Y. Li, and N. Liu, “A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring,” Expert Syst. Appl., vol. 78, pp. 225–241, Jul. 2017.

[37] C. Zhang, Y. Zhang, X. Shi, G. Alpannidis, G. Fan, and X. Shen, “On incremental learning for gradient boosting decision trees,” Neural Process. Lett., vol. 50, no. 1, pp. 957–987, Aug. 2019.

[38] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002.

[39] S. Ben Jabeur, A. Sadaoui, A. Sghaier, and R. Aloui, “Machine learning models and cost-sensitive decision trees for bond rating prediction,” J. Oper. Res. Soc., pp. 1–19, 2019, doi: 10.1080/01656568.2019.1581405.

[40] Y. Zhang and Q. Yang, “A survey on multi-task learning,” 2017, arXiv:1707.08114. [Online]. Available: http://arxiv.org/abs/1707.08114

RICCARDO ROSATI was born in Cagli, in 1992. He received the Master of Science degree in biomedical engineering from Università Politecnica delle Marche, in 2018, where he is currently pursuing the Ph.D. degree with the Department of Information Engineering. His research interests include machine learning and deep learning techniques applied to biomedical field and industry 4.0 landscape. He received the Scholarship financed by the Marche Region for the Project Embedded Automation and Edge Computing for Industry 4.0.

LUCA ROMEO received the Ph.D. degree in computer science from the Department of Information Engineering (DII), Università Politecnica delle Marche, in 2018. His Ph.D. Thesis was on Applied Machine Learning for Human Motion Analysis and Affective Computing. He is currently a Postdoctoral Researcher with the Department of Information Engineering, Università Politecnica delle Marche. He is also affiliated with the Unit of Cognition, Motion and Neuroscience and Computational Statistics and Machine Learning, Fondazione Istituto Italiano di Tecnologia, Genova. His research interests include machine learning applied to biomedical applications and affective computing and motion analysis.

CARLOS ALFARO GODAY was born in Madrid, Spain, in 1994. He received the double degrees in mathematics and computer science from the Universidad Autónoma de Madrid, Spain, in February 2018. He also spent one year studying with the Politecnico di Milano, Milan, Italy, where he also completed the graduation thesis on High Performance Computing. Since March 2018, he has been a Software Engineer with the Finance Industry. Since March 2019, he has also been with ATS Advanced Technology Solutions, Milan, as a part of the Capital Market Research and Development Team. His research interests include understanding how machine learning and data analysis techniques can be applied to enhance the quality of the trading workflow.
TULLIO MENGA was born in Milan, Italy, in 1973. He received the degree in physics from the Università degli Studi di Milano, Milan, Italy. Since the beginning of his career, he has been with the Capital Market Area, mainly focusing on bringing highest performance technologies on market data and order and execution management systems. He is currently In-Charge with the Capital Market Research and Development Team, ATS Advanced Technology Solutions, Milan. His research interests include real-time AI applications both from research and low-latency engineering perspectives.

EMANUELE FRONTONI (Member, IEEE) was born in Fermo, Italy, in 1978. He is currently a Professor of computer vision and deep learning with the Department of Information Engineering, Università Politecnica delle Marche. He is involved in different EU projects and technological transfer projects with national and international companies. He is the coauthor of more than 150 international articles in his research field. His main research interests include artificial intelligence, computer vision, human behavior analysis, augmented reality, and sensitive spaces.

* * *