IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 25, 2021, accepted January 28, 2021, date of publication February 12, 2021, date of current version February 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059298

A Hybrid Feature Selection Method RFSTL for Manufacturing Quality Prediction Based on a High Dimensional Imbalanced Dataset

HONG ZHOU[1,3], KUN-MING YU[3], YEN-CHIU CHEN[2], AND HUAN-PO HSU[3]

[1]Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China
[2]Department of Information Management, Chung Hua University, Hsinchu 30012, Taiwan
[3]Ph.D. Program in Engineering Science, College of Engineering, Chung Hua University, Hsinchu 30012, Taiwan

Corresponding author: Kun-Ming Yu (yu@chu.edu.tw)

This research was supported in part by the Ministry of Science and Technology of R.O.C. under contract MOST 105-2221-E-216-015-MY2.

ABSTRACT Under Industry 4.0, manufacturing quality prediction has been gaining increased interest from researchers and manufacturers. From the analysis of previous studies on quality predictions using machine learning, it became clear that the high dimensionality and imbalance of data are major and common problems affecting the learning performance. This work uses a hybrid method to address this issue, applying a Synthetic Minority Oversampling Technique & TomekLinks balancing approach to create balanced data and using Random Forest as the feature selecting measurement to reduce the dimensionality of data. In addition, a Fine Gaussian Support Vector Machine (Fine Gaussian SVM) based on the representative set of features selected by the hybrid method utilized is employed in this work to predict product quality. The results of experimentation demonstrate that the hybrid method proposed in this work performs well for manufacturing quality prediction and offers a simple, quick and powerful way to address the problem of feature selection encountered by the imbalanced classification.

INDEX TERMS Imbalanced data, feature selection, quality prediction, Industry 4.0, random forest, fine Gaussian SVM.

I. INTRODUCTION

With the advent of Industry 4.0, also referred to as the fourth industrial revolution, smart factory and manufacturing has become a new trend that seems to be the future for industrial development. As a result, advances in the Internet of Things (IoT), Big Data, Cloud, Artificial Intelligence (AI) and other technology are impelling Industry 4.0 to become a reality. By processing massive amounts of manufacturing data (internal and external) and leveraging AI technology it is possible to enable intelligent and quick decision-making for manufacturers to enhance product quality, increased yield, and reduced cost.

In contrast to traditional quality prediction, which relies on professional or statistical analysis, AI technology offers an advanced approach and superior performance due to its self-learning ability without having to consider manufacturing processes. Due to the characteristics of the modern manufacturing process, manufacturing quality data are generally complex, high-dimensional, and skewed, which causes some problems for constructing an efficient quality prediction model.

To address the imbalanced classification issue, some researchers prefer to resolve from the perspective of a data set adopting a resampling algorithm [1 $\sim$ 3], while others prefer to design new algorithms or improve upon existing algorithms [4 $\sim$ 6]. On the other hand, for the issue of high-dimensional features, feature selection [7] or feature extraction [8] are mostly applied.

This study attempts to investigate the dimension reduction issue through feature selection algorithms based on the imbalanced data, taking the manufacturing quality prediction as an application example. A hybrid algorithm RFSTL, is proposed based on the SMOTE&Tomek links algorithm for balancing data, and Random Forest for feature selection. By this way, the imbalanced and high dimension issues are solved in the data and feature processing stage, before model learning. Furthermore, the Fine Gaussian SVM was

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

simulated to predict manufacturing quality as a case study. These experiments demonstrate that, assisted by the RFSTL algorithm, the conventional classification algorithms can work effectively in the case of an imbalanced dataset with a high dimension.

In this paper, Section 2 introduces work related to feature selection, Section 3 illustrates the methodology being used, Section 4 describes the simulation and experiments, and Section 5 concludes the simulation results of this study.

II. RELATED WORK

A. IMBALANCED DATA RECONSTRUCTION

Imbalanced data means there is a disparity in the number of different classes in the dataset. If one dataset exhibits an unequal distribution between classes, one class (majority class) outnumbering the other class (minority class) by a large proportion, it will be considered as being imbalanced data implying imbalanced classification.

Imbalanced classification [9] is a learning problem that results in classification performance deterioration causing biased predictions for the majority class and would generally be misleading. The fundamental reason [10] is that traditional machine learning algorithms are accuracy driven, which assumes the data set to be balanced and aims at minimizing the overall error obtained from different classes with the same cost.

Two measurements [11] are commonly employed to solve the imbalanced classification problem: one is to reconstruct imbalanced data and the other is to design a proprietary algorithm [12]. Compared to the algorithm design [13], data reconstruction is simpler and more direct. Furthermore, the core mechanism of data reconstruction is to alter the class distributions by resampling the data, which can be divided into three categories:

1. Undersampling

The undersampling (downsampling) [14] method aims to balance the data set by reducing the number of observations from the majority class. Typical undersampling algorithms are introduced below.

- Random undersampling

The random undersampling method is a non-heuristic method based on randomly selecting observations from the majority class. It cannot make full use of existing information and may throw out the potentially useful information pertaining to the majority class.

- Tomek Links

The Tomek Links algorithm [15] removes the observations according to a pre-specified criterion by deleting samples belonging to the majority class in Tomek Links or removing them all.

- Informed undersampling

The Informed undersampling algorithm can solve the problem of data loss caused by traditional random undersampling. The best known informed undersampling algorithms are the EasyEnsemble algorithm and the BalanceCascade algorithm [16].

2. Oversampling

The oversampling (upsampling) [17] method works by replicating observations from the minority class to obtain balanced data. The prominent disadvantage of oversampling is the likelihood that overfitting may be increased due to the extra copies of the minority class examples that are created. Typical oversampling algorithms are introduced below:

- Random oversampling

The random oversampling method is designed to add replicated observations of the minority class normally, which can lead to the information learned by the model to be too specific, not general, and causing overfitting.

- SMOTE

The Synthetic Minority Over-sampling Technique (SMOTE) [18] generates artificial data and adds observations for the minority class based on similarities of feature space. First, it selects a random sample from its nearest neighbor for each minority sample. Then it chooses a random point along the line segment connecting these two samples as the new synthetic sample.

Several methods have been developed to improve the original SMOTE algorithm such as borderline-SMOTE1 and borderline-SMOTE2 [19], as well as Synthetic Minority Over-sampling Technique Nominal (SMOTE-N) and Synthetic Minority Over-sampling Technique Nominal Continuous (SMOTE-NC).

3. Combined sampling

The SMOTE&TomekLinks method [20] has been proposed for study on the protein classification in bio-informatics using a decision tree where the SMOTE&TomekLinks method performs well for imbalanced data.

B. FEATURE SELECTION

Feature selection, a frequently-used measurement to deal with high dimensionality, can decrease the data dimensionality by removing irrelevant features from the raw data. It automatically selects features from the raw data (input of one machine learning model) that contribute most to the prediction variable (output of this machine learning model). In this way, three benefits can be obtained for modeling: shortening the training time, reducing the risk of overfitting, and improving the performance of the model. According to the different learning algorithms, feature selection methods can be primarily divided into three categories:

1. Filter methods for feature selection

Filter methods [21] for feature selection act as a pre-processing step that independently selects the feature subset and runs quickly. Some widely applied filters for feature selection are methods based on correlation coefficients, methods using Fisher ratio, methods based on ReliefF, methods utilizing minimum redundancy maximal relevance and so on.

- Feature Selection based on Correlation Coefficient

Feature selection based on correlation coefficient [22], a simple filter algorithm, ranks feature subsets according to an evaluation formula measuring the correlation coefficient computed by the linear correlation coefficient, Pearson's correlation coefficient, etc.. In addition, a heuristic search strategy is utilized to identify relevant features that are highly correlated to the target.

- Feature Selection based on Fisher Ratio

Fisher Ratio (FR) scores are estimated to identify the subset of features in the data space spanned by those features. The distance between data points in the same class should be as small as possible, while the distance between data points in different classes should be as large as possible.

- Feature Selection based on ReliefF

Feature selection based on ReliefF [23] is an extension method of the original Relief algorithm [24]. In contrast to the original Relief method designed for binary classification problems, ReliefF has no limitations on the number of classes, and is capable of dealing with incomplete and noisy data. On the other hand, similar to Relief, ReliefF is aware of the contextual information and can correctly estimate the quality of attributes in problems with strong dependencies between attributes.

- Feature Selection based on Minimum Redundancy and Maximum Relevance

Minimum Redundancy and Maximum Relevance (mRMR) [25], [26] measures the similarity between features and targets according to the mutual information and aims to select a subset of features where each feature has the maximum relevance between the feature and the target, as well as the minimum redundancy among the rest of the features in the subset.

2. Wrapper methods for feature selection

Wrapper methods for feature selection [27] work like a black box. Regardless of the chosen learning machine, they can assess and score features depending on the prediction performance of a given learning machine to find the relative, useful variables. Some popular wrappers for feature selection are methods based on each single feature's prediction accuracy, methods adopting sequential forward selection, methods applying the genetic algorithm, and so on.

- Sequential Forward Feature Selection

The Sequential Forward Selection (SFS) algorithm is one of the heuristic algorithms commonly used to choose the representable features by adopting the methodology of K-Nearest Neighbor as classifier, and the approach of leave-one-out as the recognition rate estimate method. It sequentially selects a feature as the candidate feature until the addition of further features does not decrease the criterion.

- Feature Selection based on Genetic Algorithm

Genetic algorithms (GAs) [28], a form of inductive learning strategy, are employed for feature selection which is treated as a combinatorial optimization problem. Yang and Honavar [29] has demonstrated that searching the "optimal" features for targets by Genetic algorithm is a substantial improvement on a variety of random and local search methods.

3. Embedded methods for feature selection

Embedded methods for feature selection have recently been proposed as competition to filters and wrappers. They perform in the process of training and are usually specific to the given learning machines. Some typical embedded methods for feature selection are methods using random forest, methods based on stepwise fitting, methods applying SVM-recursive feature elimination (SVM-RFE), such as the original linear version of that, or the kernel version, and so on.

- Feature Selection based on Random Forest

The Random Forest (RF) [30] is an ensemble predictor consisting of numerous weak classifiers (decision trees). It can be used to measure the importance of an attribute as a straightforward method for feature selection. It has been successfully applied to do feature selection for high dimensional data, arising from microarrays [31], time series [32], and even on spectra [33]. The most widely used measures to calculate the score of importance for feature ranking are the mean decrease accuracy (MDA) and the mean decrease Gini (MDG). MDA quantifies the importance of a feature by measuring the change in prediction accuracy when values of features are randomly permuted and compared to the original observations. MDG adds up all decreases in Gini impurity due to the given feature which forms a split in the Random Forest.

- Feature Selection based on Stepwise Fitting

Feature selection based on stepwise fitting, in contrast to the general sequential feature selection, normally makes use of optimizations that are only possible with least-squares criteria.

- SVM-recursive feature elimination

SVM-recursive feature elimination (SVM-RFE) [34] is a feature selection algorithm based on SVM that calculates weights of features, removes features with the lowest weights iteratively and ranks features' importance according to the removing sequence. The linear SVM-RFE [35] is applied to select critical features by overall ranking or class-specific ranking. Moreover, the Gaussian kernel SVM-RFE performs better than the linear SVM-RFE when dealing with complex and nonlinear issues.

In addition, there are many hybrid methods that comprehensively apply multiple feature selection algorithms belonging to the different types mentioned above. For instance, the hybrid feature selection of combining filters and wrappers is applied in some of the researched methods [36], [37].

C. CLASSIFICATION BY SUPPORT VECTOR MACHINE

Support vector machine (SVM), first introduced by Vapnik [38], is a successful modeling technique based on machine learning for classification (especially the binary classification) and regression. Because of the Structural Risk

Minimization (SRM) principle [39], SVM has a powerful ability to handle overfitting by minimizing the upper limit of error generalization. In addition, SVM can be classified in general categories due to the kernel functions adopted, for example, the linear, Radial Basis function, Polynomial function, and Gaussian functions.

The Fine Gaussian SVM utilizes the Gaussian function as a kernel and has a high model flexibility to decrease scale setting. It can make finely detailed distinctions between classes, with kernel scale set to $\sqrt{P}/4$ where P is number of predictors.

III. METHODOLOGY

Due to the reality of manufacturing under Industry 4.0, the production data collected from manufacturing generally have a high dimension and the product data, identifying the defective product and qualified product, have a skewed class distribution. These two issues have been considered and resolved through two different approaches: feature selection and data resampling.

In this paper, the Random Forest algorithm is used to evaluate the importance of each feature, and on this basis, select feature variables from the original data satisfying the following conditions: (1) highly correlated with the dependent variable; (2) fully predict the results of the dependent variables with a small number. In this way, the dimensionality of feature space can be reduced and the performance of the algorithm or the model can be further improved.

In this paper, a combination of resampling methods, SMOTE&TomeLinks, is used to solve the classification problem of imbalanced data sets and generate reasonably balanced data. This is done so that the standard machine learning classification algorithm can be directly and successfully implemented on the generated data set, without the need to design an exclusive imbalanced classification algorithm.

Therefore, a hybrid method, RFSTL (Random Forest combined with SMOTE&TomekLinks), is proposed for doing feature selection based on imbalanced data. This method implements data resampling using the SMOTE&TomekLinks method and uses Random Forest for feature selection. The flowchart of RFSTL is shown in Fig 1.

As is shown in Fig. 1, there are two crucial modules in the RFSTL method, one is for generating balanced data and another is for selecting optimal features for classification.

A. DATA BALANCING

Problems often criticized in the single use of over-sampling or under-sampling methods are:

1. Deviation: sampling may change the distribution of initial data which will lead to deviation. Most oversampling methods will make the variance of variables appear smaller than it actually is, while under-sampling will make the variance of independent variables appear higher than it actually is.

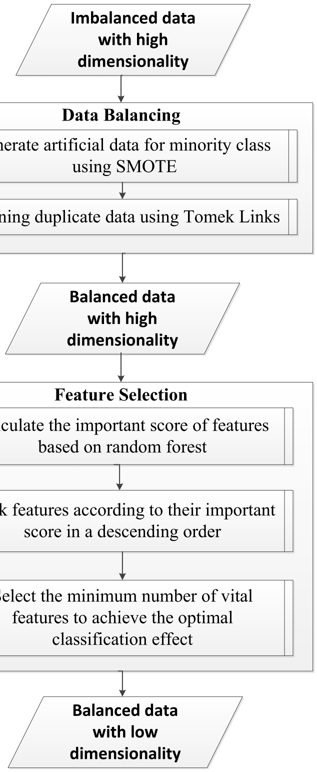
Imbalanced data with high dimensionality
Data Balancing
Generate artificial data for minority class using SMOTE
Cleaning duplicate data using Tomek Links
Balanced data with high dimensionality
Feature Selection
Calculate the important score of features based on random forest
Rank features according to their important score in a descending order
Select the minimum number of vital features to achieve the optimal classification effect
Balanced data with low dimensionality

FIGURE 1. Flowchart of the method RFSTL.

2. Overfitting: it is difficult to increase the information contained in the data by oversampling to increase the amount of data of a small number of samples, which is easy to cause overfitting of the model.

3. Overlapping: when there is too much overlapping of data, especially the noise, the classification effect may become worse because the noise is also repeatedly used.

Due to the respective defects of oversampling and undersampling, this work combines these two methods to ensure the data's quality after resampling,and to guarantee that the data points are as diverse as possible (non or less overlapping) in data sets, in order to avoid or solve problems listed above. Therefore, the main steps for implementing the data balancing module of the RFSTL method are illustrated below:

1. Create artificial samples for minority class by SMOTE utilizing the similarity between the existing samples belonging to the minority class in the feature space based on bootstrapping and k-nearest neighbors.

- Let S_{min} be the subset of samples for minority class and let S_{maj} be the subset of samples for the majority class.
- Identify the K-nearest neighbor based on Euclidean Distance for each observation x_i belongs to S_{min}.
- Randomly choose a few neighbors, the number of neighbors depends on the rate of over-sampling.
- Generate the synthetic samples for minority class and spread them along the line joining the K-nearest neighbor to its nearest neighbors.

The algorithm to generate artificial samples based on SMOTE is illustrated as follows:

Algorithm 1 Generate Artificial Samples

```
  Input: $M$; $O$; $n\backslash$
/*$M$: number of samples belong to the minority class; $O$: amount of SMOTE $O$%; $n$: number of nearest neighbors*/
  Output: S_synthetic
// S_synthetic: synthetic samples belong to the minority class
  Process:
if $O < 100$ then
Randomize the $M$ samples belong to the minority class
    $M = M*(O/100)$
    $O = 100$
end if
Compute: $O$ = (int) ($O/100$)
Create: S_minority[][]
//array for samples of the original minority class
Create: S_synthetic[][]
//array for synthetic samples
Create: nn[][]
//array for nearest neighbors
Define: $na$ = number of attributes
Define: new_index = 0
/*number of synthetic samples generated and initialized to 0*/
Define: $i$ = index of original samples
for $i \leftarrow 1$ to $M$ do
    while $O \neq 0$ do
    Define: index = random(1,$n$)
    // a random number between 1 and $n$
    Compute: nn_index = nn[index]
    //index for one of $n$ nearest neighbors of $i$
        for $j \leftarrow 1$ to $na$ do
        Compute:
  diff = S_minority [nn_index] [j] − S_minority [i] [j]
        Define: gap = random(0, 1)
        // a random number between 0 and 1
      Compute:
        S_synthetic[new_index][j]= S_minority [i] [j]
                                          +gap*diff
        end for
    new_index ++
    O −−
    end while
end for
return S_synthetic
```

The SMOTE algorithm is blind in the selection of nearest neighbors. As can be seen from the above algorithm process, it is necessary in the process of algorithm execution to determine the n value, that is, how many nearest neighbor samples to select. This must be solved by the users themselves. From the definition of n-value, we can see that the lower limit of n-value is the number of neighbor samples randomly selected from k nearest neighbors, which can be determined by the number of negative samples, the number of positive samples and the final balance rate of the data set. However, there is no way to determine the upper limit of n value, so we can only test repeatedly or use the empirical value according to the specific data set.

2. Cleaning duplicate instances from the sampling data based on Tomek Links.

- Definition:

Tomek links (T-Links) are defined as connections between a pair of nearest-neighbor samples from the opposite-class. Given an instance pair: $d(x_i, x_j)$, where x_i is an instance of minority class I and x_j is an instance of majority class J which belongs to different classes.

- Criteria:

If for any instance x_k, $d(x_i, x_j) < d(x_i, x_k)$ or $d(x_i, x_j) < d(x_j, x_k)$, then the (x_i, x_j) will be denoted as a T-Links. Furthermore, if any two samples constitute a T-Links then one of them will be either a noise or both will be located on or close to the boundary of classes.

The Tomek links are utilized in this work to remove unwanted duplicate samples after synthetic sampling by SMOTE. Two different operations are performed on the sample pair in Tomek links:

1. Under-sampling: If the sample pair contains the minority class sample of the original imbalanced data set, then the sample belonging to the majority class in the pair is eliminated.

2. Data cleaning: If the sample pair does not contain the minority class samples of the original imbalanced data set, then remove both samples in the pair.

The algorithm to clean duplicate instances from the sampling data based on Tomek Links is illustrated as follow:

B. FEATURE SELECTION

The ordinary Random Forests, an ensemble of CART (Classification and Regression Tree) Decision Trees, is constructed to measure the feature importance score and select the most predictive features for classification. The main steps are illustrated as follow:

1. Generate the Random Forest to maximum size and do not prune.

First, CART, acting like an individual learner, is combined to form the Random Forest [40]. In each CART, incoming observations are divided and sent from the root, or the parent node, to their child node according to a specific split rule. This recursive partitioning process aims at purifying the incoming observations to ensure that it only contains one class of observation.

2. Measure Feature importance based on the Mean Decrease in Gini Impurity (MDG)

As a general indicator of feature relevance, the Mean Decrease in Gini Impurity (MDG) [41] can be adopted to measure the feature importance. The MDG for a given feature is the average of the difference between the Gini Impurity

Algorithm 2 Clean Duplicate Instances

```
Input: S_SMOTE,S
/* S_SMOTE: a balanced data set obtained by the algorithm SMOTE; S: original imbalanced data set*/
Output: S_STL
/* S_STL: a balanced data set obtained by the algorithm SMOTE removing duplicate instances contained in the Tomek_Links. */
Process:
Create: \ S_minority,SS_minority
/*S_minority: array store instances of the minority class in S; SS_minority: array store instances of the minority class in S_SMOTE*/
Create: SS_majority
/*array store instances of the majority class in S_SMOTE*/
Create: Distance
/*array store distances between each instances from SS_minority, SS_majority, and S_SMOTE denoted as d ($x_i$, $x_j$), d ($x_i$, $x_j$), and d ($x_i$, $x_k$)*/
for each instance $x_i$ in SS_minority, $x_j$ in SS_majority do
    Compute: d ($x_i$, $x_j$)
    Store d ($x_i$, $x_j$) in Distance
end for
for each instance $x_i$ in SS_minority, $x_k$ in S_ SMOTE do
    Compute: d ($x_i$, $x_k$)
    Store d ($x_i$, $x_k$) in Distance
end for
for each instance $x_j$ in SS_majority, $x_k$ in S_ SMOTE do
    Compute: d ($x_j$, $x_k$)
    Store d ($x_j$, $x_k$) in Distance
end for
for all instances in the Distance do
    if d ($x_i$, $x_j$) < d ($x_i$, $x_k$) or d ($x_i$, $x_j$) < d ($x_j$, $x_k$) then
        if instance $x_i$ is\in S_minority then
            Delete the $x_j$ from S_majority.
        else
            Delete the $x_i$ from S_minority and the $x_j$ from S_majority.
        end if
    end if
end for
Create: S_STL
Store S_minority and S_majority into S_STL
return S_STL
```

of the parent node and the Gini Impurity of the child nodes over from all trees in the forest for that feature. The higher the MDG value of a given feature, the more important that feature is, making it more effective.

The vital indexes of the Mean Decrease in Gini Impurity (MDG) to calculate the feature's importance score are described as follow:

- Gini index

If the incoming data set of node v contains samples from c classes, the Gini index $Gini$ (v) for the node impurity is defined as the Formula 3.1.

$$Gini(v) = 1 - \sum_{j=1}^{c} p_j^2 \tag{3.1}$$

p_j represents the proportion (relative frequency) of class j (value from 1 to c where c is the number of class) in the incoming data set of node v. $Gini$ (v) is minimized if the classes in j are skewed.

If the node v is split into two child nodes v_l (left child) and v_r (right child), whose sample size are denoted as $D(v_l)$ and $D(v_r)$, the Gini index of the split data is defined as the Formula 3.2.

$$Gini_{splitted}(v) = \frac{D(v_l)}{D(v)}Gini(v_l) + \frac{D(v_r)}{D(v)}Gini(v_r) \tag{3.2}$$

The feature providing smallest $Gini_{\text{splitted}}(v)$ is chosen to split the node.

- Decrease in Gini Impurity

Decrease in Gini Impurity is measured by Formula 3.3, which identifies the change in impurity of a node caused by data splitting. A high Decrease in Gini Impurity implies that the node and its corresponding feature are valuable to separate the classes.

$$\Delta Gini(v) = Gini(v) - Gini_{splitted}(v) \tag{3.3}$$

- Importance score of features

Importance score of the feature X_j in a decision tree T_k is defined as the Formula 3.4.

$$IS(X_j) = \sum_{v \in T_k} \Delta Gini(v) \tag{3.4}$$

Importance score of the feature X_j overall K trees in a Random Forest are defined as in Formula 3.5.

$$IS(X_j) = \frac{1}{K}\sum_{k=1}^{k} IS(X_j) \tag{3.5}$$

C. STOPPING CRITERIA FOR FEATURE SELECTION

The stopping criteria adopted to determine when the feature quantity can reach the optimal number and stop the process of feature selection is defined as follows.

Step 1: Input N features according to the importance score measured in a descending order recursively to the classification model Fine Gaussian SVM, where $N \in \{1,2,\ldots,n\}$ and is increased from 1 to n (sum of all features):

Step 2: Evaluate the value of F1-Measure and Area Under the Curve (AUC) achieved by the classification respectively. Whenever the criteria 1 or 2 is satisfied the loop of feature selection will stop and features currently selected will be identified as the optimal subset of features for classification.

Criteria1: The maximum value of F1-Measure achieved by the minimum number of features. In other words, features selected are identified as the optimal subset when the highest value of F1-Measure occurs at the first time, before the value of F1 decreases, while the number of features selected increases.

Criteria2: The value of AUC is approximately equal to 1 at the first time.

The algorithm to perform feature selection based on Random Forest, is illustrated as follow:

Algorithm 3 Feature Selection Based on Random Forest

```
**Input:** $S = \{(X_i, Y_i)_{i=1}^N | X_i ini^M, Y_i \in \{1, 2, \ldots, c\}\}; K$
/*$S$: data set obtained after data balancing described in section 3 where $X_i$ represents predictor variables, $Y_i$ represents the class response feature, $N$ is the number of training samples and $M$ is the number of features; $K$ is the number of trees*/
**Output:** $VF$
// $VF$: the most valuable features selected.
for $k \leftarrow 1$ to $K$ do
    Draw a bagged subset of $S_k$ from $S$
    while (stopping criteria of tree construction is not satisfied) do
        Construct a CART $T_k$
    end while
end for
Combine all $K$ CART together to form a Random Forest RF
for $j \leftarrow 1$ to $N$ do
    Compute importance score of the feature $X_j$
    /*compute the importance score of the feature by Formula 3.5*/
end for
Rank features according to its importance score in a descending order and store them in SF
for $f \leftarrow 1$ to $M$ do
    Select $IF_i$ from $X_i$ according to top $f$ features from $SF$
    Generate new data set $S'$ by combine $IF_i$ with $Y_i$
        $S = \{(IF_i, Y_i)_{i=1}^N | IF_i \in i^f, Y_i \in \{1, 2, \ldots, c\}\}$
    Input $S'$ into the Fine Gaussian SVM for classification
    Evaluate F1 and AUC of the Fine Gaussian SVM
    while (stopping criteria of feature selection is satisfied) do
        Define: $VF$= top $f$ features from $SF$
    end while
end for
return $VF$
```

IV. SIMULATION AND EXPERIMENT RESULTS

To evaluate the performances of the hybrid method proposed in this study, a complete simulation plan was designed and performed taking manufacturing quality prediction as a research case. The flowchart of the simulation is shown as Fig.2.

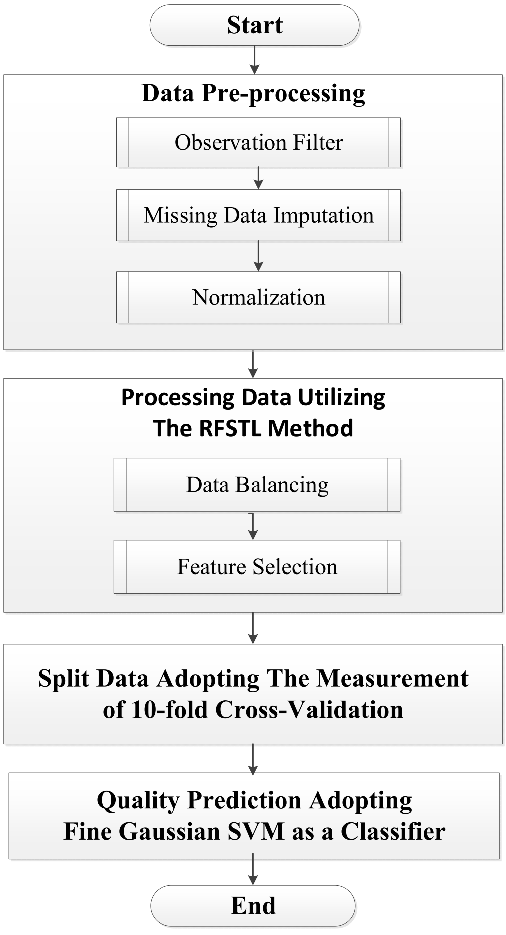
Start
Data Pre-processing
Observation Filter
Missing Data Imputation
Normalization
Processing Data Utilizing The RFSTL Method
Data Balancing
Feature Selection
Split Data Adopting The Measurement of 10-fold Cross-Validation
Quality Prediction Adopting Fine Gaussian SVM as a Classifier
End

FIGURE 2. Flowchart of Simulation performed in this study.

A. DATA PREPROCESSING

The data set (Secom) adopted in this work is achieved from the website of the University of California Irvine [42]. It contains 1567 instances, 590 features and 1 yield. There are a lot of missing data denoted as "NAN" in this data set, because it is collected during a real semiconductor manufacturing. Two different missing data processing strategies were proposed to generate a complete data set in order to prevent the adverse effects on the model performance.

1. Observations Filter

Calculate the percentage of the missing value for each instance and delete instances if its missing percentage is greater than 10%. Finally, 34 instances were deleted in which only 1 instance was included in the minority class (defective products) and 33 instances were included in the majority class (qualified products).

2. Missing data imputation

K Nearest Neighbor (KNN) algorithm [43] is implemented to generate a complete data set, described in Table 1. Taking into account that observations in this data set are collected in a short time interval; the nearest neighbor is selected according to the minimum Euclidean distance. In addition, the missing

TABLE 1. Description of data set for simulation.

Data set	Quantity of Samples	Samples in Majority Class	Samples in Minority Class	Feature Dimension
Secom[a]	1567	1463	104	590
Secom1[b]	1533	1430	103	590
Secom2[c]	2860	1430	1430	590
Secom3[d]	1533	1430	103	10
Secom4[e]	2860	1430	1430	10

[a] The raw data set.
[b] The data set after pre-processing.
[c] The data set after pre-processing and then data balancing by the SMOTE&Tomek method.
[d] T he data set after pre-processing and then be further processed by the feature selection method based on Random Forest.
[e] The data set after pre-processing and then be further processed by the RFSTL method.

value is estimated utilizing the weighted average value of the nearest neighbor.

3. Normalization

Since the unit of features in the data set was unknown, the measurement value of each feature was dramatically different. To eliminate the limitation of the data unit, the feature value was converted into the dimensionless number which facilitated the comparison and weighting of features in different units or magnitudes. This study specifically scaled the data in the data set and normalized the measured value of each feature to a value range from 1 to -1.

B. FEATURE SELECTION BASED ON AN IMBALANCED DATA USING THE RFSTL METHOD

The RFSTL method proposed in this study is simulated to select features from an imbalanced data set, and various feature selection methods are simulated for comparison.

1. Data Balancing

The data set after preprocessing, including 1430 instances for qualified yield (majority class) and 103 instances (minority class) for defective yield, shows a typical characteristic of between-class imbalance whose sample ratio of majority to minority is 14:1. The RFSTL method proposed in this study is adopted to achieve a balanced data set containing 2860 observations in total in which there are 1430 instances for the majority class and 1430 instances for the minority class, The sample ratio of majority to minority is 1:1, which is detailed in Table 1. The vital parameters for RFSTL are choosing a typical K value as 5 for regular SMOTE sampling and applying two different removing criteria for T-Links.

2. Feature Selection

To evaluate the RFSTL method, 10 feature selection methods are simulated adopting typical parameters for comparative study shown in Table 2.

All of the feature selection algorithms in Table 2 are simulated to select the most valuable 10 features from the data set Secom1 (imbalanced data set) and Secom2 (balanced data set) described in Table 1. In addition, the results of feature selection experiments on different data sets are listed in Table 3 and Table 4 respectively.

TABLE 2. Feature selection methods selected for simulation.

Feature Selection Method	Abbreviation
Feature selection based on Correlation Coefficients	CC
Feature ranking using Fisher ratio	Fisher
Feature selection utilizing the genetic algorithm	GA
A feature weighting algorithm	ReliefF
Feature ranking by random forest	RF
Feature selection using sequential forward selection	SFS
Feature ranking based on each single feature's prediction accuracy	Single
Feature selection based on stepwise fitting	SWF
The kernel version of SVM-recursive feature elimination (SVM-RFE)	SRK
The original linear version of SVM-recursive feature elimination (SVM-RFE).	SRO
Feature selection depend on the minimum redundancy maximal relevance	mRMR

TABLE 3. Results of feature selection based on the imbalanced data set Secom1.

Feature Selection Method	Execution Time[a] (sec)	The top 10 most valuable features (No.[b])
CC	0.02	60,104,511,349,432,435,431,436,22,437
Fisher	0.02	60,104,130,29,511,126,317,125,123,131
GA	37.38	1,4,6,13,17,19,20,21,27,30
mRMR	9.45	60,122,65,1,131,206,563,461,41,32
ReliefF	42.41	60,65,22,349,122,333,197,76,112,438
RF	10.97	60,65,66,104,140,133,413,332,268,469
SFS	176.5	1,2,3,4,5,6,7,349,8,9
Single	163.03	1,2,3,4,5,6,7,8,9,10
SWF	0.93	60,65,122,130,32,68,76,146,57,424
SRK	13.81	500,165,431,489,549,559,28,41,39,56
SRO	27.84	387,521,577,575,105,429,250,360,344,60

[a] The execution time refers to the time spent to rank all features and choose the top 10 most valuable features.
[b] The value range of the feature No. is from 1 to 590

As shown in Table 3, the top 10 valuable features chosen from the imbalanced data set Secom1 by the different feature selection algorithms listed in Table 2, are totally different from each other.

Moreover, the execution times of these feature selections are not the same except for the two feature selection methods based on correlation coefficients and Fisher ratio which have

TABLE 4. Results of feature selection based on the balanced data set Secom2.

Feature Selection Method	Execution Time[a] (sec)	The top 10 most valuable features (No.[b])
CC	0.38	60,104,130,29,125,511,122,124,123,128
Fisher	0.13	60,104,130,29,125,511,122,124,123,128
GA	104.18	1,2,5,7,9,11,12,14,15,19
mRMR	13.96	487,34,86,255,248,384,122,490,96,281
ReliefF	42.41	420,512,419,501,487,500,483,490,489,549
RF	20.46	60,487,96,248,478,122,104,125,279,131
SFS	176.5	60,66,156,28,29,474,349,16,296,350,
Single	43.52	60,104,478,511,206,342,80,426,418,128
SWF	12.65	60,65,122,130,32,68,76,146,57,424
SRK	126.55	60,29,122,130,112,447,487,86,420,32
SRO	27.84	60,205,65,349,215,156,360,494,341,201

[a] The execution time refers to the time spent to rank all features and choose the top 10 most valuable features.

[b] The value range of the feature No. is from 1 to 590.

the shortest execution time. The top 2 most time-consuming algorithms are the wrapper feature selection method based on each single feature's prediction accuracy and the method adopting sequential forward selection. The accuracy of feature selection methods based on the imbalanced data set Secom1 is discussed through simulation of the Fine Gaussian SVM as a classifier. The simulation result is listed in Table 6.

Analyzing Table 4, it can be summed up that the top valuable features chosen by feature selection algorithms above are different from each other, except for the method based on correlation coefficients and the method based on the Fisher ratio, which have the same results. However, their execution time is not the same.

Moreover, these two feature selection methods run faster than the other 9 methods. The top 3 most time-consuming algorithms are the feature selections utilizing the genetic algorithm, the feature selections using sequential forward selection, and the kernel version of SVM-recursive feature elimination (SVM-RFE). The accuracy of the feature selection methods based on the balanced data set Secom2 is discussed after simulation of the classifier, Fine Gaussian SVM. The classification performance is compared in Table 7.

From a comparison between the feature selection results (listed in Table 3 and Table 4) adopting the same method based on different data sets (one imbalanced, another balanced), it can be found that, excluding the feature selection method based on stepwise fitting, which obtained the same top 10 representable features regardless of whether the data set is balanced or not, all other feature selection methods achieved the top 10 representable features which had a significant change. Fig. 3 and Fig. 4 are two scatter plot images that illustrate the distinctive results obtained when doing feature selection for an imbalanced data set and a balanced data set. The top 2 features selected by the RF method based on Secom1 and Secom2 are shown in Fig. 3 and Fig. 4, with the blue points identifying instances for qualified products (labeled as -1), and the red points identifying instances for defective products (labeled as 1). It is not difficult to infer from the differences in the two figures that it is very important and necessary to design an effective feature selection method for an imbalanced data set.

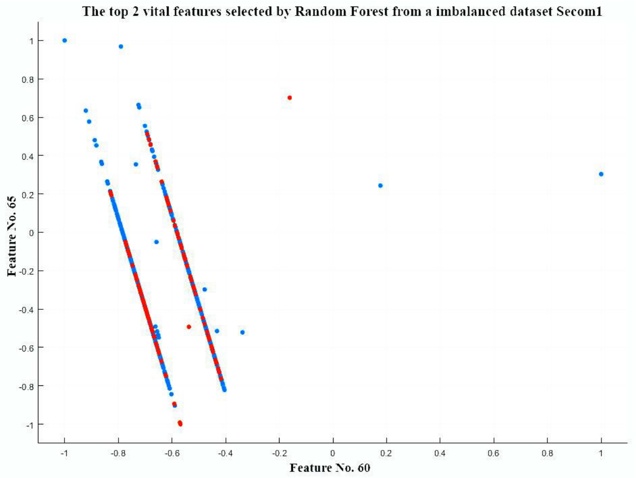

FIGURE 3. Scatter plot of the top 2 features selected by the method RF based on the imbalanced data set Secom1.

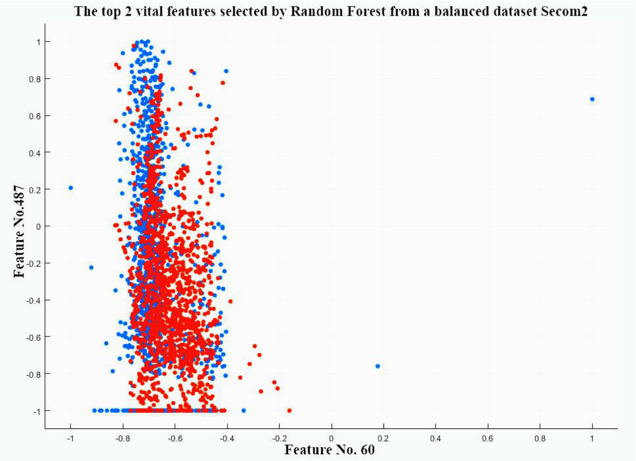

FIGURE 4. Scatter plot of the top 2 features selected by the method RF based on the balanced data set Secom2.

C. CLASSIFICATION USING FINE GAUSSIAN SVM

1. Divide Data for 10-fold Cross-validation

The 10-fold Cross-validation (CV) [44] method is applied to train the Fine Gaussian SVM model for evaluating the predictive performance of defective products. The original data set Secom is equally partitioned into 10 folds that k-1 folds are adopted for the training model and 1 fold left out is provided to test model. On these partitioned folds, training and testing as described above is executed in 10 iterations

TABLE 5. Performance metrics.

Metric	Abbreviation	Calculation Formulas
Accuracy	ACC	(TP+TN)/(TP+FP+TN+FN)[a]
False Positive Rate	FPR	FP/(TN+FP)
True Negative Rate (Specificity)	TNR	TN/(TN+FP)
Recall (Sensitivity)	REC	TP/(TP+FN)
Precision	PRE	TP/(TP+FP)
F1-Measure	F1	(Precision×Recall×2)/(Precision+Recall)
Area Under the Curve	AUC	$(\sum Rank(+) - \lvert + \rvert \times (\lvert + \rvert + 1)/2)/(\lvert + \rvert + \lvert - \rvert)$ [b,c,d]

[a] In all data sets used in this work, samples for defective products (labeled as 1) are defined to be the Positive. On the other hand, samples for qualified products (labeled as -1) are defined to be the Negative. True Positive (TP), the number of observations whose actual class is positive and predicted as a positive class. False Positive (FP), the number of observations whose actual class is negative, but predicted as a positive class. True Negative (TN), the number of observations whose actual class is negative and predicted as a negative class. False Negative (FN), the number of observations whose actual class is positive and predicted as a negative class.

[b] $\sum Rank(+)$ is the sum of the ranks of all positively classified samples.

[c] |+| is the number of positive samples in the data set.

[d] |+| is the number of negative samples in the data set.

TABLE 6. Prediction performance of the fine Gaussian SVM model using different predictors obtained from different feature selection algorithms on the same imbalanced data Secom1.

Input[a]	TP	FP	TN	FN	Accuracy	Precision	Recall	F1	AUC	Training Time
					(%)	(%)	(%)	(%)		(sec)
F10_CC	0	2	1428	103	93.2	0	0	-	0.63	1.45
F10_Fisher	0	0	1430	103	93.3	-	0	-	0.65	2.01
F10_GA	0	0	1430	103	93.3	-	0	-	0.49	1.57
F10_ mRMR	0	0	1430	103	93.3	-	0	-	0.61	1.89
F10_ReliefF	0	0	1430	103	93.3	-	0	-	0.66	1.88
F10_RF	0	2	1428	103	93.2	0	0	-	0.67	0.89
F10_ SFS	0	0	1430	103	93.3	-	0	-	0.54	1.60
F10_ Single	0	0	1430	103	93.3	-	0	-	0.51	1.62
F10_ SWF	0	1	1429	103	93.2	0	0	-	0.63	1.68
F10_ SRK	0	0	1430	103	93.3	-	0	-	0.54	1.62
F10_SRO	0	0	1430	103	93.3	-	0	-	0.58	1.41

[a] The name of input contains two parts, one part is noted as "FX" where "F" means "Feature" while "X" means the number of features and another part is the name or the abbreviation of the feature selection algorithm.

while the 1 fold selected for training changes one-by-one in each iteration until all partitioned folds have been chosen for testing once. The whole classification performance of the model is obtained by averaging the model performance of each iteration.

2. Construct a Fine Gaussian SVM model for classification

Fine Gaussian SVM is used for empirical simulation, which is an SVM that makes fine distinctions between classes with the help of the Gaussian kernel. The kernel scale is set to $\sqrt{}$P/4 where P is the number of predictors.

Core parameters of the Fine Gaussian SVM model are set as follows: Kernel function: Gaussian; Kernel scale: 0.79; Box constraint level: 1; Multiclass method: One-vs-One; Standardize data: true.

D. QUALITY PREDICTION

1. Performance metrics

The vital performance metrics applied in this work for classification based on the imbalanced data set are Precision, Recall, F1-Measure, Receiver Operating Characteristic curve (ROC curve.), and Area Under the Curve (AUC). Related calculation formulas are listed in Table 4.

- Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

- Recall

Recall (Sensitivity) is the ratio of correctly predicted positive observations to all observations in the actual class.

TABLE 7. Prediction performance of the Fine Gaussian SVM model using different predictors obtained from different feature selection algorithms on the same data Secom2.

Input[a]	TP	FP	TN	FN	Accuracy	Precision	Recall	F1	AUC	Training Time
					(%)	(%)	(%)	(%)		(sec)
F2_CC	928	268	1162	502	73.1	77.5	64.9	70.6	0.77	6.67
F4_CC	1196	262	1168	234	82.7	82.0	83.6	82.8	0.90	4.18
F6_CC	1254	134	1296	176	89.2	90.3	87.7	89.0	0.95	4.66
F8_CC	1289	105	1325	141	91.4	92.5	90.1	91.3	0.97	4.65
F10_CC	1312	74	1356	118	93.3	94.7	91.7	93.2	0.98	5.16
F11_CC	1331	66	1364	99	94.2	95.3	93.1	94.1	0.99	5.23
F12_CC	1338	60	1370	92	94.7	95.7	93.6	94.6	0.99	5.27
F13_CC	1339	36	1394	91	95.6	97.4	93.6	95.4	0.99	5.36
F14_CC	1338	28	1402	92	95.8	98.0	92.4	95.7	0.99	5.39
F15_CC	1322	14	1416	108	95.7	99.0	92.4	95.6	0.99	5.29
F16_CC	1313	7	1423	117	95.7	99.5	91.8	95.5	1	4.92
F2_Fisher	936	274	1156	494	73.1	77.4	65.5	70.9	0.77	9.67
F4_Fisher	1189	260	1170	241	82.5	82.1	83.1	82.6	0.89	5.17
F6_Fisher	1245	135	1295	185	88.8	90.2	87.0	88.6	0.95	4.67
F8_Fisher	1302	105	1325	128	91.9	92.5	91.0	91.8	0.97	4.66
F10_Fisher	1330	70	1360	100	92.7	95.0	93.0	94.0	0.98	5.16
F11_Fisher	1335	65	1365	95	94.4	95.4	93.4	94.3	0.99	5.21
F12_Fisher	1338	59	1371	92	94.7	95.8	93.6	94.7	0.99	5.33
F13_Fisher	1339	37	1393	91	95.5	97.3	93.6	95.4	0.99	5.24
F14_Fisher	1329	23	1407	101	95.7	98.3	92.9	95.5	0.99	5.40
F15_Fisher	1324	15	1415	106	95.8	98.9	92.6	95.6	0.99	5.32
F16_Fisher	1318	5	1425	112	95.9	99.6	92.2	95.8	1	4.86
F2_GA	1009	558	872	421	65.8	64.4	70.6	67.3	0.69	4.66
F4_GA	1129	521	909	301	71.3	68.4	79.0	73.3	0.78	4.16
F6_GA	1250	253	1177	180	84.9	83.2	87.4	85.2	0.92	5.18
F8_GA	1294	185	1245	136	88.8	87.5	90.5	89.0	0.95	4.66
F10_GA	1330	70	1360	100	94.1	95.0	93.0	94.0	0.99	5.18
F11_GA	1335	63	1367	95	94.5	95.5	93.4	94.4	0.99	5.25
F12_GA	1337	43	1387	93	95.2	96.9	93.5	95.2	0.99	5.30
F13_GA	1333	26	1404	97	95.7	98.1	93.2	95.6	0.99	5.40
F14_GA	1329	21	1409	101	95.7	98.4	92.9	95.6	0.99	8.05
F15_GA	1326	14	1416	104	95.9	99	92.7	95.7	1	5.44
F2_ mRMR	1220	556	874	210	73.2	68.7	85.3	76 .1	0.78	1.14
F4_ mRMR	1244	308	1122	186	82.7	80.2	87.0	83.4	0.89	3.90
F6_ mRMR	1308	203	1227	122	88.6	86.6	91.5	88.9	0.94	4.42
F8_ mRMR	1324	83	1347	106	93.4	94.1	92.6	93.3	0.98	5.26
F10_ mRMR	1328	27	1403	102	95.5	98.0	92.9	95.4	0.99	7.74
F11_ mRMR	1317	15	1415	113	95.5	98.9	92.1	95.4	0.99	6.22
F12_ mRMR	1301	5	1425	129	95.3	99.6	91.0	95.1	0.99	5.08
F2_ ReliefF	886	308	1122	544	70.2	74.2	62.0	67.5	0.75	6.16
F4_ ReliefF	1113	275	1155	317	79.3	80.2	77.8	79.0	0.85	4.65
F6_ ReliefF	1263	219	1211	167	86.5	85.2	88.3	86.7	0.93	5.19
F8_ ReliefF	1291	47	1383	139	93.5	96.5	90.3	93.3	0.98	5.16
F10_ReliefF	1269	17	1413	161	93.8	98.7	88.7	93.4	0.99	5.68
F11_ReliefF	1268	12	1418	162	93.9	99.1	88.7	93.6	0.99	5.50
F12_ReliefF	1264	4	1426	166	94.1	99.7	88.4	93.7	1	5.54
F2_ RF	1114	395	1035	316	75.1	73.8	77.9	75.8	0.81	5.66
F4_ RF	1263	226	1204	167	86.3	84.8	88.3	86.5	0.92	4.68
F6_ RF	1277	111	1319	153	90.8	92.0	89.3	90.6	0.96	4.67
F8_ RF	1307	78	1352	123	93	94.4	91.4	92.9	0.98	4.67
F10_RF	1341	33	1397	89	95.7	97.6	93.8	95.6	0.99	5.03

TABLE 7. ***(Continued.)*** **Prediction performance of the Fine Gaussian SVM model using different predictors obtained from different feature selection algorithms on the same data Secom2.**

F11_RF	1327	26	1404	103	95.5	98.1	92.8	95.4	0.99	5.23
F2_ SFS	911	204	1226	519	74.7	81.7	63.7	71.6	0.79	6.67
F4_ SFS	1013	203	1227	417	78.3	83.3	70.8	76.6	0.85	4.18
F6_ SFS	1218	208	1222	212	85.3	85.4	85.2	85.3	0.92	4.68
F8_ SFS	1281	153	1277	149	89.4	89.3	89.6	89.5	0.95	4.69
F10_ SFS	1313	84	1346	117	93	94.0	91.8	92.9	0.98	5.16
F11_ SFS	1328	48	1382	102	94.8	96.5	92.9	94.7	0.99	4.86
F12_ SFS	1337	46	1384	93	95.1	96.7	93.5	95.1	0.99	4.76
F13_ SFS	1333	51	1379	97	94.8	96.3	93.2	94.7	0.99	4.84
F2_ Single	939	284	1146	491	72.9	76.8	65.7	70.8	0.77	7.18
F4_ Single	1106	296	1134	324	78.3	78.9	77.3	78.1	0.85	4.17
F6_ Single	1115	307	1123	315	78.3	78.4	78.0	78.2	0.86	4.17
F8_ Single	1227	224	1206	203	85.1	84.6	85.8	85.2	0.93	4.14
F10_ Single	1314	133	1297	116	91.3	90.8	91.9	91.3	0.97	4.66
F11_ Single	1334	102	1328	96	93.1	92.9	93.3	93.1	0.98	4.86
F12_ Single	1337	76	1354	93	94.1	94.6	93.5	94.1	0.99	4.82
F13_ Single	1341	52	1378	89	95.1	96.3	93.8	95.0	0.99	4.94
F14_ Single	1340	38	1392	90	95.5	97.2	93.7	95.4	0.99	5.20
F15_ Single	1319	24	1406	111	95.3	98.2	92.2	95.1	0.99	5.33
F2_ SWF	928	177	1253	502	76.3	84.0	64.9	73.2	0.77	5.19
F4_ SWF	1233	239	1191	197	84.8	83.8	86.2	85.0	0.90	4.66
F6_ SWF	1249	197	1233	181	86.8	86.4	87.3	86.9	0.92	4.15
F8_ SWF	1320	111	1319	110	92.3	92.2	92.3	92.3	0.97	4.68
F10_ SWF	1332	46	1384	98	95	96.7	93.1	94.9	0.99	5.18
F11_ SWF	1333	31	1399	97	95.5	97.7	93.2	95.4	0.99	5.23
F12_ SWF	1354	24	1406	76	96.5	98.3	94.7	96.4	0.99	5.26
F13_ SWF	1338	25	1405	92	95.9	98.2	93.6	95.8	0.99	5.23
F2_ SRK	1054	419	1011	376	72.2	71.6	73.7	72.6	0.79	4.66
F4_ SRK	1250	236	1194	180	85.5	84.1	87.4	85.7	0.91	4.16
F6_ SRK	1327	103	1327	103	92.8	92.8	92.8	92.8	0.98	5.16
F8_ SRK	1332	29	1401	98	95.6	97.9	93.1	95.4	0.99	5.17
F10_ SRK	1323	14	1416	107	95.8	99.0	92.5	95.6	1	5.14
F2_ SRO	886	317	1113	544	69.9	73.6	62.0	67.3	0.72	9.65
F4_ SRO	1046	226	1204	384	78.7	82.2	73.1	77.4	0.85	6.17
F6_ SRO	1112	166	1264	318	83.1	87.0	77.8	82.1	0.90	4.15
F8_ SRO	1221	159	1271	209	87.1	88.5	85.4	86.9	0.94	4.15
F10_SRO	1265	149	1281	165	89	89.5	88.5	89.0	0.96	4.17
F12_SRO	1320	125	1305	110	91.8	91.3	92.3	91.8	0.97	4.28
F14_SRO	1327	104	1328	103	92.8	92.7	92.8	92.8	0.98	4.12
F16_SRO	1342	69	1361	88	94.5	95.1	93.8	94.5	0.99	4.32
F18_SRO	1351	40	1390	79	95.8	97.1	94.5	95.8	0.99	4.92
F19_SRO	1351	24	1406	79	96.4	98.3	94.5	96.3	0.99	4.94
F20_SRO	1352	19	1411	78	96.6	98.6	94.5	96.5	0.99	5.06
F21_SRO	1346	7	1423	84	96.8	99.5	94.1	96.7	0.99	5.16
F22_SRO	1356	10	1420	74	97.1	99.3	94.8	97.0	1	5.16

[a] The name of input contains two parts, one part is noted as "FX" where "F" means "Feature" while "X" means the number of features and another part is the name or the abbreviation of the feature selection algorithm.

- F1-Measure

F1 measure (F1 Score) is the weighted average of Precision and Recall, which is usually more useful than accuracy, especially for an imbalanced classification.

- The Receiver Operating Characteristic curve (ROC curve)

The ROC curve [36], a plot shows the True Positive Rate (Recall or Sensitivity) against the False Positive Rate (FPR) for different cut-off points of a diagnostic test. In the ROC curve, an increase of Sensitivity is often accompanied by a decrease of Recall. A curve in the ROC space higher than the 45-degree diagonal line identifies the result as meaningful. The closer the curve is to the left vertex, the more accurate the result is.

- Area Under the Curve (AUC)

The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two classes. A larger AUC value demonstrates a better classification performance. The AUC value of 1 means a perfect classification effect; the AUC value of 0.5 indicates the classification effect to be equivalent to a random guess.

2. Performance evaluation

- Performance evaluation of manufacturing quality prediction applies the same model inputting of different data achieved by different feature selection algorithms on imbalanced data set Secom1

The performance of classification using Fine Gaussian SVM, which inputs different subsets of features achieved by various feature selection methods from the imbalanced data set Seom1, are listed in Table. 6. The results of the experiment indicate that these feature selection algorithms cannot perform effectively on the imbalanced data set which represents an obviously biased classification prone to the majority class. Almost all examples in the minority class are mistakenly classified into the majority class and this situation appears commonly for all feature selection listed in Table 6. The classification results appear to be a slight change when applying the feature selection methods based on the correlation coefficients and random forest respectively. Two examples for the majority class are classified into the minority class by mistake. Unfortunately, this is not what we need at all.

- Performance evaluation of manufacturing quality prediction applying the same model inputting different data achieved by different feature selection algorithms on balanced data set Secom2

The classification performance using Fine Gaussian SVM is listed in Table 2. Inputs with different feature dimensions are applied which are obtained by various feature selection methods based on the same balanced data set achieved by resampling method SMOTE&TomekLinks. The experiment's results illustrated that these feature selection algorithms performed well on the balanced data set and achieved dimensionality reduction. With the increase in the number of selected features, manufacturing quality prediction improved at first, but deteriorated when the number of features exceeded a certain limit. These feature selection algorithms achieved the best classification before the number of selected features was increased to a certain point, which is the optimal feature dimension for classification (detailed in Table 8).

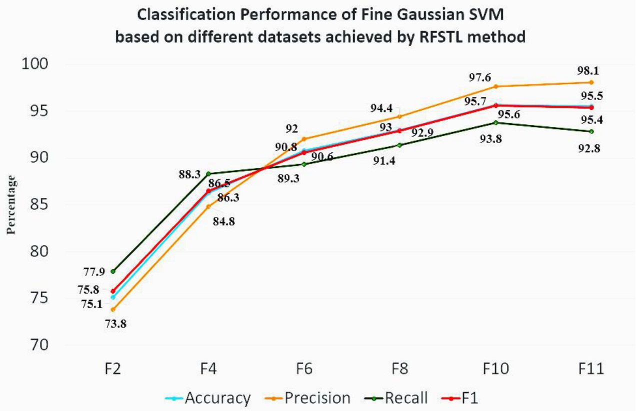

FIGURE 5. Performance of imbalanced classification by Fine Gaussian SVM inputting different dataset with different vital features achieved by RFSTL method based on Secom1.

In Table 8, two different criteria are applied to determine the optimal subset of features based on the experiments listed in Table 7. One way is to determine optimal features when the maximum value of F1 is achieved by the minimum number of features. Another way is to determine optimal features when the value of AUC is equal to 1. In the process of finding the optimal subset of features, the two different determination criteria are performed at the same time. Whenever one of them is satisfied the optimal feature search will be stopped immediately, and the currently selected features will be denoted as the vital features for classification.

Taking the quality prediction effect of the classification model applying RFSTL method on the data set Secom1 (illustrated in Table 7 described as RF method based on Secom2) as an example, the value of F1 is 95.6 when the number of selected features is 10, while the value of F1 is 95.4 when the number of selected features is 11. When the number of selected features is increased from 10 to 11, the F1 value has a slight decrease. Therefore, the top 10 features ranked by RF is the optimal feature subset to obtain the best classification performance. Fig. 5 shows the change trend of classification performance (Accuracy, Precision, Recall, and F1) brought by the different feature subset selected by RFSTL method.

Seen from Table 8, the two best feature selection methods, which can achieve the maximum classification performance (F1) using a minimum number of features, are RF and SRK. Although the vital features they selected are not the same, the quantities of vital features are the same (10 features). The classification performance of Fine Gaussian SVM inputting the top 10 of most vital features achieved by SRK and RF method based on balanced dataset Secom2 is shown in Fig. 6.

TABLE 8. Optimal features selected by different feature selection algorithms based on Secom2 for quality prediction utilizing the Fine Gaussian SVM model.

Feature selection algorithm	Quantity of essential features	No. of essential features[a]	Criteria to determine the quantity of essential features
CC	14	60,104,130,29,125,511,122,124,123,128,131,134, 80,65	Achieve the maximum value of F1 (F1=95.7) with the minimum number of features.
Fisher	16	60,104,130,29,125,511,122,124,123,128,131,134, 80,65,96,418	Achieve the maximum value of F1 (F1=95.8) and the maximum value of AUC (AUC=1).
GA	15	1,2,5,7,9,11,12,14,15,19,20,21,23,24,25	Achieve the maximum value of AUC (AUC=1).
mRMR	11	487,34,86,255,248,384,122,490,96,281,564	Achieve the maximum value of F1 (F1=95.4) with the minimum number of features.
ReliefF	12	420,512,419,501,487,500,483,490, 489,549,86,52	Achieve the maximum value of AUC (AUC=1).
RF	10	60,487,96,248,478,122,104,125,279,131	Achieve the maximum value of F1 (F1=95.6) with the minimum number of features.
SFS	12	60,66,156,28,29,474,349,16,296,350,201,118	Achieve the maximum value of F1 (F1=95.1) with the minimum number of features.
Single	14	60,104,478,511,206,342,80,426,418,128,125,18,8 1,469	Achieve the maximum value of F1 (F1=95.4) with the minimum number of features.
SWF	12	60,65,122,130,32,68,76,146,57,424,144,201	Achieve the maximum value of F1 (F1=96.4) with the minimum number of features.
SRK	10	60,29,122,130,112,447,487,86,420,32	Achieve the maximum value of AUC (AUC=1).
SRO	22	60,205,65,349,215,156,360,494,341,201,252,290, 426,425,100,474,57,54,61,456,112,416	Achieve the maximum value of AUC (AUC=1).

[a] The value range of the feature No. is from 1 to 590.

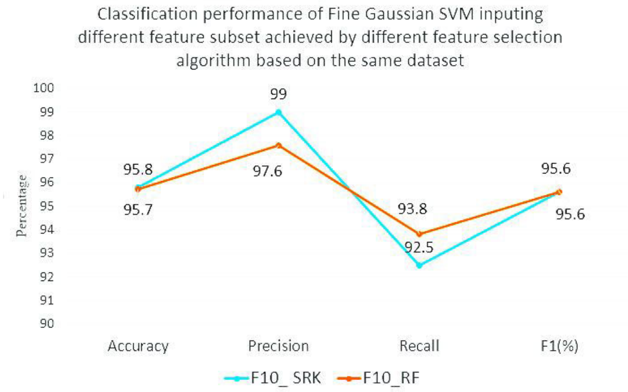

FIGURE 6. Classification performance of Fine Gaussian SVM inputting the top 10 of most vital features achieved by SRK and RF method based on balanced data set Secom2.

These two feature methods can obtain a similar classification effect (Accuracy, F1, and AUC) and achieve a small difference in Precision and Recall. From the view of classification performance, it is difficult to clearly point out which feature selection algorithm performs better in balancing data sets. However, from the analysis of the algorithm execution time (listed in Table 4), it is easily found that the RF algorithm is more applicable in the face of large data problems. Because of this, the execution time of the SRK algorithm is about 6 times as much as that of the RF algorithm. Hence, compared to other feature selection algorithms, the RF method is a superior way to obtain less quantity of value features from a big balanced data set with a superior classification performance in a short time.

- Performance evaluation of manufacturing quality prediction based on the Fine Gaussian SVM model inputting different data

Four different data sets are input into the classifier to evaluate the classification performance listed in Table 2: Secom1, an imbalanced data set after pre-processing containing 590 features and 1533 observations; Secom 2, a balanced dataset obtained from the imbalanced dataset Secom1 applying SMOTE&TomekLinks algorithm containing 590 features and 2860 observations; Secom3, a imbalanced dataset with a low dimensionality acquired through the RF method containing 10 features and 1533 observations; Secom4, a balanced dataset with a low dimensionality acquired through the RFSTL method containing 10 features and 2860 observations.

Upon comparison of the experiment's results, it is obvious that the Fine Gaussian SVM model cannot run well when faced by an imbalanced data set (Secom1 and Secom3) which achieves a misleading outcome that a false high accuracy is obtained by classifying all defective products (the minority class) into the qualified products (the majority

TABLE 9. Prediction performance of the Fine Gaussian SVM model using different input.

Input	TP	FP	TN	FN	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC	Training Time (sec)
Secom1	0	0	1430	103	93.3	-	0	-	0.56	16.43
Secom2	1321	0	1430	109	96.2	100	92	95.8	1	26.23
Secom3	0	2	1428	103	93.2	0	0	-	0.67	0.89
Secom4	1341	33	1397	89	95.7	97.6	93.8	95.6	0.99	5.03

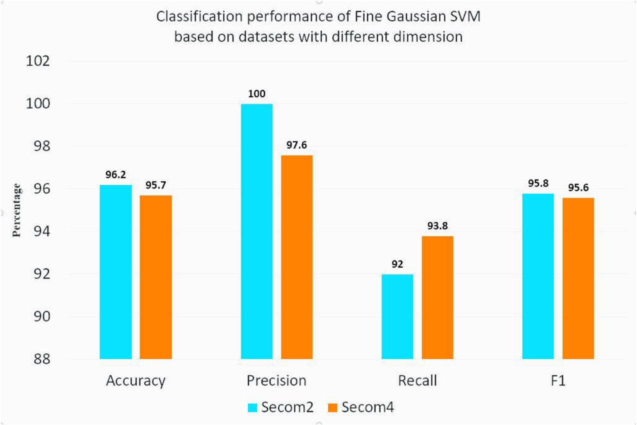

FIGURE 7. Classification performance of Fine Gaussian SVM model based on two different balanced datasets obtained by SMOTE & tomekLinks method and RFSTL method respectively.

class). On the other hand, if inputting the balanced data set with high dimensionality (Secom2) obtained by the method of SMOTE&TomekLinks or the balanced data set with low dimensionality (Secom4) achieved by the RFSTL algorithm, this model can acquire a super classification performance especially for the minority class shown in Fig. 7.

In Fig. 7, some differences occurred in the value of Precision and Recall. The Precision value of this model taking Secom2 as input is 2.4% higher than that of Secom4. Otherwise, the Recall value of this model taking Secom2 as input is 1.8% lower than that of Secom4. Additionally, the F1 acquired by the same classifier using different inputs Secom2 and Secom4 just has a slight difference which can be ignored. Relative to the differences in classification performance, the significant difference between this model utilizing Secom2 and Secom4 is the feature dimensionality and the training time of the model. When applying RFSTL algorithm, the dimensionality of feature can be reduced to only 1.7% of the original features and the training time of the classification model can be shortened to only one fifth of that input Secom2.

Furthermore, the value of AUC acquired by the classification model applying RFSTL method is almost double to that obtained without do data balancing. The ROC curve of the Fine Gaussian SVM taking the Secom4 as input is displayed in Fig. 8. The ROC curve in this figure is close to the top left, which illustrates the classification problems encountered by this model due to the imbalanced data, which can be resolved by the RFSTL method.

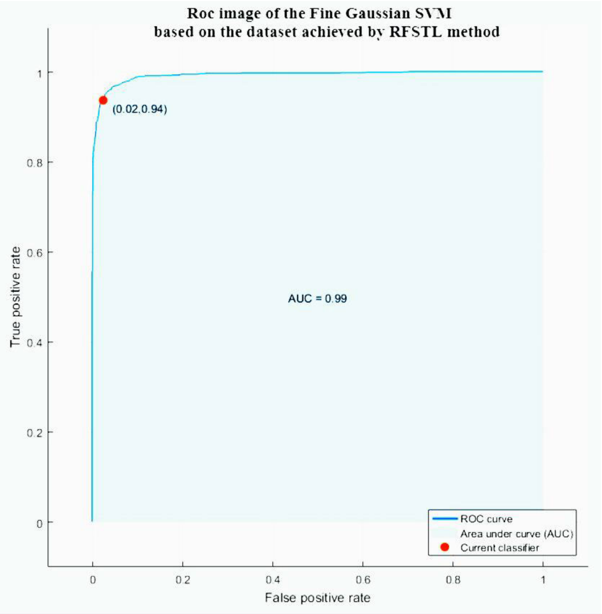

FIGURE 8. Roc Image of Imbalanced Classification based on the Fine Gaussian SVM model using RFSTL Method.

V. CONCLUSION

Skewed class distribution data and high dimensional features in the real-world industry pose an intense challenge for design learning algorithms, and have a high negative impact on the performance of learning models. This study first provided an introduction of the research and detailed in some existing imbalanced handling, feature dimension reduction, and classification approaches. Secondly, it proposed the hybrid method, RFSTL, which balanced the data set based on the SMOTE&TomekLinks approach, then selected the most valuable features according to the Random Forest algorithm. Finally, the RFSTL method was simulated on a realistic imbalanced data set Secom with a high dimension using the Fine Gaussian SVM to do manufacturing quality prediction. Some thorough experimental comparisons were taken and its impact on classification performance was discussed.

Experimental results demonstrated that the classifiers employing the preprocessed data with the RFSTL method, achieved a high value for F1 (95.6%) and AUC (0.99). It was more accurate than those employing the raw data and shifted the classifier learning bias towards the minority class. Another compelling justification was that it is helpful to reduce the feature dimension and decrease the training time of the model. Compared to using the data set after simple preprocessing, by using the RFSTL method, the feature dimension was reduced from 590 to 10 and the training time of the model shortened from 16.43 seconds to 5.03 seconds. Therefore, the RFSTL method can run effectively for classification on a large, high-dimensional, imbalanced data set, and could be applied to intelligent manufacturing for quality prediction as well as other intelligent applications under Industry 4.0.

REFERENCES

[1] F. Kamalov, "Kernel density estimation based sampling for imbalanced class distribution," *Inf. Sci.*, vol. 512, pp. 1192–1201, Feb. 2020.

[2] M. Koziarski, "Radial-based undersampling for imbalanced data classification," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107262.

[3] Z. Xu, D. Shen, T. Nie, and Y. Kou, "A hybrid sampling algorithm combining M-SMOTE and ENN based on random forest for medical imbalanced data," *J. Biomed. Informat.*, vol. 107, Jul. 2020, Art. no. 103465.

[4] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Int. J. Speech Technol.*, vol. 50, no. 8, pp. 2488–2502, Aug. 2020.

[5] X. Tao, Q. Li, W. Guo, C. Ren, C. Li, R. Liu, and J. Zou, "Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification," *Inf. Sci.*, vol. 487, pp. 31–56, Jun. 2019.

[6] C. Jimenez-Castaño, A. Alvarez-Meza, and A. Orozco-Gutierrez, "Enhanced automatic twin support vector machine for imbalanced data classification," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107442.

[7] H. Xu, J. Zhang, Y. Lv, and P. Zheng, "Hybrid feature selection for wafer acceptance test parameters in semiconductor manufacturing," *IEEE Access*, vol. 8, pp. 17320–17330, 2020.

[8] W. Lee and K. Seo, "Early failure detection of paper manufacturing machinery using nearest neighbor-based feature extraction," *Eng. Rep.*, Sep. 2020.

[9] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, Feb. 2012.

[10] H. K. Lee and S. B. Kim, "An overlap-sensitive margin classifier for imbalanced and overlapping data," *Expert Syst. Appl.*, vol. 98, pp. 72–83, May 2018.

[11] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017.

[12] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[13] V. López, A. Fernández, J. G. Moreno-Torres, and F. Herrera, "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics," *Expert Syst. Appl.*, vol. 39, no. 7, pp. 6585–6608, Jun. 2012.

[14] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 39, no. 2, pp. 539–550, Apr. 2009.

[15] I. Tomek, "Two modifications of CNN," *IEEE Trans. Syst., Man, Cybern.*, vol. 6, no. 11, pp. 769–772, Nov. 1976.

[16] T.-Y. Liu, "EasyEnsemble and feature selection for imbalance data sets," in *Proc. Int. Joint Conf. Bioinf., Syst. Biol. Intell. Comput.*, 2009, pp. 517–520.

[17] W. A. Rivera and P. Xanthopoulos, "*A priori* synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets," *Expert Syst. Appl.*, vol. 66, pp. 124–135, Dec. 2016.

[18] 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.

[19] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new oversampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.*, 2005, pp. 878–887.

[20] G. E. Batista, A. L. Bazzan, and M. C. Monard, "Balancing training data for automated annotation of keywords: A case study," in *Proc. Brazilian Workshop Bioinf.*, 2003, pp. 10–18.

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

[22] M. A. Hall, "Correlation-based feature selection for machine learning," Doctoral dissertation, Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, 1999.

[23] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. Eur. Mach. Learn.*, 1994, pp. 171–182.

[24] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1992, pp. 249–256.

[25] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[26] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *J. Bioinf. Comput. Biol.*, vol. 3, no. 2, pp. 185–205, Apr. 2005.

[27] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.

[28] K. De Jong, "Learning with genetic algorithms: An overview," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 121–138, Oct. 1988.

[29] J. Yang and V. Honavar, "Feature subset selection using a genetic algorithm," *IEEE Intell. Syst. Appl.*, vol. 13, no. 2, pp. 44–49, Mar./Apr. 1998.

[30] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[31] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *J. Amer. Stat. Assoc.*, vol. 97, no. 457, pp. 77–87, Mar. 2002.

[32] K.-Q. Shen, C.-J. Ong, X.-P. Li, Z. Hui, and E. P. V. Wilder-Smith, "A feature selection method for multilevel mental fatigue EEG classification," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 7, pp. 1231–1237, Jul. 2007.

[33] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemometric Intell. Lab. Syst.*, vol. 83, no. 2, pp. 83–90, Sep. 2006.

[34] M. Shieh and C. Yang, "Multiclass SVM-RFE for product form feature selection," *Expert Syst. Appl.*, vol. 35, nos. 1–2, pp. 531–541, Jul. 2008.

[35] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[36] J. M. Cadenas, M. C. Garrido, and R. Martínez, "Feature subset selection Filter–Wrapper based on low quality data," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6241–6252, Nov. 2013.

[37] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Syst. Appl.*, vol. 38, no. 7, pp. 8144–8150, Jul. 2011.

[38] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[39] S. R. Gunn, M. Brown, and K. M. Bossley, "Network performance assessment for neurofuzzy data modelling," in *Proc. Int. Symp. Intell. Data Anal.*, 1997, pp. 313–323.

[40] T.-T. Nguyen, J. Z. Huang, and T. T. Nguyen, "Unbiased feature selection in learning random forests for high-dimensional data," *Sci. World J.*, vol. 2015, Dec. 2015, Art. no. 471371, doi: 10.1155/2015/471371.

[41] H. Han, X. Guo, and H. Yu, "Variable selection using mean decrease accuracy and mean decrease gini based on random forest," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Aug. 2016, pp. 219–224.

[42] M. McCann and A. Johnston. (2008). *UCI Machine Learning Repository: SECOM Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/SECOM

[43] G. E. Batista and M. C. Monard, "A study of K-nearest neighbour as an imputation method," *HIS*, vol. 87, nos. 251–260, p. 48, 2002.

[44] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. Roy. Stat. Soc., B, Methodol.*, vol. 36, no. 2, pp. 111–133, 1974.

HONG ZHOU received the B.S. degree in computer science and technology from the Huaiyin Institute of Technology, Huaian, Jiangsu, China, in 2004, the M.S. degree in Internet computing from the University of Abertay, Dundee, U.K., in 2008, and the Ph.D. degree in engineering science from Chung Hua University, Hsinchu, Taiwan, in 2020. Since 2004, she has been a Teacher with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, where she is currently an Assistant Professor. Her research interests include machine learning, data mining, big data, and artificial intelligence.

KUN-MING YU received the B.S. degree in chemical engineering from National Taiwan University, in 1981, and the M.S. and Ph.D. degrees in computer science from The University of Texas at Dallas, in 1988 and 1991, respectively. From 2005 to 2017, he was the Dean of the College of Computer Science and Informatics, Chung-Hua University. He is currently a Vice President of Chung-Hua University. His research interests include artificial intelligence, big data, the Internet of Things, high-performance computing, ad hoc networks, and computer algorithms.

YEN-CHIU CHEN received the B.S. degree in information management from Chung Hua University, Hsinchu, Taiwan, in 2004, and the Ph.D. degree in computer science from National Tsing Hua University, Hsinchu, in 2010. From 2010 to 2018, she was a Research and Development Engineer with Information and Communications Research Labs (ICL), Industrial Technology Research Institute (ITRI), Hsinchu. She is currently an Assistant Professor with the Department of Information Management and the CEO of AI$^+$ Experience Center, Chung Hua University. Her research interests include artificial intelligence technology, edge computing of 5G/6G, real-time scheduling algorithms, and graph theory.

HUAN-PO HSU received the B.S. degree in mechanical engineering from Chinese Culture University, in 2013, and the M.S. degree in computer science and information engineering from Chung Hua University, in 2015, where he is currently pursuing the Ph.D. degree in engineering science. He is also the Chairman of SmartPearls Company Ltd. His research interests include the IoT and intelligent evacuation guiding technology.