Feature Selection Using Approximated High-Order Interaction Components of the Shapley Value for Boosted Tree Classifier

CARLIN CHUN FAI CHU1, (Member, IEEE), AND DAVID PO KIN CHAN2

1School of Science and Technology, The Open University of Hong Kong, Hong Kong
2Data Analytics Team, PCCW Ltd., Hong Kong

Corresponding author: Carlin Chun Fai Chu (ccfchu@ouhk.edu.hk)

This work was supported in part by the Open University of Hong Kong, the Katie Shu Sui Pui Charitable Trust - Research and Publication Fund (Applied Research) under Project KS 2016/2.1.

ABSTRACT The Shapely value originates from the cooperative game theory and has been widely used in solving machine learning problems due to its high interpretability and consistency. The typical Shapley value evaluates the importance score of a feature as its average marginal contribution to a fully parameterized model under all possible feature combinations; as a result, its value includes the influences from both selected features and unselected ones. To better separate the corresponding sources, it is suggested to decompose the Shapley value into high-order interaction effect components such that the influences from each individual feature can be effectively untangled. A feature’s contribution is decomposed into distinct interaction components, and each component corresponds to a joint contribution resulting from a particular feature combination. The feature ranking is therefore evaluated with respect to the selected feature subset and calculated as the total incremental contribution by summing up its corresponding decomposed interaction values. In this study, a computationally efficient model-dependent greedy search algorithm on the high-order interaction components is proposed to solve an optimal subset selection problem with hundreds of time-lagged interrelated input features. Our algorithm extends a recently developed low-order polynomial time method for calculating the interaction component values. The empirical analysis demonstrates that the proposed method always outperforms other methods that are based on the typical Shapley value, the gain or the split count criteria, in terms of in-sample representativeness and out-of-sample forecasting performance for handling a problem with hundreds of time-lagged input features.

INDEX TERMS Computational intelligence, feature selection, machine learning, Shapley value.

I. INTRODUCTION Feature subset selection is one of the key preprocessing actions for building an efficient machine learning (ML) model. An appropriate feature selection process not only helps to improve the model’s generalizability for unseen data but also improves the corresponding understandability and reduces the related computational expenses in processing time and storage resources [13], [28]. The underlying rationale is to rank and select features according to their relevance to the target variable. Apart from the typical relevance measurements for boosted tree models such as the gain and the split count [3], the Shapley value, which is originally developed in the game theory discipline to fairly distribute the gains and losses in a multiplayer cooperative game [24], has recently been considered as another effective alternative measurement [5], [6], [9], [14], [17], [18], [22], [30]. The Shapley value of a feature is calculated as its average marginal contribution to a fully parameterized ML model under all possible feature combinations [6], [16]. Despite the success of direct usage of the Shapley value for feature selection in recent literature, it is speculated that its ranking performance can be further improved by scrutinizing the contributions from the selected feature subset. This study investigates that the subset contributions can be more properly evaluated by discarding the influence from unselected features. For example, the total contribution of a selected feature subset, which contains attributes A and B, should not include the influence from an unselected feature C. The interaction effect between attributes A and C should be excluded from the total
contribution value. In other words, the contribution value of the feature subset should only reflect the influence of selected features, not unselected ones. However, the calculation of the Shapley value averages both selected and unselected features into the considerations, which defeats the above purpose.

Decomposing the Shapley value into high-order interaction components [10], [20] helps in accurately measuring the contributions attained from the interactions among different features. A feature’s contribution is decomposed into distinct interaction components, and each component corresponds to a joint contribution resulting from a particular feature combination. Different from the direct usage of the Shapley value for feature ranking, which is based on the undecomposed contribution, we proposed that the ranking can be better evaluated by using the decomposed interaction values. The ranking score is evaluated with respect to the selected feature subset and is calculated as the total incremental contribution of a feature by summing up its corresponding decomposed interaction values. The underlying rationale is to ensure that the features are ranked and selected without the influence of unselected feature candidates.

This study is motivated by the need for a computationally efficient feature selection algorithm for handling problems with features having strong interrelated relationships and, at the same time, a large number of features are involved. It is commonly observed that input features in problems related to environmental issues are strongly related. This is because many physicochemical features are presented complementarily in nature. The joint effect resulting from specific complementary features will have a significant influence on the modeling performance; therefore, the direct use of the Shapley value for subset selection would be inappropriate, as it cannot separate the joint effects resulting from the selected and the unselected features appropriately. Moreover, for time-dependent ML problems, the input feature size is directly proportional to the number of lags used in the analysis. The use of information from $l$ lagged time periods will cause an increase in the numbers of features by a factor of $l$. In practice, the exact number of lags used in an analysis is difficult to be determined, and some researchers resort to a trial-and-error approach. A wrapper-based feature selection algorithm will become impractical to as a lots of computational resources are required for each trial. To solve the above issues, a model-dependent filter-based [21] feature subset selection algorithm using a greedy search procedure on the cumulative increments of approximated decomposed contribution values is proposed instead.

This research work addresses that the Shapley value can be decomposed into high-order interaction components and proposes a computationally efficient greedy searching algorithm for feature subset selection. It is shown that the contributions from individual features can be more appropriately evaluated with the involvement of the interaction components mathematically. In addition, to avoid the high computational resource requirements of a full decomposition, an approximated contribution evaluation method is proposed for carrying out the analysis in practice. To the best of our knowledge, this is the first study to advocate the use of the Shapley decomposition techniques for solving non-wrapper-based feature subset selection problems with hundreds of interrelated lagged input features. The empirical analysis demonstrates that the proposed method always outperforms the direct usage of the Shapley value in terms of representativeness and forecasting performance. Our method also provides better in-sample and out-of-sample results in terms of area under the receiver operating characteristic curve (ROC) than methods that are based on the gain or split count when lagged input features are used.

The paper is organized as follows: A literature review of the Shapley value and feature section is presented in section II. Section III delineates the formulation of Shapley interaction values and the proposed algorithm. Section IV reports the results of the empirical analysis together with detailed information about the data, model parameter settings and measurement methods. Concluding remarks are provided in the last section.

II. LITERATURE REVIEW

The explainability and solid theoretical ground of the Shapley value has recently attracted researchers’ attention in machine learning applications [1], [5]–[8], [15]–[17], [19]. The Shapley value originates from the concept of coalitional game under the cooperative game theory. Multiple players participate in a game with unequal individual contributions, and some particular coalitions can achieve better (or lower) overall contributions than the sum of their corresponding individual contributions. The Shapley value provides a fair estimation of the contribution from each player by considering the results from all possible coalitions. It has been shown to possess a set of desirable properties, such as efficiency, symmetry, dummy and additivity, which constitute a fair payout as a whole. Efficiency means that the sum of all players’ contributions should be equal to the grand contribution. Symmetry states that the measured values of two different players should be identical if they provide equal contributions in all possible coalitions. Dummy denotes that a player’s contribution should be measured as zero if one does not affect the prediction result under all possible coalitions. Additivity ensures that the measured results from different game components can be summed up to represent the final result [16], [19], [24].

In the machine learning discipline, the Shapley value evaluates the importance of a feature as its average marginal contribution derived from a fully parameterized machine learning model under all possible feature combinations [6], [16]. Many applications have demonstrated that the Shapley value is an effective measure of the importance of features in machine learning models. It has been employed in feature ranking processes for developing the XGBoost tree classification model for car accident detection [22]; constructing the RUSBoost tree models for hospital intensive care unit classification problems [30]; setting up gradient boosted tree
prediction models on crowd-sourced mobile speed test measurements [9]; training convolutional neural networks for recognizing the MNIST digit datasets [16], building classification and regression models on 23 well-known Weka datasets [25], training an SVM classifier with the UCI and Statlog datasets [18]; developing an iterative feature selection classification framework for 7 publicly available datasets [6] and building bootstrapped models for customer satisfaction studies for a telephone customer service center [14]. These studies illustrate that the Shapley value can serve as a measure of the importance of input features for ranking and selection purposes.

Despite the success of the direct usage of the Shapley value for feature ranking and selection, it is speculated that the performance can be further improved by scrutinizing the contributions from the selected feature subset. A typical estimation of the Shapley value allocates corresponding scores on every feature individually by averaging its contribution under the consideration of all possible combinations. When the scores are used for feature selection purposes, it would be more sensible to only consider the influence of the selected features but not the unselected ones. However, the typical estimated score is the averaged result obtained from using all features, which does not follow the above advice. The separation of the influence from unselected features requires a finer Shapley value estimation technique that can allocate the contribution for a specific combination of features [10], [20]. The contribution from a feature subset with size \( k \) is named the \( k \)-order interaction effect. With the use of the interaction effect components, the Shapley value of a feature can be formulated as the sum of its corresponding interaction components over all valid orders. In this study, we propose that the feature importance can be calculated as the sum of the interaction effect components derived from the selected feature subset instead of the typical Shapley value. Theoretically, when applying the concepts into feature selection process, the ranking of the first selected variable can be determined by the corresponding 1-order interaction effect, the ranking of the second selected variable can be determined by the sum of the corresponding 1-order and 2-order interaction effects and the ranking of the \( k \)-th selected variable can be determined by the sum of the corresponding interaction effects up to order \( k \).

Apart from having an appropriate tool for the evaluation of feature importance, a selection algorithm is another key component for carrying out a feature selection process. The algorithms are commonly classified into three categories: filter approach, wrapper approach and embedded approach [11], [13], [26]. The filter approach performs feature selection independent of any ML model, and features are selected based on their corresponding univariate or multivariate statistics. In contrast to the model independence, the wrapper approach selects features based on their performance on a specific ML model. Features are selected sequentially in an iterative manner that involves repeated training of the underlying ML model. Finally, the embedded approach aims to balance the model complexity and the fitting accuracy with the use of a penalty component (this approach is regarded as the regularization in mathematics discipline), which makes use of a penalty component to drive the model coefficients of some features to zero values and, as a result, a feature selection process is carried out.

Computational feasibility is an important consideration for practical usage [2]. Performing a feature selection process under a wrapper approach requires repeated training of the underlying ML model and estimation of the Shapley values. When the forward selection approach is employed, it requires \( \sum_{i=1}^{k} (N - i + 1) \) times of model trainings, where \( N \) and \( k \) denote the number of input features and selected features respectively. The total required computation resources for model re-training become tremendous when the size of the feature set is large and/or the complexity of the underlying ML model is high [27]. To ensure the practicality by limiting the repetition of model training, this study follows the ‘model-dependent filtering’ approach which only requires one ML model training in the whole feature selection process [21]. The feature importance score is evaluated through the result of a model trained with all the input features. This means that our method only requires a single estimation of the Shapley value and the interaction effect components. Nonetheless, a complete decomposition of the Shapley value into high-order interaction components requires large computational resources, and the situation becomes exponentially worse as the size of input features increases [10], [17]. For practicality, approximating the decomposition by limiting the interaction order is suggested as the workaround method in this study.

III. METHODOLOGY

The typical Shapley value evaluates the importance score of a feature as its average marginal contribution to a fully parameterized machine learning model under all possible feature combinations [16], [24]. However, when the ranking is applied for the feature selection process, it would be more sensible to measure the feature’s contribution without the influence of the unselected features. For example, the total contribution of a feature subset, which contains attributes A and B, should not include the influence from an unselected feature C. The interaction effect between attributes A and C should be excluded from the total contribution value. Nonetheless, the calculation of the typical Shapley value averages the contributions from all features, which defeats the above purpose. Decomposing the Shapley value into high-order interaction components [10], [20] helps to enhance the capability for feature subset selection, as the contributions from the respective features can be better separated into interaction components, and their cumulative effect can be effectively modeled. In contrast to the typical Shapley value, where the influence from all possible feature combinations is considered, our method focuses on the contribution stemming from the selected subset and excludes the effects originating from unselected features.
However, in practice, a complete decomposition of the Shapley value into high-order interaction components is impractical due to computational complexity and resource issues [10], [17], and a workaround approach is employed in our empirical analysis. We propose a computationally efficient filter-based greedy search algorithm for the Shapley interaction effect components to solve the feature subset selection problem. Our algorithm extends the recently developed low-order polynomial time method for calculating the interaction components of the Shapley value for boosted tree models [17].

A. SHAPLEY VALUE AND APPROXIMATED MAIN CONTRIBUTION

Despite the success of the direct usage of the Shapley value for feature selection in recent literature, it is speculated that its ranking performance can be further improved by scrutinizing the contributions from the selected feature subset. It is proposed that the subset contributions can be more properly evaluated by discarding the influence from the unselected features.

In this paper, we denote \( f_i(S) = E[f(x) | x_S] \) as the expected value of the machine learning model \( f \) conditioned on the selected subset of input features \( S \). \( S \) represents the set of nonzero binary indexes indicating the presence of particular features, and \( x_i \) denotes the selected subset [16], [17].

The typical Shapley value of feature \( i \) is defined as:

\[
\phi_i = \sum_{S \subseteq \mathbb{N} \setminus \{i\}} \frac{|S|!(M-|S|-1) !}{M!} [f_i(S \cup \{i\}) - f_i(S)]
\]

where \( f_i(S) \) is the approximated main contribution from feature \( i \) excluding the interaction effect; \( \Phi_i \) is the contribution resulting from the 2-order interaction between features \( i \) and \( j \); \( \Phi_{i,j,k} \) is the contribution resulting from the 3-order interaction among features \( i, j \) and \( k \); and \( \Phi_{i,j,k, l} \) is the 4-order interaction component.

Additionally,

\[
\Phi_{i,j} = \frac{1}{2!} f_{i,j} ; \quad \Phi_{i,j,k} = \frac{1}{3!} f_{i,j,k} ; \quad \Phi_{i,j,k,l} = \frac{1}{4!} f_{i,j,k,l}
\]

where \( \Phi_i \) is the contribution from feature \( i \) excluding the interaction effect; \( \Phi_{i,j} \) is the contribution resulting from the 2-order interaction between features \( i \) and \( j \); \( \Phi_{i,j,k} \) is the contribution resulting from the 3-order interaction among features \( i, j \) and \( k \); and \( \Phi_{i,j,k, l} \) is the 4-order interaction component.

The permutation factor indicates that the high-order interaction contribution value is divided equally among the possible sequences. In other words, it defines \( \Phi_{i,j} = \Phi_{j,i} \), \( \Phi_{i,j,k} = \Phi_{j,k,i} = \Phi_{j,k,i} = \Phi_{k,j,i} \) and so on.

The decomposition formula states that the typical Shapley value can be decomposed into high-order interaction components. In practice, calculating a complete set of high-order interaction components is computationally expensive [10] and requires an extremely large space for storing the results [15], [17]. Consequently, a complete decomposition of the Shapley value can be very expensive, especially for cases where the number of input features is not small. For example, when the size of the input features is 20, there are 1,048,555 high-order interaction components for a single data instance (row). The situation worsens further for boosted tree models, where tens of tree models are involved in the calculation process. To make the decomposition process computationally feasible, it is proposed to limit the decomposition up to the 2-order interaction components and consolidate the high-order components into a single random term \( \varepsilon \).

\[
\phi_i \approx \phi_i + \sum_{j \neq i} \Phi_{i,j} + \varepsilon
\]

Recently, a low-order polynomial time algorithm was developed to calculate the 2-order interaction components \( \Phi_{i,j} \) for boosted tree models using the SHAP (SHapley Additive exPlanations) framework [17]. By ignoring the effect from the random term \( \varepsilon \), the contribution from feature \( i \) excluding the interaction effect can be approximated as:

\[
\Phi'_i = \phi_i - \sum_{j \neq i} \Phi_{i,j}
\]

where \( \Phi'_i \) is the approximated main contribution from feature \( i \) excluding the interaction effect from the other features.

B. FEATURE SUBSET SELECTION ALGORITHM (ISV)

Our feature subset selection algorithm employs a greedy approach to search for the candidate features with the highest increment in cumulative interaction effects in each step. The ranking of the first selected variable is determined by the corresponding approximated main contribution (1-order interaction effect), and the rankings of the subsequently selected variables are determined by the sum of the corresponding approximated main contribution and 2-order interaction effect components. The algorithm, which chooses \( K \) features out of the original \( M \) input features, can be summarized as follows:
Step 1: Initialize the selected feature subset $Q$ to an empty set.

$$Q = \{\}$$ (8)

Step 2: Calculate the average SHAP contributions.

$$\mu_i = \text{Mean of } |\Phi_i| \text{ for all data instances (rows)}$$

$$\mu_{i,j} = \text{Mean of } |\Phi_{i,j}| \text{ for all data instances (rows)}$$

where $\mu_i$ denotes the average absolute approximated SHAP contribution value of feature $i$ excluding the interaction effect. $\mu_{i,j}$ denotes the average absolute 2-order interaction SHAP value.

Step 3: Search for the attribute $i^*$ with the largest average absolute approximated SHAP contribution and include it in the selected feature set.

$$Q = Q \cup \{i^*\}$$ (9)

Step 4: Calculate the incremental values attained by the unselected features.

$$\Delta_j = \mu_j + \sum_{q \in Q} \mu_{j,q} + \sum_{q \in Q} \mu_{q,j} - \mu_j - 2 \sum_{q \in Q} \mu_{j,q}$$ (10)

where $\Delta_j$ represents the increment in the cumulative contribution value by including feature $j$.

Step 5: Search for the attribute with the maximal increment $j^*$ and include it in the selected feature subset.

$$Q = Q \cup \{j^*\}$$ (11)

Step 6: Repeat Steps 4 and 5 until the number of selected variables reaches $K$.

The proposed model-dependent filter-based greedy selection algorithm, which aims to locate features with the largest increments, is named the Interaction Shapley Value (ISV) algorithm in this paper. The logic flow of the proposed algorithm is depicted in Figure 1, and the computational complexity of the algorithm is $O(MK^2)$. The source code of the algorithm is available in the IEEE DataPort and a detailed explanation on the complexity can be found in Appendix.

IV. EMPIRICAL ANALYSIS

A. DATA

The prediction of the occurrence of red tide requires the use of time-lagged interrelated chemical and climatological information. Recent studies have demonstrated that the modeling accuracy can be further improved by including information collected from nearby coastal regions [4], [12], [29]. As a result, the prediction of red tide involves the use of hundreds of interrelated time-lagged features where some of them are presented complementarily in nature. The set of complementary features give significance influence (interaction effect) on the prediction model and the feature ranking process should be carried out cautiously to separate the interaction effects resulted from the selected and the unsellected features appropriately. Our empirical analysis aims to model the one-step-ahead forecast of red tide at the DM1 station using the lagged data collected from seven surrounding water quality monitoring stations and related climatological information. In this study, the empirical data are collected from the Hong Kong Environmental Protection Department and the Hong Kong Observatory from January 1997 to December 2016, totally 240 monthly records. There are 342 physiochemical input features collected from seven water stations surrounding the DM1 station. The lag period 0 denotes the use of data obtained in time $t$ to predict the 1-step-ahead period target (i.e. at time $t+1$). The lag period 1 and 2 indicates the use of additional historical data from time $t-1$ and time $t-2$ periods as the input features. A concentration

1 The Hong Kong Environmental Protection Department - Marine Water Quality Data. http://epic.epd.gov.hk/EPICRIVER/marine/?lang=en

2 The Hong Kong Observatory - Climatological Information Services. http://www.hko.gov.hk/cis/climat_e.htm
level of chlorophyll-a of at least 20 $\mu$g/L is considered to be a positive class for the occurrence of red tide. Three distinct datasets with 0 to 2 lag periods are constructed, and their properties are summarized in TABLE 1. The empirical data used in this study is accessible via the IEEE DataPort.

**TABLE 1. Properties of water quality datasets.**

| Lag period | Overlapping with method | Up to the first 5 features | Up to the first 10 features | Up to the first 15 features | Up to the first 20 features |
|------------|------------------------|---------------------------|----------------------------|----------------------------|---------------------------|
| 0          | Gain                   | 0%                        | 12.5%                      | 16.7%                      | 20%                       |
| Split Count|                        | 75%                       | 80%                        | 86.7%                      | 83.8%                     |
| 1          | Gain                   | 5%                        | 10%                        | 20%                        | 31.30%                    |
| Split Count|                        | 70%                       | 70%                        | 80%                        | 81.30%                    |
| 2          | Gain                   | 15%                       | 25%                        | 21.7%                      | 25%                       |
| Split Count|                        | 70%                       | 72.5%                      | 75%                        | 75%                       |

**B. TREE MODEL AND PARAMETER SETTING**

The XGBoost algorithm is employed to make use of the selected features for building a classification model to predict the one-step-ahead occurrence of red tide cases. The maximum number of boosting trees used in the classifier is 500, and the maximum depth of each tree is 6. By specifying large values for the upper bounds of these parameters, the complexity of the classifier will not be limited, and the influence of the features can be fully explored [3]. In addition, the classifier is enabled with an early stopping process to prevent overfitting of the training data. The validation data used in the early stopping process is a stratified sample of 30% of training data, and the training will stop when there is no improvement for 50 successive training epochs.

**C. MEASUREMENT**

The performance of the ISV algorithm is compared with that using features selected by three different XGBoost feature importance scores, including the typical Shapley value, the gain and the split count. The Shapley value based method evaluates the feature importance as the average of the absolute values of the Shapley values of the corresponding features. The gain originates from information theory and is a measure of the total contribution from a particular feature in reducing impurity or loss in the machine learning model [23]. The split count denotes the total number of times a particular feature is used in the tree ensemble model. It assumes that a more frequently used feature should be considered to be more important. Furthermore, 4-fold cross-validation with stratified sampling on the class variable is employed in the empirical analysis. All the results reported in subsequent sections are the corresponding average values.

The following aspects are measured to investigate the distinct characteristics among the different feature selection methods.

1) **PROPORTION OF OVERLAPPING FEATURES**

Input features are scored and ranked for selection under different criteria. To demonstrate the difference between the ISV method and the other methods, the proportions of overlapping features are reported. The larger the value of the overlapping proportion, the greater the similarity between the two methods.

**TABLE 2. Overlapping proportions among various methods.**

The similarity between the selected features from the ISV algorithm and the typical Shapley value is high. The average result of the 4-fold cross-validation indicates that the value of the overlapped proportion is greater than 85% for 5, 10, 15 and 20 features. On the other hand, the ISV method produces feature sets with low similarities to those selected using the gain criteria. The overlapping proportions are very small for either a small number of selected feature size or a small lag period (i.e., fewer input features). In particular, the two methods rank 5 completely different features (0% overlapping) when the lag period is 0. When comparing the ISV algorithm with the split count method, the overlapping proportions are approximately 70%-85%. The results confirm that the ISV method ranks the input features differently from the other commonly used methods.

2) **REPRESENTATIVENESS OF SELECTED FEATURES**

It is speculated that our proposed method, which takes the interaction effect into consideration, will provide a more reasonable ranking result than those provided by the typical Shapley value method. In other words, using the same number of input features, the ranked results from our proposed method should give a better in-sample fitness. The magnitude of the in-sample area under the ROC curve (AUC) is used to measure the representativeness of a feature subset. A large AUC value indicates that a feature subset satisfactorily represents the underlying dynamics. The average AUC value is calculated by averaging all the AUCs obtained from models trained with a feature size that is smaller than a specific number. By comparing the averaged AUCs obtained under various feature sizes, the effectiveness of the corresponding feature selection method can be determined.

Referring to Table 3 and Figure 2, it is observed that the ISV method always gives a better in-sample AUC than the typical Shapley value method under most situations. However, the ISV method does not beat the split count.
TABLE 3. In-sample average AUC obtained under different feature sizes.

| Lag period | Selection method | Up to the first 5 features | Up to the first 10 features | Up to the first 15 features | Up to the first 20 features |
|------------|------------------|----------------------------|----------------------------|-----------------------------|-----------------------------|
|            | ISV              | 0.9250                    | 0.9491                    | 0.9627                      | 0.9693                      |
| 0          | Shapley          | 0.9139                    | 0.9506                    | 0.9635                      | 0.9687                      |
|            | Gain             | 0.9119                    | 0.9540                    | 0.9659                      | 0.9741                      |
|            | Split Count      | 0.9413                    | 0.9532                    | 0.9551                      | 0.9645                      |
|            | ISV              | 0.9637                    | 0.9783                    | 0.9836                      | 0.9876                      |
| 1          | Shapley          | 0.9598                    | 0.9756                    | 0.9835                      | 0.9875                      |
|            | Gain             | 0.8798                    | 0.9210                    | 0.9470                      | 0.9602                      |
|            | Split Count      | 0.9547                    | 0.9655                    | 0.9755                      | 0.9816                      |
|            | ISV              | 0.9706                    | 0.9851                    | 0.9901                      | 0.9925                      |
| 2          | Shapley          | 0.9700                    | 0.9841                    | 0.9893                      | 0.9919                      |
|            | Gain             | 0.9187                    | 0.9531                    | 0.9687                      | 0.9760                      |
|            | Split Count      | 0.9572                    | 0.9784                    | 0.9855                      | 0.9891                      |

FIGURE 2. Average in-sample AUC with feature sizes from 1 to 10.

method when the lag period is 0. Nevertheless, the results demonstrate that the ISV method does help to provide better representativeness on average and can achieve the largest in-sample AUC.

3) FORECASTING PERFORMANCE
The average out-of-sample AUC values that reflect the forecasting performance under different input feature sizes are reported in Table 4. On the other hand, it would be beneficial to know whether the use of feature subsets can provide better performance than models that are trained with the complete feature set. The smallest and second-smallest feature sizes that give larger AUC values than those of models trained with the full feature set are reported in Table 5. A ranked feature list is considered to be superior if fewer features are required to produce the same AUC value. Similar to the setting applied in previous subsections, the maximum feature size is limited to 20.

The average out-of-sample AUC values in Table 4 show that the ISV method outperforms the Shapley value method when the feature size is 10, 15 or 20. Furthermore, it gives the best performance among the four selection methods when the feature size is 15 or 20. Additionally, the first row in Table 5 indicates that using the first 2 ranked features from the ISV method can produce an out-of-sample result better than that using the full set of 342 features. Similarly, the fifth and ninth rows show that using the first 8 or 13 ranked features produces better results than using 684 or 1026 features. This can be explained by the fact that the use of the full feature set may

TABLE 4. Average out-of-sample AUC obtained under different feature sizes.

| Lag period | Selection method | Up to the first 5 features | Up to the first 10 features | Up to the first 15 features | Up to the first 20 features |
|------------|------------------|----------------------------|----------------------------|-----------------------------|-----------------------------|
|            | ISV              | 0.6789                    | 0.7029                    | 0.7088                      | 0.7110                      |
| 0          | Shapley          | 0.6804                    | 0.6933                    | 0.7004                      | 0.7045                      |
|            | Gain             | 0.6492                    | 0.6753                    | 0.6891                      | 0.6973                      |
|            | Split Count      | 0.6356                    | 0.6736                    | 0.6900                      | 0.7003                      |
|            | ISV              | 0.7149                    | 0.7562                    | 0.7763                      | 0.7825                      |
| 1          | Shapley          | 0.7308                    | 0.7505                    | 0.7673                      | 0.7757                      |
|            | Gain             | 0.6305                    | 0.6305                    | 0.6706                      | 0.7029                      |
|            | Split Count      | 0.7076                    | 0.7312                    | 0.7372                      | 0.7448                      |
|            | ISV              | 0.6848                    | 0.7149                    | 0.7374                      | 0.7446                      |
| 2          | Shapley          | 0.6763                    | 0.6975                    | 0.7184                      | 0.7396                      |
|            | Gain             | 0.6325                    | 0.6717                    | 0.6881                      | 0.7007                      |
|            | Split Count      | 0.6824                    | 0.7107                    | 0.7318                      | 0.7414                      |

TABLE 5. Comparison with models trained with the complete feature set.

| Lag period | Selection method | Smallest feature size | Second smallest feature size |
|------------|------------------|-----------------------|-----------------------------|
|            | ISV              | 2                     | 6                           |
| 0          | Shapley          | 2                     | 5                           |
|            | Gain             | 7                     | 8                           |
|            | Split Count      | 5                     | 6                           |
|            | ISV              | 8                     | 11                          |
| 1          | Shapley          | Nil                   | Nil                          |
|            | Gain             | Nil                   | Nil                          |
|            | Split Count      | Nil                   | Nil                          |
|            | ISV              | 13                    | 14                          |
| 2          | Shapley          | 18                    | 19                          |
|            | Gain             | Nil                   | Nil                          |
|            | Split Count      | Nil                   | Nil                          |
result in the creation of overfitted models. Overall, the ISV method is the only method that outperforms models trained with the full feature set under all lag periods. In addition, it always produces a better ranked feature list than the Shapley value method.

V. CONCLUSION

By decomposing the typical Shapley value into high-order interaction components, it was shown mathematically and empirically that the contributions from individual features can be better evaluated. It is because the decomposition of interaction components helps in measuring the contribution attained from a specific combination of features and, as a result, the ISV algorithm can better estimate the underlying contribution from the selected feature set without the influence from the unselected features. We proposed a low-order polynomial time feature subset selection algorithm using a greedy search procedure on the cumulative increments of approximated Shapley contribution values. The underlying rationale was to ensure that the features are ranked and selected without the influence of the unselected feature candidates. The empirical analysis demonstrated that the proposed ISV method always outperforms the typical Shapley value-based method in terms of representativeness and forecasting performance. The ISV method also provides better in-sample and out-of-sample AUC results than methods that are based on the gain or the split count criteria when lagged input features are used. Furthermore, the ISV algorithm is the only method that can identify small feature subsets with superior out-of-sample forecasting performance compared to models trained with a complete feature set under 3 different lag periods. It is expected that the performance of ISV method will be further improved if the 3-order interaction components are involved in the approximation of the main contribution.

The concept of ranking features using high-order interaction components of Shapley value can be further extended to other machine learning problems. However, an efficient Shapley value decomposition procedure for general machine learning models is required, and it is not currently available.

APPENDIX COMPUTATIONAL COMPLEXITY OF THE ISV ALGORITHM

Our proposed greedy algorithm begins with locating the largest average absolute approximated SHAP contribution value and then searches for the attribute with the largest increment of cumulative contribution in each iteration. The total number of subset evaluations of the greedy approach can be calculated as:

\[ M + \sum_{i=1}^{K-1} i \times (M - i) \]  \(\text{for } K < M\)  \(\text{(12)}\)

The above formula can be rewritten as:

\[ M + \sum_{i=1}^{K-1} i \times (M - i) = M + \sum_{i=1}^{K-1} (iM - i^2) \]  \(\text{(13)}\)

VOLUME 8, 2020

REFERENCES

[1] M. Ancona, C. Zireli, and M. Gross, “Explaining deep neural networks with a polynomial time algorithm for Shapley values approximation,” in Proc. 36th Int. Conf. Mach. Learn. (ICML), 2019, pp. 272–281.

[2] J. A. Bins and B. Draper, “Feature selection from huge feature sets,” in Proc. IEEE Int. Conf. Comput. Vis., vol. 2, Jul. 2001, pp. 159–165.

[3] T. Chen and C. Gaestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 785–794.

[4] C. C. F. Chu, S. C. Yuen, and Y. K. Wong, “Deep neural network for marine water quality classification with the consideration of coastal current circulation effect,” in Proc. Int. Conf. Intell. Sustain. Syst. (ICISS), Dec. 2017, pp. 389–394.

[5] S. Cohen, G. Dror, and E. Ruppin, “Feature selection via coalitional game theory,” Neural Comput., vol. 19, no. 7, pp. 1939–1961, Jul. 2007.

[6] S. Cohen, E. Ruppin, and G. Dror, “Feature selection based on the shapley value,” in Proc. ICIAR Int. Jt. Conf. Artif. Intell., 2005, pp. 665–670.

[7] A. Datta, S. Sen, and Y. Zick, “Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2016, pp. 598–617.

[8] K. Fujimoto, I. Kojadinovic, and J.-L. Marichal, “Axiomatic characterizations of probabilistic and cardinal-probabilistic interaction indices,” Games Econ. Behav., vol. 55, no. 1, pp. 72–99, Apr. 2006.

[9] A. Ghasemi, “Predictive modeling of LTE user throughput via crowd-sourced mobile spectrum data,” in Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN), Oct. 2018, pp. 1–5.

[10] M. Grabisch, “K-order additive discrete fuzzy measures and their representation,” Fuzzy Sets Syst., vol. 92, no. 2, pp. 167–189, Dec. 1997.

[11] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Jan. 2003.

[12] C. Hu, B. Murch, A. A. Corcoran, L. Zheng, B. B. Barnes, R. H. Weisberg, K. Atwood, and J. M. Lenes, “Developing a smart semantic Web with linked data and models for near-real-time monitoring of red tides in the eastern gulf of Mexico,” IEEE Syst. J., vol. 10, no. 3, pp. 1282–1290, Sep. 2016.

[13] S. Khalid, T. Khalil, and S. Nasreen, “A survey of feature selection and feature extraction techniques in machine learning,” in Proc. Sci. Inf. Conf., Aug. 2014, pp. 372–378.

[14] S. Lipovetsky and M. Conklin, “Analysis of regression in game theory,” Fuzzy Sets Syst., vol. 123, no. 1, pp. 21–35, Apr. 2001.

[15] C. C. F. Chu, D. P. K. Chan: Feature Selection Using Approximated High-Order Interaction Components

\[ M + \sum_{i=1}^{K-1} i \times (M - i) \]  \(\text{for } K = M\)  \(\text{(14)}\)

The number of attributes to be evaluated is greatly reduced. For example, when the total number of input attributes is 20 and the goal is to look for the best attribute subset with size less than or equal to 5 (i.e., \(M = 20\) and \(K = 5\)), the exhaustive search method will require the evaluation of 21,699 subsets, whereas the ISV search method takes 190.

Remark: When the number of selected attributes equals the size of the input attributes, the last element in the list should be added without the need for evaluation. The total number of subset evaluations will be equal to:

\[ M + \sum_{i=1}^{K-2} i \times (M - i) \]  \(\text{for } K = M\)  \(\text{(14)}\)
[16] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Proc. Adv. Neural Inf. Process. Syst. (NIPS), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 4765–4774. [Online]. Available: http://www.proceedings.com/info.html
[17] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent individualized feature attribution for tree ensembles,” 2018, arXiv:1802.03888. [Online]. Available: http://arxiv.org/abs/1802.03888
[18] F. Mokdad, D. Bouchaffra, N. Zerrouki, and A. Touazi, “Determination of an optimal feature selection method based on maximum shapley value,” in Proc. 15th Int. Conf. Intell. Syst. Design Appl. (ISDA), Dec. 2015, pp. 116–121.
[19] C. Molnar. (2020). Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Accessed: Mar. 25, 2020. [Online]. Available: https://christophm.github.io/interpretable-ml-book/
[20] S. Murofushi and T. Soneda, “Techniques for reading fuzzy measures (III): Interaction index,” in Proc. 9th Fuzzy Syst. Symp., May 1993, pp. 693–696.
[21] F. Pan, T. Converse, D. Ahn, F. Salvetti, and G. Donato, “Feature selection for ranking using boosted trees.” in Proc. 18th ACM Conf. Inf. Knowl. Manage. CIKM, 2009, pp. 2025–2028.
[22] A. B. Parsa, A. Movahedi, H. Taghipour, S. Derrible, and A. Mohammadian, “Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis,” Accident Anal. Prevention, vol. 136, Mar. 2020, Art. no. 105405.
[23] J. R. Quinlan, “Induction of decision trees,” Mach. Learn., vol. 1, no. 1, pp. 81–106, 1986.
[24] L. S. Shapley, “A value for n-person games,” Ann. Math. Stud., vol. 28, pp. 307–317, 1953.
[25] E. Štrumbelj and I. Kononenko, “Explaining prediction models and individual predictions with feature contributions,” Knowl. Inf. Syst., vol. 41, no. 3, pp. 647–665, Dec. 2014.
[26] H. Tang, J. Alelyani, and S. Liu, “Feature selection for classification: A review,” in Data Classification: Algorithms and Applications, Boca Raton, FL, USA: CRC Press, 2014, pp. 37–64.
[27] E. Tuv, A. Borisov, G. Runger, and K. Torkkola, “Feature selection with ensembles, artificial variables, and redundancy elimination,” J. Mach. Learn. Res., vol. 10, pp. 1341–1366, Jul. 2009.
[28] S. Visalakshi and V. Radha, “A literature review of feature selection techniques and applications: Review of feature selection in data mining,” in Proc. IEEE Int. Conf. Comput. Intell. Comput. Res., Dec. 2014, pp. 1–6.
[29] R. H. Weisberg, L. Zheng, Y. Liu, A. A. Corcoran, C. Lembke, C. Hu, J. M. Lenes, and J. J. Walsh, “Karenia brevis blooms on the west florida shelf: A comparative study of the robust 2012 Bloom and the nearly null 2013 event,” Continental Shelf Res., vol. 120, pp. 106–121, Jun. 2016.
[30] M. Zaeri-Amirani, F. Afgah, and S. Mousavi, “A feature selection method based on shapley value to false alarm reduction in ICUs a genetic-algorithm approach,” in Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), Jul. 2018, pp. 319–323.