Behaviour of Imbalanced Data in Presence of Borderline and Noisy Examples using Hybrid SPIDER2-IPF Boosting Ensemble Method

Neelam Rout[1], Debahuti Mishra[2] and Manas Kumar Mallick[3]

[1]Research Scholar, Department of Computer Science & Engineering, Siksha 'O' Anusandhan Deemed to be University

[2,3]Professor, Department of Computer Science & Engineering, Siksha 'O' Anusandhan Deemed to be University

[1]neelamrout@yahoo.com [2]debahutimishra@soauniversity.ac.in [3]director.iter@soauniversity.ac.in

Corresponding Author- Neelam Rout

Abstract: A crucial task in many different fields is classification. The use of conventional classification algorithms has been limited by issues like class imbalance, overlaps, and noise. When mismatched class distributions are found among the instances of class of classification datasets, an imbalance problem occurs. When datasets are unbalanced and contain noisy and borderline data, classification becomes significantly more difficult. Noisy data are comparable to minority samples, and any method for resolving the class imbalance may focus excessively on the noise, impairing performance. The study suggests using the SPIDER2-IPF Models to handle noisy and borderline situations. An experiment employing the Saturation, PANDA, Classification, ANR, and SPIDER2-IPF filter methods shows the impact of borderline and noisy data from the rare class on the performance of the classifier. To handle noisy and borderline samples, the SPIDER2-IPF model is built using SPIDER2 resampling techniques. In imbalanced datasets, an Iterative-Partitioning Filter (IPF) can reduce noise from both majority and minority classes and address issues brought on by noisy and borderline samples. The results of the SPIDER2-IPF model are 99.51%, 67.78%, 81.39%, 99.36%, 99.63%, and 99.48% for sensitivity, specificity, G-Mean, precision, recall, and F-Measure. According to the results of the experiments, the suggested methods can effectively address the issue of class imbalance with noisy and borderline challenges. The Wilcoxon test results show that the suggested method worked effectively with unbalanced data. Finally, the suggested solutions can successfully address this class of issues.

Keywords: Classification, Imbalance Data, Borderline & Noisy Examples, Ensemble Method, SPIDER2-IPF Model, Filter Methods, Boosting Method, Performance Metrics, Wilcoxon Test

1. Introduction

The evaluation of input and target variables is the aim of classification. The effectiveness of the classification algorithms can be hampered by unbalanced, borderline, and noisy data. Unbalanced issues depict a situation in which there are much more examples of the majority class than the minority class [1] [2]. Unbalanced categorization is a carefully researched area in data mining. Fraud detection, fault classification in manufacturing, text classification, disease diagnosis, event classification, oil spill detection, intrusion

detection, and others are some of the real-world issues [3] – [9]. Data imbalance issues can be solved using ensemble approaches [10]. Several classifiers are combined into one classifier using the ensemble technique to get more accurate results [11] [12]. The phrase "boosting ensemble approach" describes a collection of methods that can improve weak learners. A powerful learner is very close to fine performance, whereas a weak learner is only marginally finer than a random utterance on the surface. A sequential ensemble technique called boosting is used to lower bias error and build effective prediction models. A group of algorithms collectively known as "boosting" support in the development of a weak learner into a strong one. The topics of discussion are noisy, borderline, and safe instances, with the “borderline and noisy” examples being the source of learning algorithm issues. The area near class boundaries, where the majority and minority classes overlap, is where borderline occurrences can be identified. Safe examples are placed in locations that are reasonably homogeneous and have the same class label. Finally, noisy examples can be found in the other class's safe zones [13]. As the performance metrics accuracy and its complement, misclassification rate is not worked well on imbalanced data, so other metrics should take into consideration like G-mean [14]. Due to the effectiveness of the Selective pre-processing of Imbalanced Data 2 (SPIDER2) approach, the SPIDER2-IPF model is proposed to control noisy and borderline examples [15] [16]. SPIDER2 is the improved version of selective pre-processing and the iterative-partitioning filter (IPF) is a good noise filtering approach.

The imbalanced categorization problem presents a number of challenges from a scientific and empirical perspective. Data imbalance's nature is frequently unclear or changes from case to case. On the one hand, categories in the existing data set may be inherently biased as a result of the problem's direct genesis. However, the data gathering procedure could be to blame for the disparity. An interesting topic's initial data distribution may be balanced, but practical issues like time, storage, or cost cause the obtained data set to be uneven. The "intrinsic" and "extrinsic" imbalances require various learning techniques in order to provide acceptable classifiers for future events. Minority-class learning can be hampered by a variety of data-complexity requirements, such as inequality between classes or between several sub clusters within a single class. Learning inner concepts for generic classifiers is more challenging when there is internal imbalance. If the concepts produced within or between classes are mixed, the generalisation potential of an algorithm will be severely constrained. Learning a distinct separation border is challenging because the knowledge represented in the majority and minority classes may overlap, which may have overlapped instances in the final data. Even if a precisely adjusted border can be constructed between them, such classifiers will be over fitted and their potential to generalise on [17] [18] future data would be impaired.

Numerous data characteristics may create difficult learning obstacles. Developing effective learning methods is already challenging for data sets with multiple characteristics and small sizes; adding the imbalance feature makes it even more challenging. Only a few examples of

the minority class, which are insufficient to reflect the minority notion, may occur due to insufficient or highly skewed relevant data. As a result, the classifiers' taught ideas might not be accurate. Unbalanced learning is significantly harmed by data noise and missing values, especially for minority classes. Actual knowledge is more challenging to extract from those unusual circumstances. It has never been possible to evaluate the performance of imbalance learning classifiers using any widely accepted consistent datasets or assessment metrics. Finally, the attention should be on how to create new unbalanced data techniques, [19] as well as how to create uniform performance measures and standardise extensive datasets.

To get better insight into dealing with borderline and noisy datasets issues, few literatures have been studied and discussed. There are two existing approaches such as; data level approach and algorithm level approach to handle these types of datasets. In the data level approach, noisy or irrelevant examples are eliminated from the training data. In the algorithm level, the learning mechanism of the algorithm is improved to make it less sensitive to deal with these types of datasets. In [20], the authors have represented an experimental investigation of noisy and imbalanced data using random under sampling and classification methods. The authors [21] have used the noise tolerance GDES-AD method to minimize the bias in classification error. The researchers [22] have explained the challenges of class noise with imbalance factor on eleven different algorithms with seven types of data sampling techniques to identify most robust method. In [23], the authors have analysed the behaviour of noisy data using MCSs for the experimental study. In [24], an extension of SMOTE [25] with Iterative-Partitioning Filter is proposed to overcome the problems faced by borderline and noisy instances in imbalanced data and finally, it has performed well than existing SMOTE. In this [26] work, a combined technique Soft-Hybrid is proposed to handle overlapping, non-overlapping and borderline data. The researchers [10] have explored the connection between ensemble diversity [27] and noisy data by learning different types of diversity measures. In [28], noisy instances are either filtered or managed by fixing the labels for classes. Finally, using the cleansed training data, a new ensemble is created. Extensive studies in a variety of classification tasks show that this technique is efficient in constructing correct ensembles even when there is a lot of class-label noise. Some modifications [29] of fuzzy rule-based and algorithmic are used to improve the performance and the experiment is done for highly unbalanced and borderline datasets which is passed through a test (statistical). In [30], first fuzzy rough prototype selection algorithm discards noisy examples from the unbalanced data, then clears the data produced by SMOTE method. The graph stream model is useful because of less expensive to handle imbalance noisy data [31]. The OSM classifier have proposed [32] to get better performance than existing methods for overlapping with imbalance situations. In the work of [13], the re-sampling methods, NCR, and SPIDER2 are used to do experiments on datasets to analyse the impact of noisy instances from the uncommon class with borderline performance on [33] classifiers.

1.1 **Motivations**

There are numerous reasons for selecting this topic for this study, some of which are listed below.

a) Many algorithms in data mining suffer with unbalanced datasets. The crucial aspect is appropriately predicting minority class cases. In cancer data, for example, precisely classifying a cancerous patient is significantly more critical than accurately classifying a non-cancerous patient. While ensemble approaches aim to create a group of learners and combine them, traditional learning systems aim to create one learner from training data.

b) Dealing with unbalanced data in the context of noisy and borderline examples is extremely difficult.

c) Boosting is the process of putting together a group of weak learners in a sequential order. The boosting strategy focuses on training things the preceding model did incorrect. The idea of rectifying prediction errors is an important feature of boosting ensembles. The models are fitted and given to the ensemble one at a time, with the second model correcting the first model, the third model correcting the second model, and so on. Boosting algorithms only select features that have a significant impact on the objective, potentially reducing dimensionality while also increasing computational efficiency.

d) Selective pre-processing of unbalanced data is used to combine local over-sampling of the minority class with filtering difficult situations from the majority classes.

e) IPF uses noise filtering techniques to improve software quality prediction. In numerous repetitions, this approach removes noisy instances until a stopping requirement is met.

1.2 Original Contributions

In this study, the ways to dealing with unbalanced data, as well as noisy and borderline situations, are suggested. It contains the following contributions:

a) Input to the classifiers will be a variety of imbalanced datasets from the "**KEEL"** [34] dataset source.

b) The **SPIDER2-IPF** model is built using SPIDER2 as the resampling method and IPF [24] as the noise filter to deal with noisy and borderline datasets. By reducing noisy data from training datasets, the IPF noise removal method enhances its quality. Only one base learner is used by the Iterative-Partitioning Filter, although iterations are performed [13] numerous times.

c) The results of these models are recorded and displayed in a tabular format.

d) The performance metrics (specificity, specificity, G-mean, precision, recall, F1-measure) [35] are used to evaluate the model.

e) The important findings in terms of experimentation have been thoroughly examined.

1.3 Paper Organization

The remainder of the document is structured as follows.

--Section-2 focuses on a concise discussion of methodology adoption.

--Section-3 contains a broad overview of the proposed model.

--Section-4 discusses about the research metrics and empirics, system configuration, datasets preparation, and performance metrics.

--Section-5 proposes the framework for noisy & borderline examples and presents experimental
results.
--Section-6 briefly discuss the key points of the entire work.
--Section-7 concludes this paper and recommendations for future work.

2. Preliminaries/ Methodologies Adopted

The various methods such as Saturation filter, PANDA Filter, Classification Filter, ANR filter, SPIDER2-IPF and Boosting methods are used for the experimentation. Ensemble approaches make use of a large number of learners to improve the performance of any one of them. These techniques combine a bunch of weak learners to generate a powerful model.

3. Broad Overview of the Proposed Model

Figure 1 depicts the overall framework of the work, whereas figure 1 describes the planning flow for noisy and borderline samples. The noisy and borderline datasets are trained in by applying Saturation filter, PANDA Filter, Classification Filter, ANR filter, SPIDER2-IPF, and Boosting methods, and the results are given using a confusion matrix. The Description of different methods are given below.

Figure 1. Diagrammatical representations of the proposed method for noisy & borderline examples

Saturation Filter Method: Since it is theoretically based on the saturation property of the training set, the entire process for noise identification and elimination [36] is known as the saturation filter. The minimal-covering algorithm is a heuristic included in this algorithm.

PANDA Filter Method (Pairwise Attribute Noise Detection Algorithm Filter): **"**Pairwise Attribute Noise Detection Algorithm", often known as PANDA [37]. Considering the values of a couple of characteristics, PANDA aims to discover occurrences with a large deviation from the [38] norm.

ANR Filter Method (Automatic Noise Reduction): In order to find and eliminate noisy data items with incorrectly identified classes, ANR is utilised as a filtering strategy. The multi-layer artificial neural network framework [39] provides a basis for the ANR's underlying process.

SPIDER2 Resampling Method: The studies of Stefanowski and Wilk on the SPIDER (Selective Preprocessing of Imbalanced Data) approach to selective preprocessing. This approach uses the " Edited Nearest Neighbor Rule (ENNR) " to pinpoint the local characteristics of samples before combining the removal of majority class objects that might lead to the misclassification of minority class objects with local oversampling of those minority class objects that are "overwhelmed" by nearby majority class objects. Two steps of SPIDER2 make up the pre-processing of c_{min} and c_{maj}, respectively. The relabel option either eliminates or relabels noisy instances from c_{maj} after identifying the features of the examples from that key in the first phase (i.e., modifications their classification to c_{min}). The characteristics of instances from c_{min} are identified in the second phase while taking into account the [40] previous phase's alterations. Then, using the ampl option, noisy instances from c_{min} are replicated and amplified.

Algorithm:

Input: Data set (DS), minority class (CMin), number of closest neighbours (k), option to relabel (yes/no), and option to amplify (ampl) (“no, weak, strong“)

Output: Preprocessing DS

An artificial class uniting all classes except $c_{min} := c_{maj}$
foreach x ∈ (belongs to) class (DS, c_{maj}) **do**
if right (DS, x, k) **then** flag x as safe
else flag x as not-safe
RS := flagged(DS, c_{maj} , not-safe)
if relabel **then**
foreach y ∈ (belongs to) RS **do**
transformation classification of “y to c_{min}“
SR := SR \ {y}
else DS := DS \ RS
foreach x ∈ (belongs to) class (DS, c_{min}) **do**
if correct (DS, x, k) **then** flag x as safe
else flag x as not-safe
if ampl = weak **then**
foreach x ∈ (belongs to) flagged (DS, c_{min}, not-safe) **do** amplify (DS, x, k)
else if ampl = strong **then**
foreach x ∈ (belongs to) flagged (DS, c_{min}, not-safe) **do**
if accurate (DS, x, k + 2) **after that** amplify (DS, x, k)
else amplify (DS, x, k + 2)

IPF Noise Filter Method: The advantage of these filters is that noisy cases are eliminated iteratively with the understanding that doing so will not affect the noise detection in later [41] iterations. The partitioning filter's specialised form is known as an iterative partitioning filter. A model is then constructed on each of the n subsets that were initially created from the practise data. An instance is labelled as noise when this filter method employs the majority strategy and more than 50% of the models misclassify it. If the estimation of all n models differs from the actual class label of the instance, the consensus iterative partitioning filter removes the instance. Additionally, an instance must be misclassified at least by the model that is induced on the subset that contains it in order to be classified as noisy. Up until the halting requirement is met, noisy occurrences are deleted iteratively. The needed number of filtering rounds can be changed to alter how conservative the filter should be [42].

A pre-processing method modelled on the Partitioning Filter is called the Iterative-Partitioning Filter (IPF). In huge data sets, it is used to locate and get rid of instances that were incorrectly categorised. Most noise filters make the supposition that data sets are small and can be learned with just one learning attempt. However, partitioning methods can be necessary because this isn't always the case. Until a stopping requirement is met, IPF removes noisy instances iteratively. If, for a sequence of s iterations, the number of recognised noisy instances in each of those iterations is less than a percentage p of the size of the initial training data set, the iterative procedure comes to an end. At first, a set of useful data ($DG = \emptyset$) and a set of noisy examples ($DN = \emptyset$) are used. Each iteration's fundamental steps are:

a. Divide the trained data set D_T into Γ subsets of same size.
b. According to the authors' advice, C4.5 is trained on each of these Γ parts. Different trees are produced as a result.
c. By comparing the training label with the label supplied by the classifier, these *Γ* resulting classifiers are then adopted to categorise each instance in the training set D_T as either correct or mislabelled.
d. By employing majority voting, add the noisy instances found in D_T to D_N while considering the accuracy of the labels the built-in *Γ* classifier produced in the previous stage.
e. Take the training set and delete the good and noisy examples: $D_T \leftarrow D_T \setminus \{D_N \cup D_G\}$.

The remaining D_T instances [31] and the good D_G data are combined to generate the filtered data at the end of the iterative procedure, resulting in $D_T \cup D_G$.

A decision tree-generating algorithm is C4.5. Many recent analyses of unbalanced data have employed it. From a set of provided examples, it generates [43] classification rules in the form of decision trees. The normalised information gain (difference in entropy) that results from selecting an attribute to split the data is used to build the decision tree top-down. The attribute utilised to make the decision has the highest normalised information gain.

Confidence level: c = 0.25
Minimal instances per leaf: i=2
Prune after the tree building

Algorithm for Boosting:

1. Set up the dataset and give each data point the same amount of weight
2. Give this as model input and find the data points that were incorrectly classed
3. Increase the weight of the data points that were incorrectly categorised
4. if (got necessary results)
   Goto step five
   else
   Goto the second step
5. End

"Boosting is an ensemble modelling strategy that aims to create a strong classifier out of a large number of weak ones. It is accomplished by constructing a model from a sequence of weak models. To begin, a model is created using the training data. The second model is then created, which attempts to correct the faults in the first model. This approach is repeated until either the entire training data set is properly

predicted or the maximum number of models has been added".

4. Research Metrics and Empirical Observation

The system setup, numerous datasets, applied settings, and experiment-wide performance measures are covered in this section. Boosting ensemble method [44] is taken to solve the imbalanced problems for its popularity.

4.1 System Configuration

The entire experiment was carried out on a single-language version of Windows 10 with an Intel® Core(TM) i5-7300HQ processor running at 2.50GHz. The operating system is 64-bit, with an x64-based processor and 8.00 GB of installed memory (RAM).

4.2 Datasets Preparation and Parameters

In the first part of the experiment, there are three types imbalanced datasets are used. **Table 1** has shown the information of Imbalanced datasets. In the second part of the experiment, noisy and borderline examples are taken. There are three shapes of datasets are used (clover, subclus, and paw). **Figure 2** shows clover data set having five petals and it is very difficult to solve where the rare class appears like a flower. **Figure 3** shows subclus dataset where the minority class instances are positioned inside rectangle [45]. Finally, **Figure 4** shows paw datasets where the rare class is transferred into 3-elliptic sub-regions having different cardinalities, where 2 parts are closed to one another, and the smaller remaining part is kept separate. Safe with borderline, and noisy examples are diagrammatically shown in **Figure 5** [46] [47]. All the datasets are freely available in Keel dataset repository site [35].

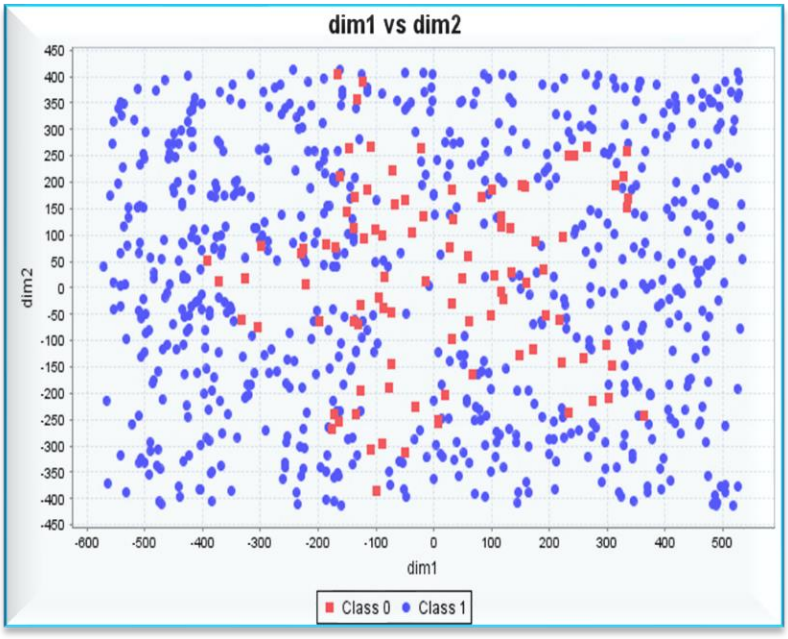

Figure. 2. Clover dataset

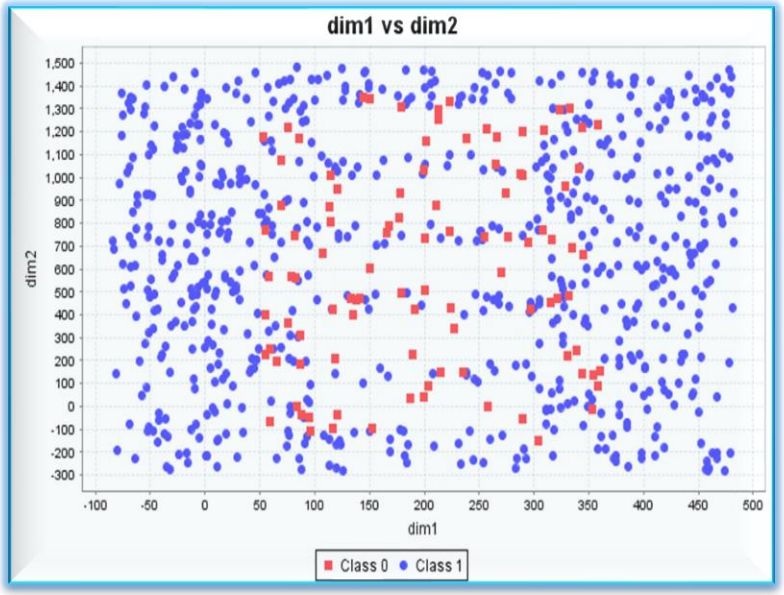

Figure. 3. Subclus dataset

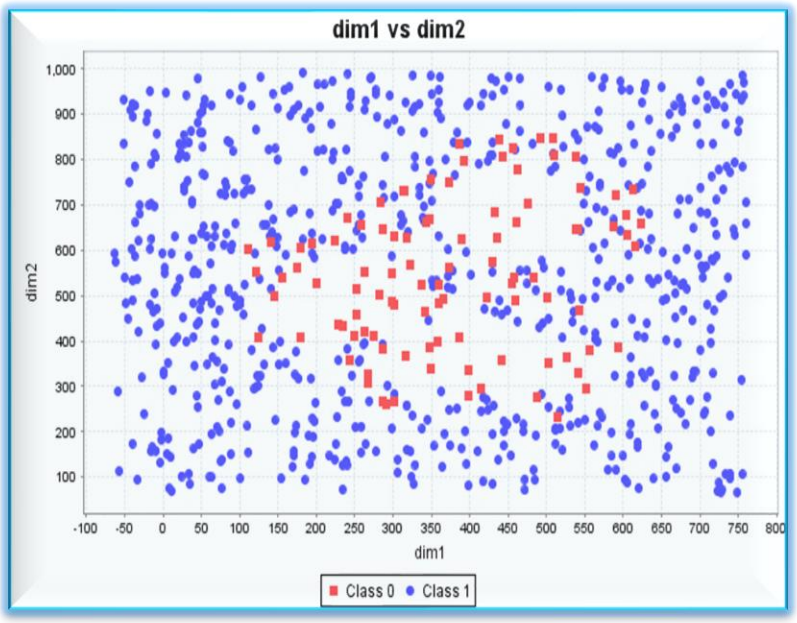

Figure. 4. Paw dataset

Table 1 is used to describe the information about the borderline and noisy datasets. There are nine types of datasets taken for the experiment. The number of borderline cases from rare class sub-regions is increasing as the disturbance ratio, which is where borders of sub-regions in rare classes are disturbed. In the both the experiment, 5-fold cross-validation method is used where the dataset is split into five folds and each fold has 80% training data and 20% testing data [35].

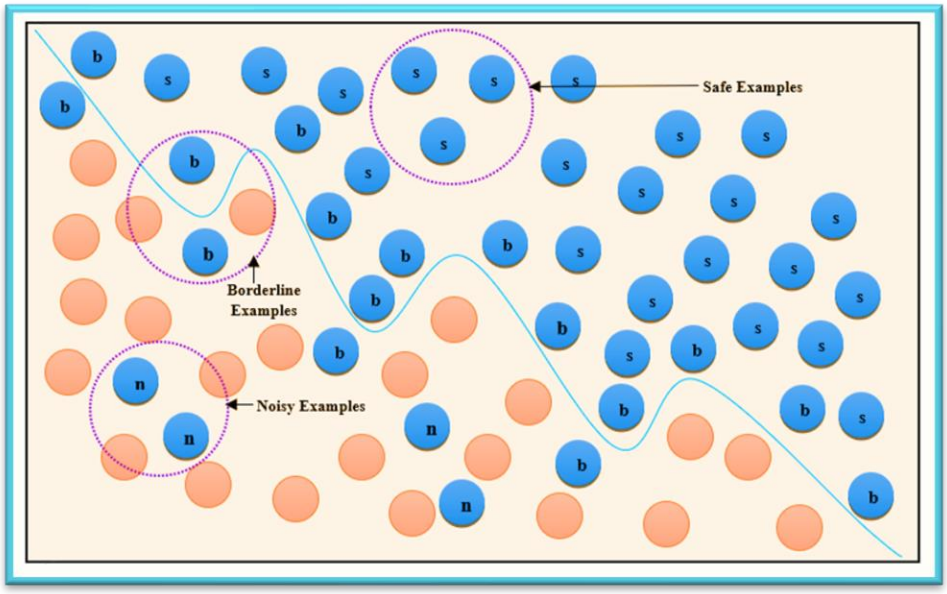

Figure 5. Safe examples (s), borderline examples (b) and noisy examples (n) are shown diagrammatically
and the continuous line is used for the decision boundary to separate the two classes

Table 1
Information of borderline and noisy datasets

Datasets Names	Instances	Features	Classes	Imbalance Ratio (IR)	Disturbance Ratio (%)
04clover5z_600-5_50_BI	600	2	2	5	50
04clover5z_600_5-60-BI	600	2	2	5	60
04clover5z_600_5_70_BI	600	2	2	5	70
03subcl5_600_5_50_BI	600	2	2	5	50
03subcl5_600_5_60_BI	600	2	2	5	60
03subcl5_600_5_70_BI	600	2	2	5	70
paw02a_600_5_50_BI	600	2	2	5	50
paw02a_600_5_60_BI	600	2	2	5	60
paw02a_600_5_70_BI	600	2	2	5	70
04clover5z_800_7_30-BI	800	2	2	7	30
04clover5z_800_7_50_BI	800	2	2	7	50
04clover5z_800_7_60_BI	800	2	2	7	60
04clover5z_800_7_70_BI	800	2	2	7	70
03subcl5_800_7_30_BI	800	2	2	7	30
03subcl5_800_7_50_BI	800	2	2	7	50
03subcl5_800_7_60_BI	800	2	2	7	60
03subcl5_800_7_70_BI	800	2	2	7	70
paw02a_800_7_30_BI	800	2	2	7	30
paw02a_800_7_50_BI	800	2	2	7	50
paw02a_800_7_60_BI	800	2	2	7	60

paw02a_800_7_70_BI	800	2	2	7	70

4.3 Performance Metrics

The confusion matrix [48] has shown in the **Table 2** and many metrics are based on this for effectiveness evaluation on classification problem. The matrix overall accuracy doesn't work on imbalanced datasets. Different metrics are given below. Sensitivity (Eq. 1) is known as true positive rate and Specificity (Eq. 2) is known as true negative rate, G – Mean (Eq. 3) is the geometric mean of sensitivity and specificity. F - measure (Eq. 6) is used to integrate precision (Eq. 4) and recall (Eq. 5) into a single metric for the support of the modelling and β is a coefficient to manage the relative importance of precision vs. recall [49] [50].

Table 2 Confusion matrix

	Actual_Positive	**Actual_Negative**
Predicted_Positive	TP (True Positive)	FP (False Positive)
Predicted_Negative	FN (False Negative)	TN (True Negative)

$$Sensitivity = \frac{TP}{TP+FN} \quad (1)$$

$$Specificity = \frac{TN}{TN+FP} \quad (2)$$

$$G - Mean = \sqrt{Sensitivity * Specificity} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F - Measure = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Recall + Precision},$$

Where, $\beta = 1$ (6)

5. Proposed Model for Imbalanced Dataset

The flow chart of the proposed method is shown in **Figure 6** to handle imbalanced datasets. There are three types of imbalanced datasets (**Table 1**) that are used as input to the model, then a Ten-fold cross-validation technique is chosen for examining the models [50] and getting target results from the models. The new balanced training set is found from the original imbalanced training set using the SMOTE method. After that, an extended binomial GLM and boosting method are combined to construct the model [51]. Finally, the results are analysed using the performance metrics.

5.1 Proposed Model for the Imbalanced Dataset in the Existence of Noisy and Borderline Examples

Figure 6 illustrates the framework for the SPIDER2-IPF model. **Table 1** shows the nine benchmark datasets used to test the model. As re-sampling, the SPIDER2 [13] [52] oversampling approach was applied. The IPF method [17] [53] is used in SPIDER 2 as a noise filter to reduce noise

before training base classifiers (boosting classifiers).

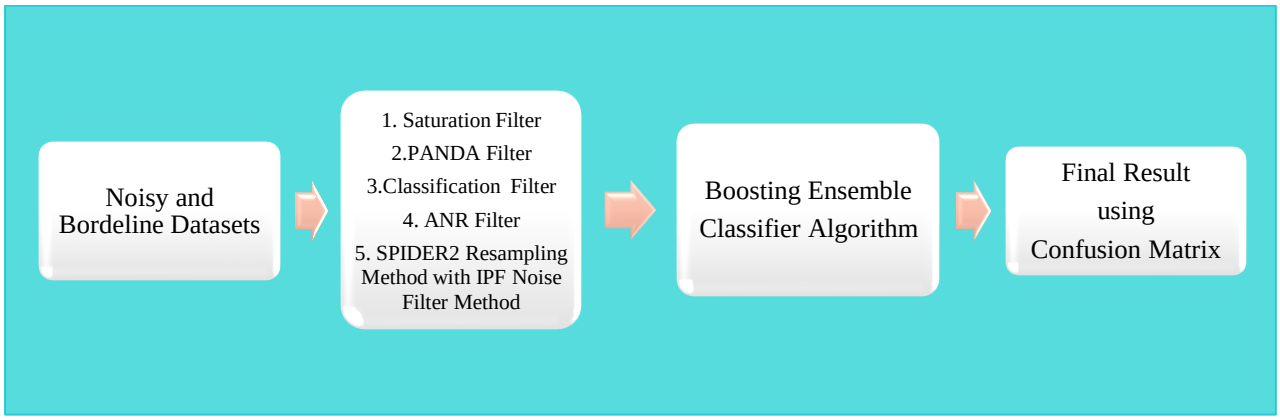

Figure. 6. Framework of SPIDER2-IPF model

Prior research on selective pre-processing by Stefanowski and Wilk using their method for selective pre-processing of unbalanced data [54] [55]. This methodology uses the "Edited Nearest Neighbor Rule (ENNR)" to define the local characteristic of instances, and then combines removing majority class objects that may result in misclassifying minority class objects with local over-sampling of minority class objects that are overwhelmed by surrounding majority class objects [56]. SPIDER2 is divided into two phases, one for majority classes and the other for minority classes. It determines the features of instances from the majority in the first phase, and then either removes or re-label the noisy examples from the majority, depending on the re-label option (i.e., reclassifies themselves as a minority). The second phase determines the characteristics of minority cases while taking into account the improvements made in the first phase. Then, depending to the amplification choice, noisy examples from the minority are amplified (by replicating them) [54] [57]. IPF is used as a noise filter to remove sounds before training base classifiers (boosting classifiers). With the IPF noise filtering approach, noisy samples are eliminated repeatedly until a termination condition is reached [58]. The main idea is to use the votes of numerous classifiers to filter out noisy samples [59] [60]. IPF is used to ensure that the examples removed in one iteration have no impact on detection in following iterations, as a result, noise filtering is more precise. Furthermore, IPF's ensemble nature allows it to collect predictions from multiple classifiers, which may result in a more accurate estimation of difficult noisy cases than collecting data from a single classifier. IPF also allows for the production of more diverse classifiers – for example, utilising random partitions – compared to other ensemble-based filters since it allows for greater flexibility when building the partitions from which these classifiers are constructed. When using ensembles of classifiers, creating variety among the classifiers built is crucial. IPF is easier since it just requires one classification method [53] [61]. Boosting attracts a large number of learners. Because the data samples are weighted, some of them may appear more frequently in new sets. The mis-predicted data points are discovered and their weights are increased in each iteration so that the following learner pays additional attention to get them

correctly [62] [63]. The stop criterion is set to $k = 3$ iterations, and the proportion of deleted examples is set to $p = 1\%$ in IPF's standard parameters.

6. Result Analysis and Discussion

By using the confusion matrix, **Table 3** has been prepared to show the result of the training datasets for the Saturation Filter Model. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **94.43**%, **34.79, 49.29**%, **81.94**%, **95.25**%, and **87.46**% respectively. **Table 4** has prepared to show the result of the testing datasets for the proposed method. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **97.34**%, **30.19**%, **43.79**%, **76.73**%, **97.29**%, and **84.72**% respectively.

Table 3

Results of the training datasets for the saturation filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	90.06	70.09	79.45	95.26	93.06	94.14
04clover5z_600_5-60-BI	95.24	06.54	24.95	82.23	95.24	88.25
04clover5z_600_5_70_BI	96.31	08.35	28.35	70.42	95.31	80.99
03subcl5_600_5_50_BI	90.66	05.74	22.81	70.14	95.66	80.93
03subcl5_600_5_60_BI	92.68	10.87	31.74	80.29	96.68	87.72
03subcl5_600_5_70_BI	97.44	06.37	24.91	79.03	96.44	86.87
paw02a_600_5_50_BI	95.90	53.14	71.38	95.58	96.90	96.23
paw02a_600_5_60_BI	95.16	06.45	24.77	73.45	95.26	82.94
paw02a_600_5_70_BI	94.17	86.29	90.14	95.16	96.17	95.66
04clover5z_800_7_30-BI	96.40	09.17	29.73	73.10	96.60	83.22
04clover5z_800_7_50_BI	96.01	03.33	17.88	58.19	96.01	72.46
04clover5z_800_7_60_BI	95.45	71.34	82.51	95.38	95.45	95.41
04clover5z_800_7_70_BI	93.40	06.28	24.21	69.52	93.60	79.78
03subcl5_800_7_30_BI	89.16	03.19	16.86	71.09	95.46	81.49
03subcl5_800_7_50_BI	94.73	14.02	36.44	85.41	96.46	90.59
03subcl5_800_7_60_BI	96.07	19.77	43.58	93.13	96.05	94.56
03subcl5_800_7_70_BI	95.62	92.23	93.90	96.62	96.43	96.52
paw02a-800_7_30_BI	98.38	83.45	90.60	95.43	94.63	95.02
paw02a_800_7_50_BI	93.63	06.17	24.03	54.31	97.34	69.72
paw02a_800_7_60_BI	94.34	84.63	89.35	94.62	92.39	93.49
paw02a_800_7_70_BI	92.23	83.21	87.60	92.41	89.16	90.75

Averages	**94.43**	**34.79**	**49.29**	**81.94**	**95.25**	**87.46**

Table 4
Results of the testing datasets for saturation filter the model

Datasets	**Sensitivity**	**Specificity**	**G-Mean**	**Precision**	**Recall**	**F-Measure**
04clover5z-600-5_50_BI	96.73	94.73	95.72	96.65	97.63	97.13
04clover5z_600_5-60-BI	97.32	07.03	26.15	78.31	96.35	86.39
04clover5z_600_5_70_BI	96.22	15.36	38.44	72.18	96.33	82.52
03subcl5_600_5_50_BI	96.40	04.50	20.82	69.10	97.30	80.81
03subcl5_600_5_60_BI	96.65	06.82	25.67	73.59	95.65	83.18
03subcl5_600_5_70_BI	96.32	04.62	21.09	73.64	97.32	83.84
paw02a_600_5_50_BI	94.13	93.32	93.72	98.30	95.15	96.69
paw02a_600_5_60_BI	97.27	04.75	21.49	72.79	95.14	82.47
paw02a_600_5_70_BI	97.63	30.12	54.22	97.56	97.64	97.60
04clover5z_800_7_30-BI	97.37	05.23	22.56	72.32	96.37	82.63
04clover5z_800_7_50_BI	98.13	02.37	15.25	53.17	98.11	68.96
04clover5z_800_7_60_BI	98.63	92.77	95.65	98.81	98.63	98.71
04clover5z_800_7_70_BI	97.37	03.31	17.95	74.60	97.38	84.48
03subcl5_800_7_30_BI	97.71	07.79	27.58	72.18	97.71	83.02
03subcl5_800_7_50_BI	98.85	89.53	94.07	98.75	98.84	98.79
03subcl5_800_7_60_BI	98.84	89.00	93.79	98.76	98.85	98.80
03subcl5_800_7_70_BI	98.71	68.60	82.28	98.43	98.71	98.56
paw02a-800_7_30_BI	97.71	03.34	18.06	52.06	97.71	67.92
paw02a_800_7_50_BI	97.12	02.59	15.86	49.16	97.12	65.27
paw02a_800_7_60_BI	98.08	02.69	16.24	56.01	98.08	71.30
paw02a_800_7_70_BI	97.07	05.53	23.16	55.02	97.07	70.23
Averages	**97.34**	**30.19**	**43.79**	**76.73**	**97.29**	**84.72**

Table 5 has been prepared to show the result of the training datasets for the proposed PANDA Filter Model. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **97.75%, 52.96%, 68.60%, 94.31%, 97.75%, and 95.41%** respectively. **Table 6** has prepared to show the result of the testing datasets for the proposed method. The average values of the sensitivity, specificity, G-Mean, precision,

recall, and F-Measure are **97.59%, 21.89%, 42.54%, 92.58%, 97.19%, and 94.59%** respectively.

Table 5

Results of the training datasets for the panda filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	98.35	37.63	60.83	95.26	98.34	96.77
04clover5z_600_5-60-BI	98.07	40.02	62.64	98.14	98.07	98.10
04clover5z_600_5_70_BI	97.51	90.68	94.03	97.34	97.51	97.42
03subcl5_600_5_50_BI	98.27	71.63	83.89	98.76	98.36	98.55
03subcl5_600_5_60_BI	97.18	17.65	41.41	96.85	96.55	96.69
03subcl5_600_5_70_BI	97.19	17.53	41.27	97.46	97.18	97.31
paw02a_600_5_50_BI	98.41	50.51	70.50	97.45	98.41	97.92
paw02a_600_5_60_BI	98.32	53.52	72.54	97.69	98.04	97.86
paw02a_600_5_70_BI	97.04	97.14	97.09	98.87	98.05	98.45
04clover5z_800_7_30-BI	97.94	67.23	81.14	98.32	97.84	98.07
04clover5z_800_7_50_BI	92.87	06.13	23.85	36.21	92.87	52.10
04clover5z_800_7_60_BI	98.33	63.30	78.89	98.37	98.33	98.35
04clover5z_800_7_70_BI	97.65	48.68	68.94	98.51	97.67	98.08
03subcl5_800_7_30_BI	97.68	79.66	88.21	98.77	97.68	98.22
03subcl5_800_7_50_BI	98.76	88.27	93.36	98.75	98.76	98.75
03subcl5_800_7_60_BI	98.57	52.22	71.74	98.11	98.57	98.33
03subcl5_800_7_70_BI	98.90	59.97	77.01	98.25	98.89	98.56
paw02a-800_7_30_BI	98.33	18.02	42.09	96.45	98.33	97.38
paw02a_800_7_50_BI	97.61	04.54	21.05	84.61	97.61	90.64
paw02a_800_7_60_BI	98.43	75.00	85.92	98.75	98.43	98.58
paw02a_800_7_70_BI	97.42	73.00	84.33	97.74	97.42	97.57
Averages	**97.75**	**52.96**	**68.60**	**94.31**	**97.75**	**95.41**

Table 6

Results of the testing datasets for the panda filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	98.31	12.11	34.50	94.16	98.31	96.19
04clover5z_600_5-60-BI	97.96	59.00	76.02	98.76	97.96	98.35
04clover5z_600_5_70_BI	97.31	06.03	24.22	83.29	97.31	89.75
03subcl5_600_5_50_BI	97.31	02.97	17.00	67.92	97.20	79.96
03subcl5_600_5_60_BI	96.69	59.00	75.52	98.86	96.69	97.76
03subcl5_600_5_70_BI	97.22	32.31	56.04	98.51	97.22	97.86
paw02a_600_5_50_BI	98.49	27.47	52.01	95.14	97.49	96.30
paw02a_600_5_60_BI	98.16	04.68	21.43	94.18	98.16	96.12
paw02a_600_5_70_BI	97.83	25.65	50.09	96.69	97.92	97.30
04clover5z_800_7_30-BI	96.97	32.31	55.97	98.66	96.89	97.76
04clover5z_800_7_50_BI	98.28	09.42	30.42	94.87	98.27	96.54
04clover5z_800_7_60_BI	98.15	23.71	48.24	96.92	98.14	97.52
04clover5z_800_7_70_BI	97.42	32.31	56.10	98.85	97.41	98.12
03subcl5_800_7_30_BI	97.21	05.93	24.00	87.17	97.21	91.91
03subcl5_800_7_50_BI	97.86	39.00	61.77	98.85	97.77	98.30
03subcl5_800_7_60_BI	98.65	14.23	37.46	93.30	98.65	95.90
03subcl5_800_7_70_BI	98.66	22.25	46.85	95.94	98.66	97.28
paw02a-800_7_30_BI	97.83	24.00	48.45	98.84	97.73	98.28
paw02a_800_7_50_BI	97.36	02.21	14.66	94.87	97.37	96.10
paw02a_800_7_60_BI	96.94	02.83	16.56	68.80	96.95	80.48
paw02a_800_7_70_BI	94.94	22.33	46.04	89.77	87.88	88.81
Averages	**97.59**	**21.89**	**42.54**	**92.58**	**97.19**	**94.59**

Table 7 has been prepared to show the result of the training datasets for the proposed Classification Filter Model. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **96.75%, 21.09%, 37.30%, 90.12%, 96.39%, and 91.03%** respectively. **Table 8** has prepared to show the result of the testing datasets for the proposed method. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **92.32%, 19.38%, 33.16%, 85.07%, 93.51%, and 87.15%** respectively.

Table 7

Results of the training datasets for the classification filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	95.34	04.34	20.34	95.15	96.34	95.74
04clover5z_600_5-60-BI	97.48	03.00	17.10	98.11	97.57	97.83
04clover5z_600_5_70_BI	96.68	43.94	65.17	98.67	96.68	97.66
03subcl5_600_5_50_BI	98.09	39.00	61.85	98.85	98.09	98.46
03subcl5_600_5_60_BI	96.29	04.44	20.67	96.95	96.29	96.61
03subcl5_600_5_70_BI	97.03	02.32	15.00	88.59	97.03	92.61
paw02a_600_5_50_BI	97.87	61.49	77.57	98.85	97.87	98.35
paw02a_600_5_60_BI	98.35	02.54	15.80	74.94	98.35	85.06
paw02a_600_5_70_BI	97.03	01.09	10.28	86.22	97.03	91.30
04clover5z_800_7_30-BI	96.58	22.07	46.16	98.33	96.58	97.44
04clover5z_800_7_50_BI	97.95	14.21	37.30	98.32	97.95	98.13
04clover5z_800_7_60_BI	98.56	94.23	96.37	98.94	98.56	98.74
04clover5z_800_7_70_BI	97.31	02.44	15.40	98.51	97.31	97.90
03subcl5_800_7_30_BI	97.59	25.87	50.24	96.00	97.59	96.78
03subcl5_800_7_50_BI	97.74	06.39	24.99	98.19	97.74	97.96
03subcl5_800_7_60_BI	92.74	01.88	13.20	01.39	92.73	02.73
03subcl5_800_7_70_BI	97.85	65.56	80.09	90.93	88.68	89.79
paw02a-800_7_30_BI	93.45	03.55	18.21	90.51	91.89	91.19
paw02a_800_7_50_BI	95.35	04.15	19.89	94.29	96.35	95.30
paw02a_800_7_60_BI	96.39	03.40	18.10	93.36	95.49	94.41
paw02a_800_7_70_BI	96.08	37.00	59.62	97.52	98.09	97.80
Averages	**96.75**	**21.09**	**37.30**	**90.12**	**96.39**	**91.03**

Table 8

Results of the testing datasets for the classification filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	90.21	02.81	15.92	95.09	96.12	95.60
04clover5z_600_5-60-BI	92.49	02.21	14.29	80.13	92.49	85.86

04clover5z_600_5_70_BI	93.52	04.80	21.18	92.39	93.35	92.86
03subcl5_600_5_50_BI	92.42	91.33	91.87	92.33	93.89	93.10
03subcl5_600_5_60_BI	95.50	06.31	24.54	90.03	93.51	91.73
03subcl5_600_5_70_BI	92.67	13.68	35.60	90.40	92.67	91.52
paw02a_600_5_50_BI	93.70	63.61	77.20	92.92	93.69	93.30
paw02a_600_5_60_BI	92.08	03.82	18.75	71.72	93.08	81.01
paw02a_600_5_70_BI	92.49	13.39	35.19	93.20	94.49	93.84
04clover5z_800_7_30-BI	93.48	03.14	17.13	89.03	93.67	91.29
04clover5z_800_7_50_BI	93.84	04.21	19.87	92.88	93.90	93.38
04clover5z_800_7_60_BI	95.47	83.27	89.16	95.79	96.47	96.12
04clover5z_800_7_70_BI	93.28	06.00	23.65	95.08	96.28	95.67
03subcl5_800_7_30_BI	90.59	83.69	87.07	90.15	91.59	90.86
03subcl5_800_7_50_BI	90.62	03.11	16.78	90.39	94.48	92.38
03subcl5_800_7_60_BI	92.46	02.95	16.51	04.84	93.46	09.20
03subcl5_800_7_70_BI	91.81	03.93	18.99	93.59	95.28	94.42
paw02a-800_7_30_BI	94.97	03.00	16.87	93.75	94.72	94.23
paw02a_800_7_50_BI	93.12	03.49	18.02	94.22	93.12	93.66
paw02a_800_7_60_BI	83.48	05.15	20.73	55.20	83.41	66.43
paw02a_800_7_70_BI	90.65	03.22	17.08	93.49	94.23	93.85
Averages	**92.32**	**19.38**	**33.16**	**85.07**	**93.51**	**87.15**

Table 9 has been prepared to show the result of the training datasets for the proposed ANR Filter Model. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **97.20%, 56.50%, 69.41%, 95.96%, 97.42%, and 96.65%** respectively. **Table 10** has prepared to show the result of the testing datasets for the proposed method. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **93.99%, 47.15%, 63.85%, 95.18%, 93.68%, and 94.30%** respectively.

Table 9

Results of the training datasets for the ANR filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	96.28	40.11	62.14	94.32	95.81	95.05
04clover5z_600_5-60-BI	97.15	43.42	64.94	93.12	93.24	93.18

04clover5z_600_5_70_BI	95.90	95.43	95.66	96.95	97.99	97.46
03subcl5_600_5_50_BI	96.82	97.14	96.97	97.96	97.99	97.97
03subcl5_600_5_60_BI	97.53	38.88	61.57	96.69	97.57	97.12
03subcl5_600_5_70_BI	97.14	37.16	60.08	97.17	97.72	97.44
paw02a_600_5_50_BI	96.91	95.00	95.95	97.14	97.91	97.52
paw02a_600_5_60_BI	95.91	95.01	95.45	96.99	96.90	96.94
paw02a_600_5_70_BI	97.03	02.07	14.17	86.21	97.03	91.30
04clover5z_800_7_30-BI	96.72	75.01	85.17	96.42	96.73	96.57
04clover5z_800_7_50_BI	97.49	48.36	68.66	98.12	98.49	98.30
04clover5z_800_7_60_BI	98.93	98.45	98.68	99.00	98.94	98.97
04clover5z_800_7_70_BI	98.35	97.33	97.83	98.95	98.35	98.64
03subcl5_800_7_30_BI	98.90	63.81	79.44	97.66	98.89	98.27
03subcl5_800_7_50_BI	98.93	57.14	75.18	97.60	98.93	98.26
03subcl5_800_7_60_BI	98.04	65.63	80.21	98.91	98.03	98.46
03subcl5_800_7_70_BI	98.71	55.73	74.16	98.16	98.71	98.43
paw02a-800_7_30_BI	95.71	55.74	73.04	96.16	96.71	96.43
paw02a_800_7_50_BI	98.16	13.17	35.95	96.17	98.24	97.19
paw02a_800_7_60_BI	97.51	01.20	10.81	86.09	95.51	90.55
paw02a_800_7_70_BI	93.11	10.77	31.66	95.46	96.16	95.80
Averages	**97.20**	**56.50**	**69.41**	**95.96**	**97.42**	**96.65**

Table 10

Results of the testing datasets for the ANR filter model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	95.42	53.20	71.24	95.10	96.81	95.94
04clover5z_600_5-60-BI	96.20	05.66	23.33	83.42	93.20	88.03
04clover5z_600_5_70_BI	95.11	45.12	65.50	97.10	96.12	96.60
03subcl5_600_5_50_BI	97.13	32.30	56.01	96.70	95.13	95.90
03subcl5_600_5_60_BI	96.36	23.00	47.07	95.35	93.33	94.32
03subcl5_600_5_70_BI	95.00	24.00	47.74	96.63	95.00	95.80
paw02a_600_5_50_BI	89.85	46.37	64.54	96.41	89.90	93.04
paw02a_600_5_60_BI	96.83	38.00	60.65	95.67	96.83	96.24
paw02a_600_5_70_BI	96.58	16.49	39.90	96.69	97.49	97.08

04clover5z_800_7_30-BI	96.42	65.66	79.56	97.56	96.41	96.98
04clover5z_800_7_50_BI	63.63	59.00	61.27	89.43	63.64	74.36
04clover5z_800_7_60_BI	95.69	36.36	58.98	96.92	98.71	97.80
04clover5z_800_7_70_BI	96.49	47.00	67.34	97.72	94.48	96.07
03subcl5_800_7_30_BI	97.76	61.62	77.61	97.28	96.76	97.01
03subcl5_800_7_50_BI	92.09	37.17	58.50	94.08	96.09	95.07
03subcl5_800_7_60_BI	92.60	93.21	92.90	96.91	93.69	95.27
03subcl5_800_7_70_BI	97.69	92.39	95.00	98.93	98.69	98.80
paw02a-800_7_30_BI	96.06	17.00	40.41	97.79	97.07	97.42
paw02a_800_7_50_BI	97.35	35.00	58.37	93.73	93.02	93.37
paw02a_800_7_60_BI	98.90	84.93	91.64	95.88	95.91	95.89
paw02a_800_7_70_BI	90.69	76.69	83.39	89.68	89.19	89.43
Averages	**93.99**	**47.15**	**63.85**	**95.18**	**93.68**	**94.30**

Table 11 has been prepared to show the result of the training datasets for the proposed SPIDER2-IPF Model. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **99.51%, 67.78%, 81.39%, 99.36 %, 99.63%, and 99.48%** respectively. **Table 12** has prepared to show the result of the testing datasets for the proposed method. The average values of the sensitivity, specificity, G-Mean, precision, recall, and F-Measure are **98.28%, 55.86%, 70.66%, 98.87%, 98.46%, and 98.65%** respectively.

Table 11

Results of the training datasets for the SPIDER2-IPF model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	99.49	77.93	88.05	99.88	99.56	99.71
04clover5z_600_5-60-BI	99.79	68.24	82.52	99.36	99.79	99.57
04clover5z_600_5_70_BI	99.98	50.26	70.88	98.88	99.98	99.42
03subcl5_600_5_50_BI	99.92	64.13	80.04	99.37	99.95	99.65
03subcl5_600_5_60_BI	99.99	59.85	77.35	99.15	99.99	99.56
03subcl5_600_5_70_BI	99.99	92.86	96.35	99.88	99.99	99.93
paw02a_600_5_50_BI	98.72	69.68	82.93	99.97	98.76	99.36
paw02a_600_5_60_BI	99.96	79.96	89.40	99.58	99.98	99.77
paw02a_600_5_70_BI	99.93	76.08	87.19	99.58	99.99	99.78

04clover5z_800_7_30-BI	99.83	59.92	77.34	99.36	99.88	99.61
04clover5z_800_7_50_BI	99.91	82.80	90.95	99.54	99.93	99.73
04clover5z_800_7_60_BI	99.94	98.72	99.32	99.96	99.89	99.92
04clover5z_800_7_70_BI	98.84	99.59	99.21	99.97	98.89	99.42
03subcl5_800_7_30_BI	99.94	59.17	76.89	98.65	99.93	99.28
03subcl5_800_7_50_BI	98.95	45.87	67.37	98.97	98.96	98.96
03subcl5_800_7_60_BI	98.76	43.26	65.36	98.96	99.66	99.30
03subcl5_800_7_70_BI	98.77	96.29	97.52	99.97	99.33	99.64
paw02a-800_7_30_BI	97.73	55.00	73.31	99.99	98.35	99.16
paw02a_800_7_50_BI	99.58	53.00	72.64	99.72	99.68	99.70
paw02a_800_7_60_BI	99.87	43.90	66.21	97.38	99.87	98.60
paw02a_800_7_70_BI	99.89	46.92	68.46	98.48	99.93	99.19
Averages	**99.51**	**67.78**	**81.39**	**99.36**	**99.63**	**99.48**

Table 12

Results of the testing datasets for the SPIDER2-IPF model

Datasets	Sensitivity	Specificity	G-Mean	Precision	Recall	F-Measure
04clover5z-600-5_50_BI	99.93	21.42	46.26	96.45	99.93	98.15
04clover5z_600_5-60-BI	98.63	84.34	91.20	99.94	98.66	99.29
04clover5z_600_5_70_BI	97.98	31.32	55.39	98.61	97.98	98.29
03subcl5_600_5_50_BI	96.00	73.06	83.74	98.69	96.00	97.32
03subcl5_600_5_60_BI	98.88	17.88	42.04	95.40	98.87	97.10
03subcl5_600_5_70_BI	97.74	39.60	62.21	98.86	98.75	98.80
paw02a_600_5_50_BI	96.75	49.00	68.85	99.86	98.76	99.30
paw02a_600_5_60_BI	97.42	13.76	36.61	99.97	98.43	99.19
paw02a_600_5_70_BI	99.83	94.34	97.04	99.92	99.76	99.83
04clover5z_800_7_30-BI	98.93	49.00	69.62	99.96	98.72	99.33
04clover5z_800_7_50_BI	97.95	23.26	47.73	99.23	97.93	98.57
04clover5z_800_7_60_BI	99.81	97.00	98.39	99.94	99.89	99.91

04clover5z_800_7_70_BI	97.58	99.91	98.73	99.89	97.55	98.70
03subcl5_800_7_30_BI	99.82	66.49	81.46	99.05	99.83	99.43
03subcl5_800_7_50_BI	98.99	11.50	33.74	98.16	96.99	97.57
03subcl5_800_7_60_BI	97.77	28.27	52.57	99.75	98.67	99.20
03subcl5_800_7_70_BI	97.12	85.80	91.28	93.76	97.13	95.41
paw02a-800_7_30_BI	96.77	89.91	93.27	99.67	97.81	98.73
paw02a_800_7_50_BI	98.71	25.22	49.89	99.66	98.71	99.18
paw02a_800_7_60_BI	99.97	97.58	98.76	99.94	99.97	99.95
paw02a_800_7_70_BI	97.50	74.47	85.21	99.72	97.33	98.51
Averages	**98.28**	**55.86**	**70.66**	**98.87**	**98.46**	**98.65**

6.1 Statistical Test

In this research, the nonparametric Wilcoxon signed rank test is used to compare sets of two or related samples [64]. When samples have small sizes and non-normal distributions, this method is applied. After computing the differences between the two procedures (with n objects), followed by calculating the absolute values, the results are sorted by removing the zero values. The rankings of the positive differences in R+ and the ranks of the negative differences in R- are added to get the values of R+ and R-. The p-value also provides useful information about the significance differences in the classifier, and the significance value "α" is fixed at 0.05. **Table 13** provides a summary of all computations. The techniques are displayed in the selection column based on whether the hypothesis is accepted or denied. Out of the investigated strategies, the suggested technique, "Hybrid SPIDER2-IPF Boosting Ensemble," is selected as the best one.

Table 13 Wilcoxon test (for pair wise comparison)

Comparison	**R^+**	**R^-**	**p-value**	**Hypothesis (α = 0.05)**	**Selection**
SPIDER2-IPF vs. Saturation	34	32	0.9468	Not Rejected	SPIDER2-IPF
SPIDER2-IPF vs. Panda	168	212	0.6653	Not Rejected	SPIDER2-IPF
Classification vs. SPIDER2-IPF	20	58	0.1413	Rejected	SPIDER2-IPF
ANR vs. SPIDER2-IPF	54	366	0.0037	*Rejected*	*SPIDER2-IPF*

7. Conclusions and Future Work

In many different industries, such as fraud detection, video surveillance, genetic data analysis, and many others, data imbalance is a critical issue. It could be due to a very expensive or difficult data collection approach, natural rarity, skewed and/or incomplete data sources, errors, or unequal sensor location. In these cases, the cost of misclassification is never balanced

for positive and negative instances, and the usual accuracy measure is inaccurate since it presupposes those true positives and true negatives are equally important. The topic of knowledge discovery using noisy and unbalanced data, which is of tremendous interest to practitioners and researchers in a range of domains, is thoroughly explored experimentally in this study. SPIDER2's superiority in controlling noisy instances within the majority class is demonstrated in this experiment (also accompanied with borderline ones). IPF's ensemble nature, which provides a reliable and accurate technique of detecting mis-labelled samples, the iterative noise detection and elimination processes used, and the flexibility to modify classifier diversity are all significant elements of IPF that result in a more accurate set of filters. Other sampling approaches, as well as various ensemble methods, may be explored in future study to handle imbalanced datasets.

The design of a more effective parameter and other ensemble techniques will be the main topics of future research. There are numerous intriguing prospective study areas for the suggested strategy. It is possible to test the filter's performance further. The classification of unbalanced data will continue to attract attention in both the "scientific and industrial" sectors due to the interesting topics and numerous future perspectives.

ACKOWLEDGEMENT

We are thankful to the Siksha ‘O’ Anusandhan Deemed to be University for this research work.

Funding: The research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

1. Chawla, Nitesh V., Nathalie Japkowicz, and Aleksander Kotcz. Special issue on learning from imbalanced data sets, ACM SIGKDD explorations newsletter, 6(1), 1-6, 2004.
2. Bhowan, Urvesh, Mark Johnston, and Mengjie Zhang. Developing new fitness functions in genetic programming for classification with unbalanced data, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2), 406-421, 2011.
3. Koh, Hian Chye, and Gerald Tan, Data mining applications in healthcare. Journal of healthcare information management, 19(2), 65, 2011.
4. Sun, Yanmin, Mohamed S. Kamel, Andrew KC Wong, and Yang Wang, Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40(12), 3358-3378, 2007.
5. Deng, Xiaogang, Xuemin Tian, Sheng Chen, and Chris J. Harris, Statistics local Fisher discriminant analysis for industrial process fault classification. In 2016 UKACC 11th International Conference on Control (CONTROL), 1-6, 2016.
6. Lewis, David D., and William A. Gale, A sequential algorithm for training text classifiers. In SIGIR’94, Springer, London, 3-12, 1994.
7. Chan, Philip K., Wei Fan, Andreas L. Prodromidis, and Salvatore J. Stolfo, Distributed data mining in credit card fraud detection. IEEE Intelligent Systems and Their Applications, 14(6), 67-74, 1999.

8. M. Kubat, R. Holte, and S. Matwin, Machine learning for the detection of oil spills in satellite radar images, Machine learning, 30(2), 195–215, 1998.
9. Khor, Kok-Chin, Choo-Yee Ting, and Somnuk Phon-Amnuaisuk, A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection. Applied Intelligence, 36(2), 320-329, 2012.
10. Sluban, Borut, and Nada Lavrač, Relating ensemble diversity and performance: A study in class noise detection. Neurocomputing, 160, 120-131, 2015.
11. Zhao, Jiakun, Ju Jin, Si Chen, Ruifeng Zhang, Bilin Yu, and Qingfang Liu. A weighted hybrid ensemble method for classifying imbalanced data, Knowledge-Based Systems, 106087, 2020.
12. Liu, Na, Xiaomei Li, Ershi Qi, Man Xu, Ling Li, and Bo Gao. A novel Ensemble Learning Paradigm for Medical Diagnosis with Imbalanced Data, IEEE Access, 8, 171263-171280, 2020.
13. Napierała, Krystyna, Jerzy Stefanowski, and Szymon Wilk, Learning from imbalanced data in presence of noisy and borderline examples. International Conference on Rough Sets and Current Trends in Computing, Springer, Berlin, Heidelberg, 158-167, 2010.
14. García, R. A. Mollineda and J. S. Sánchez, Theoretical Analysis of a Performance Measure for Imbalanced Data. 2010 20th International Conference on Pattern Recognition, Istanbul, 617-620, 2010, doi: 10.1109/ICPR.2010.156.
15. Sun, Jie, Zhiming Shang, and Hui Li. Imbalance-oriented SVM methods for financial distress prediction: a comparative study among the new SB-SVM-ensemble method and traditional methods, Journal of the Operational Research Society, 65(12), 1905-1919, 2014.
16. Van Hulse, Jason, and Taghi Khoshgoftaar, Knowledge discovery from imbalanced and noisy data. Data & Knowledge Engineering, 68(12), 1513-1542, 2009.
17. Batista, Gustavo EAPA, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD explorations newsletter, 6(1), 20-29, 2004.
18. B Seiffert, Chris, Taghi M. Khoshgoftaar, Jason Van Hulse, and Andres Folleco. An empirical study of the classification performance of learners on imbalanced and noisy software quality data, Information Sciences 259, 571-595, 2014.
19. Seiffert, Chris, Taghi M. Khoshgoftaar, Jason Van Hulse, and Andres Folleco, An empirical study of the classification performance of learners on imbalanced and noisy software quality data. Information Sciences, 259, 571-595, 2014.
20. Van Hulse, Jason, and Taghi Khoshgoftaar, Knowledge discovery from imbalanced and noisy data. Data & Knowledge Engineering, 68(12), 1513-1542, 2009.
21. Xiao, Jin, Changzheng He, Xiaoyi Jiang, and Dunhu Liu, A dynamic classifier ensemble selection approach for noise data. Information Sciences, 180(18), 3402-3421, 2010.
22. Seiffert, Chris, Taghi M. Khoshgoftaar, Jason Van Hulse, and Andres Folleco, An empirical study of the classification

performance of learners on imbalanced and noisy software quality data. Information Sciences, 259, 571-595, 2014.
23. Sáez, José A., Mikel Galar, JuliáN Luengo, and Francisco Herrera, Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness. Information Sciences, 247, 1-20, 2013.
24. Sáez, José A., Julián Luengo, Jerzy Stefanowski, and Francisco Herrera, SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. Information Sciences, Information Sciences, 291, 184-203, 2015.
25. Fernández, Alberto, Salvador Garcia, Francisco Herrera, and Nitesh V. Chawla, SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of artificial intelligence research, 61, 863-905, 2018.
26. Vorraboot, Piyanoot, Suwanna Rasmequan, Krisana Chinnasarn, and Chidchanok Lursinsap, Improving classification rate constrained to imbalanced data between overlapped and non-overlapped regions by hybrid algorithms. Neurocomputing, 152, 429-443, 2015.
27. Wang, Shuo, and Xin Yao, Diversity analysis on imbalanced data sets by using ensemble models. In 2009 IEEE symposium on computational intelligence and data mining, IEEE, 324-331, 2009.
28. Sabzevari, Maryam, Gonzalo Martínez-Muñoz, and Alberto Suárez, A two-stage ensemble method for the detection of class-label noise. Neurocomputing, 275, 2374-2383, 2018.
29. LóPez, Victoria, Alberto FernáNdez, MaríA José Del Jesus, and Francisco Herrera, A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. Knowledge-Based Systems, 38, 85-104, 2013.
30. Verbiest, Nele, Enislay Ramentol, Chris Cornelis, and Francisco Herrera, Preprocessing noisy imbalanced datasets using SMOTE enhanced with fuzzy rough prototype selection. Applied Soft Computing, 22, 511-517, 2014.
31. Pan, Shirui, Jia Wu, Xingquan Zhu, and Chengqi Zhang, Graph ensemble boosting for imbalanced noisy graph stream classification. IEEE transactions on cybernetics, 45(5), 954-968, 2014.
32. Lee, Han Kyu, and Seoung Bum Kim, An overlap-sensitive margin classifier for imbalanced and overlapping data. Expert Systems with Applications, 98, 72-83, 2018.
33. Yu, Hualong, and Jun Ni, An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data. IEEE/ACM transactions on computational biology and bioinformatics, 11(4), 657-666, 2014.
34. http://sci2s.ugr.es/keel/datasets.php.
35. Han, Jiawei, Jian Pei, and Micheline Kamber. Data mining: concepts and techniques. Elsevier, 2011.
36. D. Gamberger, N. Lavrac, S. Dzroski, Noise detection and elimination in data preprocessing: Experiments in medical domains. Applied artificial intelligence, vol. 14, no. 2, pp. 205-223, 2000.

37. J.D. Hulse, T.M. Khoshgoftaar, H. Huang. The pairwise attribute noise detection algorithm. Knowledge and Information Systems, vol. 11, no. 2, pp. 171-190, 2007.
38. D. Gamberger, N. Lavrac, C. Groselj, Experiments with noise filtering in a medical domain. 16th International Conference on Machine Learning (ICML99), San Francisco, USA, pp. 143-151, 1999.
39. X. Zeng, T. Martinez, A Noise Filtering Method Using Neural Networks. IEEE International Workshop on Soft Computing Techniques in Instrumentation, Measurement and Related Applications (SCIMA2003), Utah USA, pp. 26-31, 2003.
40. K. Napierala, J. Stefanowski, S, Wilk. Learning from Imbalanced Data in Presence of Noisy and Borderline Examples. 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC2010), Warsaw Poland, pp. 158-167, 2010.
41. T.M. Khoshgoftaar, P. Rebours, Improving software quality prediction by noise filtering techniques. Journal of Computer Science and Technology, vol. 22, pp. 387-396, 2007.
42. Zeraatkar, S. and Afsari, F., 2021. Interval–valued fuzzy and intuitionistic fuzzy–KNN for imbalanced data classification, Expert Systems with Applications, 184, p.115510, 2021.
43. J. R. Quinlan. C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.
44. Gong, Joonho, and Hyunjoong Kim, RHSBoost: Improving classification performance in imbalance data. Computational Statistics & Data Analysis, 111, 1-13, 2017.
45. Japkowicz, Nathalie, Class imbalances: are we focusing on the right issue. Workshop on Learning from Imbalanced Data Sets II, 1723, 63, 2003.
46. L'heureux, Alexandra, Katarina Grolinger, Hany F. Elyamany, and Miriam AM Capretz., Machine learning with big data: Challenges and approaches. Ieee Access, 5, 7776-7797, 2017.
47. M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection. In: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann, 179–186, 1997.
48. He, Haibo, and Edwardo A. Garcia, Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284, 2009.
49. Kubat, Miroslav, and Stan Matwin, Addressing the curse of imbalanced training sets: one-sided selection. Icml, 97, 179-186. 1997.
50. Robertson, Stephen E., Cornelis J. van Rijsbergen, and Martin F. Porter, Probabilistic models of indexing and searching. Proceedings of the 3rd annual ACM conference on Research and development in information retrieval, 35-56. 1980.
51. Tan, Pang-Ning, Michael Steinbach, and Vipin Kumar, Introduction to data mining. Pearson Education India, 2016.
52. Qian, Yun, Yanchun Liang, Mu Li, Guoxiang Feng, and Xiaohu Shi, A resampling ensemble algorithm for classification of imbalance problems. Neurocomputing, 143, 57-67, 2014.
53. Van Hulse, Jason, Taghi M. Khoshgoftaar, and Amri Napolitano, A novel noise filtering algorithm for imbalanced data. In 2010 Ninth

International Conference on Machine Learning and Applications, IEEE, 9-14, 2010.
54. Branco, Paula, Luís Torgo, and Rita P. Ribeiro, A survey of predictive modeling on imbalanced domains. ACM Computing Surveys (CSUR), 49(2), 1-50, 2016.
55. Stefanowski, Jerzy, and Szymon Wilk, Improving rule based classifiers induced by MODLEM by selective pre-processing of imbalanced data. Proc. of the RSKD Workshop at ECML/PKDD, Warsaw, 54-65. 2007.
56. Stefanowski, Jerzy, and Szymon Wilk, Selective pre-processing of imbalanced data for improving classification performance. International Conference on Data Warehousing and Knowledge Discovery, Springer, Berlin, Heidelberg, 283-292. 2008.
57. Laurikkala, Jorma, Improving identification of difficult small classes by balancing class distribution. Conference on Artificial Intelligence in Medicine in Europe, Springer, Berlin, Heidelberg, 63-66, 2001.
58. Han, Hui, Wen-Yuan Wang, and Bing-Huan Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In International conference on intelligent computing, Springer, Berlin, Heidelberg, 878-887, 2005.
59. Chen, XiaoShuang, Qi Kang, MengChu Zhou, and Zhi Wei, A novel under-sampling algorithm based on iterative-partitioning filters for imbalanced classification. In 2016 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, 490-494, 2016.
60. Khoshgoftaar, Taghi M., and Pierre Rebours, Improving software quality prediction by noise filtering techniques. Journal of Computer Science and Technology, 22(3), 387-396, 2007.
61. Brodley, Carla E., and Mark A. Friedl, Identifying mislabeled training data. Journal of artificial intelligence research, 11, 131-167, 1999.
62. Kuncheva, Ludmila I, Diversity in multiple classifier systems. Information fusion 6(1), 3-4, 2005.
63. Zhou, Zhi-Hua, Ensemble methods: foundations and algorithms. Chapman and Hall/CRC, 2019.
64. Jacobson, James E, The Wilcoxon two-sample statistic: tables and bibliography. Journal of the American Statistical Association, 58(304), 1086-1103, 1963.