Kecia Gomes de Moura

LABEL NOISE DETECTION UNDER NOISE AT RANDOM MODEL WITH ENSEMBLE FILTERS

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2019

Kecia Gomes de Moura

LABEL NOISE DETECTION UNDER NOISE AT RANDOM MODEL WITH ENSEMBLE FILTERS

A M.Sc. Dissertation presented to the Center of Informatics of Universidade Federal de Pernambuco in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Main Area: Machine Learning
Advisor: Ricardo Bastos Cavalcante Prudêncio
Co-Advisor: George Darmiton da Cunha Cavalcanti

Recife
2019

Catalogação na fonte
Bibliotecária Mariana de Souza Alves CRB4-2106

M929l Moura, Kecia Gomes de

Label noise detection under Noise at Random model with ensemble filters – 2019.

115f.: il., fig., tab.

Orientador: Ricardo Bastos Cavalcante Prudêncio

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da computação. Recife, 2019.

Inclui referências e apêndices.

1. Machine Learning. 2. Detecção de Ruído. 3. Combinação de Classificadores. 4. Ruído Aleatório. I. Prudêncio, Ricardo Bastos Cavalcante (orientador). II. Título.

006.31 CDD (22. ed.) UFPE-MEI 2019-152

Kecia Gomes de Moura

“Label noise detection under Noise at Random model with ensemble filters”

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 15/03/2019.

BANCA EXAMINADORA

____
Prof. Dr. Cleber Zanchettin
Centro de Informática / UFPE

____
Profa. Dra. Anne Magaly de Paula Canuto
Departamento de Informática e Matemática Aplicada/UFRN

____
Prof. Dr. Ricardo Bastos Cavalcante Prudêncio
Centro de Informática / UFPE

To my family.
Especially to my brother, Bruno,
and to my sister(-in-law), Mari.

ACKNOWLEDGEMENTS

First of all, I want to express my gratitude to my advisors Prof. Ricardo Prudêncio and Prof. George Darmiton who have supported and helped me with their invaluable instructions, guidance, and patience throughout the completion of this dissertation. Their advice and expertise were of great importance to the development of this work. Thank you very much for sharing your knowledge, for being so attentive and solicitous. I could not imagine having better mentors for my dissertation.

I want also to express my gratitude to my parents back home in Ipubi-PE, without them, I might never have been able to even become a master student. I want to thank my brothers, Raoni and Bruno, and also their spouses, Lizandra and Mariana for the moments of joy and support in our reunions.

My terrific friends and also labmates, Bruna, Chaina and Renan, were equally invaluable in helping me through this process. In classes, in pubs (Peludo's especially), through countless conversations (on our bench), they provided feedback on my thinking, advice for how to approach various aspects of scholarship and life. Thank you all for the patience, support, and humor during the years of graduate school.

To all teachers of Centro de Informática (Cin) that, in one way or another, contributed to my academic and personal growth.

To my examination committee Cleber Zanchettin and Anne Canuto for their availability and for all comments and suggestions to improve this work.

To the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) for the financial support.

Finally, to all people who helped me directly or indirectly to the achievement of my goals.

ABSTRACT

Label noise detection has been widely studied in Machine Learning due to its importance to improve training data quality. Satisfactory noise detection has been achieved by adopting an ensemble of classifiers. In this approach, an instance is assigned as mislabeled if a high proportion of members in the pool misclassifies that instance. Previous authors have empirically evaluated this approach with results in accuracy, nevertheless, they mostly assumed that label noise is generated completely at random in a dataset. This is a strong assumption since there are other types of label noise which are feasible in practice and can influence noise detection results. This work investigates the performance of ensemble noise detection in two different noise models: the Noisy at Random (NAR), in which the probability of label noise depends on the instance class, in comparison to the Noisy Completely at Random model, in which the probability of label noise is completely independent. In this setting, we also investigate the effect of class distribution on noise detection performance, since it changes the total noise level observed in a dataset under the NAR assumption. Further, an evaluation of the ensemble vote threshold is carried out to contrast with the most common approaches in the literature. Finally, it is shown in a number of performed experiments that the choice of a noise generation model over another can lead to distinct results when taking into consideration aspects such as class imbalance and noise level ratio among different classes.

Keywords: Noise Detection. Label Noise. Noise at Random. Ensemble. Classification Filtering.

RESUMO

A detecção de ruído de dados tem sido amplamente estudada em Aprendizagem de Máquina devido à sua importância para melhorar a qualidade dos dados de treinamento. Uma detecção de ruído satisfatória tem sido conseguida através da utilização de um conjunto de classificadores (ensemble). Nessa abordagem, uma instância é considerada como rotulada erroneamente se uma alta proporção de classificadores a classificarem incorretamente. Trabalhos anteriores avaliaram empiricamente esta abordagem obtendo resultados na acurácia. No entanto, a maioria deles, assumem que o ruído de rótulo é gerado completamente ao acaso em um conjunto de dados. Essa suposição singular pode induzir em erro ou a resultados incompletos uma vez que existem outros tipos de ruídos de rótulo que são viáveis na prática e podem influenciar os resultados de detecção. Este trabalho investiga o desempenho da detecção de ruído levando em consideração o modelo "Noisy at Random" (NAR), no qual a probabilidade de ruído de rótulo depende da classe da instância, em comparação ao modelo "Noisy Completely at Random" (NCAR), em que o ruído de rótulo é totalmente aleatório. Nesse cenário, também investigamos o efeito do desbalanceamento de classes no desempenho da detecção de ruído, uma vez que essa desproporção altera o nível total de ruído observado quando há a suposição de NAR. Além disso, uma avaliação do limiar para a votação do ensemble é realizada para contrastar com as abordagens mais comuns na literatura. Finalmente, é demonstrado em vários experimentos realizados que a escolha por um modelo de geração de ruído em detrimento de outro pode levar a resultados distintos considerando-se aspectos como desbalanceamento de classes e proporção de ruído em cada classe.

Palavras-chaves: Detecção de Ruído. Combinação de Classificadores. Ruído Aleatório. Ruído de Classe.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CF	*Classification Filter*
CLCH	*Complexity of the Least Correct Hypothesis*
CVCF	*Cross-validated Committees Filter*
DT	*Decision Tree*
EF	*Ensemble Filter*
EM	*Expectation Maximization*
ENN	*Edited Nearest Neighbor*
IPF	*Iterative-Partitioning Filter*
IR	*Imbalance Ratio*
M	*Noise Ratio*
ML	*Machine Learning*
NAR	*Noisy at Random*
NCAR	*Noisy Completely at Random*
NF	*Noise Filtering*
NNAR	*Noisy not at Random*
p	*Noise levels in dataset*
RF	*Random Forest*
SF	*Saturation Filter*
SMOTE	*Synthetic Minority Over-sampling Technique*

CONTENTS

1 INTRODUCTION

Data quality is of great importance for *Machine Learning* (ML) applications and, in particular, for classification tasks. Conventionally in these tasks, a training set of labeled instances is given as input to an ML algorithm, which will acquire useful knowledge to make predictions for new instances. In practice, real-world datasets frequently contain irregularities such as incompleteness, noise and data inconsistencies, which can impact ML performance (HAN; KAMBER; PEI, 2012). In this light, noise detection and filtering are quite relevant tasks for ML (ZHU; WU, 2004).

According to the literature, noise may occur in both attributes and classes (ZHU; WU, 2004). This work focuses on the latter problem, in which an unknown proportion of instances in a dataset are mislabeled due to different reasons. This is a relevant problem since label noise can harm the identification of true class boundaries in a problem, increase the chance of overfitting and affect learning performance in general (FRENAY; VERLEYSEN, 2014).

Previous works adopted the *classification noise filtering* approach (BRODLEY; FRIEDL, 1999)(SLUBAN; LAVRA, 2015a)(GUAN et al., 2018) for label noise detection, which is really widespread in the literature. In this approach, mislabeled instances in a dataset are identified according to the output results of a classifier or an ensemble of classifiers. For example, in the majority vote for ensemble noise detection, an instance is marked as mislabeled if most classifiers in a pool incorrectly classify the instance. In the consensus vote for ensemble noise detection, in turn, an instance is considered as noisy if all classifiers in the pool misclassify it.

In real-world, to evaluate whether an instance from a dataset is noisy or not is usually necessary the help of a domain specialist, which is not always available. Moreover, the preprocessing step cost and duration may increase when depending on a specialist judgment. This problem is mitigated when artificial datasets are used, or simulated noise is injected into a dataset in a controlled way. The validation of noise detection techniques and the study on how noise influence the learning process is simplified when a systematic addition of noise is performed (GARCIA et al., 2019).

Injecting label noise usually follows two main approaches: random, in which each instance has the same probability of having its label exchanged by another one, or in a pairwise way, in which the majority class examples have their labels modified to the same label of the second majority class (SAEZ et al., 2015)(GARCIA et al., 2019). Label noise can be modeled in data in three main ways: (1) *Noisy Completely at Random* (NCAR), in which the probability of an instance being noisy is random, (2) *Noisy at Random* (NAR), the probability of an instance being noisy depends on its label, and (3) *Noisy not at Random* (NNAR), the probability of an instance being noisy depends also on its attributes (FRENAY; VERLEYSEN, 2014).

In many previous works (SLUBAN; LAVRA, 2015a) (BRODLEY; FRIEDL, 1999) (SAEZ et al., 2015) (GARCIA et al., 2019), one type of noise is chosen over another to model the experiments. Nevertheless, it is unknown how this choice can impact the results. In this work, it is investigated how noise models can influence the noise detection under different contexts such as noise level, class imbalance, and noise distribution. It is shown that different results are achieved depending on the context.

This chapter is structured as follows. In Section 1.1, the main problems and motivation of this research are presented. In Section 1.2, the goals and contributions are delineated. The research methodology is described in Section 1.3. Finally, in Section 1.4, it is shown how this dissertation is organized.

1.1 MOTIVATION AND PROBLEM STATEMENT

Many works on noise detection have investigated and presented noise handling techniques in a variety of scenarios such as large datasets, imbalanced data, noise in and out borderline examples and different percentage of noise (FRENAY; VERLEYSEN, 2014)(SLUBAN; LAVRA, 2015b)(ZHU X., 2003)(SAEZ et al., 2015). In order to do so, it is usually followed a step of noise injection in the research experiments in which a type of noise model is chosen. For instance, in Sluban, Gamberger e Lavrač (2014), label noise was randomly injected in dataset (NCAR model) so to measure the performance of the proposed *NoiseRank* method. In a different approach, in Saez et al. (2015), label noise was injected according to each class (NAR model) to measure the *SMOTE-IPF* approach. On the other hand, in Bootkrajang (2016), an feature-dependent label noise (NNAR model) was employed to evaluate the *gLR* method.

The main focus of these studies has been to improve the performance in classification or in detecting noise itself. While important results have been achieved in these previous works, questioning how the choice of one noise model over another can influence the results, if so, seems to be pertinent. Furthermore, given the well-known and common problem of imbalanced datasets, in which a standard classifier is biased toward the majority class due to its representation inside the dataset, it is also important to analyze the influence of different noise models in this scenario.

Once the noise is generated under different assumptions on how it is distributed in the instance space (being more or less adequate depending on the application) the performance of the noise detectors may strongly depend on such noise distributions. For instance, it may be more difficult for a human supervisor in some contexts to correctly label instances in the minority class. The impact of NAR on noise detection techniques may depend on which class is noisier and how noisy it is. For instance, in imbalanced class problems, an ensemble noise detector may be very effective to detect noise in the majority class since the ensemble tends to have a better predictive behavior on this class (GALAR A. FERNANDEZ;

HERRERA, 2012). However, if the noise level in the majority class is not high, the good quality of detection, in this case, may not be relevant. On the other hand, a fail in detecting noise in the minority class may be critical.

In addition, notwithstanding that there are several noise handling techniques, ensemble-based noise filters are widespread in the literature and they generally apply two voting schemes to identify noisy examples: consensus and majority (BRODLEY; FRIEDL, 1999)(ZHU X., 2003)(SLUBAN; LAVRA, 2015b)(SAEZ et al., 2015)(GARCIA et al., 2019). Useful insights have been attained regarding these two approaches, for instance, the consensus vote tends to achieve high levels of precision but low recall. The majority vote, in turn, may achieve high levels of recall depending on the pool diversity (SLUBAN; LAVRA, 2015a).

Nevertheless, few works also investigated the influence of ensemble threshold variations, i.e., the impact of varying the number of erroneous ensemble predictions used to identify an instance as noisy. Previous works (KHOSHGOFTAAR; ZHONG; JOSHI, 2005)(SABZEVARI; MARTINEZ-MUNOZ; SUAREZ, 2018) showed that selecting adequate values of the ensemble threshold for noise filtering is superior to using standard filtering at all noise levels. In this way, noise distribution may also influence the optimal ensemble threshold and studying NCAR and NAR impact on these conditions may be pertinent.

1.2 OBJECTIVES

In contrast to previous studies, in this work, the influence of distinct label noise models on ensemble noise detection is investigated. The main objective is to analyze how choosing one noise model generator over another can influence the results under different contexts such as class imbalance, noise distribution, ensemble thresholds and percentage of noise in data.

To that end, the specific objectives were defined in order to answer the following questions:

- Is noise detection affected by the type of noise model assumed?
- Regarding the NAR model: how different ratios of noise distribution influence the detection (if so)?
- How is detection performance impacted under NCAR and NAR when dealing with imbalanced datasets?
- Is the majority/consensus always the best vote approach? How varying the ensemble threshold would impact on noise detection?

1.3 RESEARCH METHODOLOGY

In order to investigate how noise generation models can influence label noise detection, a variety of scenarios is created taking into consideration variables such as the amount of noise in data, noise distribution among classes and class imbalance ratio.

The research is divided into two main parts. First, experiments are run on synthetic datasets generated properly to produce a more controlled environment on which analysis is performed and results measured. Later, the experiments are run on real-world datasets to validate previous findings.

In order to answer the questions raised in previous section, we attack only binary problems (minority vs majority) applying three different types of noise levels: (1) NCAR; (2) NAR, by injecting a high proportion of label noise in the majority class; (3) NAR, by injecting a high proportion of label noise in the minority class. Noise is injected in balanced and imbalanced data and the ensemble noise filter with the majority vote is used as the noise detector. Later, different thresholds for ensemble vote is assessed as well. Analysis and discussion are conducted through graphical analysis on the noise detection performance using most common measures in the literature.

1.4 ORGANIZATION OF THE DISSERTATION

The remainder of this Dissertation is organized as follows: In Chapter 2, an overview of label noise detection along with the main *Noise Filtering* (NF) techniques are presented. In the same chapter, the ensemble-based filter is detailed and the different types of noise models are defined.

With the most important concepts established in Chapter 2, the proposed methodology is described in Chapter 3. The synthetic and real-world datasets, the ensemble of algorithms and noise injection approach are also introduced in the chapter.

Experiments are conducted in Chapter 4. The results are then evaluated and discussed in the chapter according to the research methodology presented previously.

Lastly, the main points presented in this dissertation are summarized in Chapter 5. The conclusions derived from the experimental results are summarized and this work's contributions are outlined. Finally, future works are suggested at the end of the chapter.

2 LABEL NOISE DETECTION

In Machine Learning, predicting the real classes of some sample is called classification. For this, a classifier (algorithm) is trained (learning process) so to infer the labels of new data. What occurs, eventually, is that the training data may be polluted with wrong values named noise.

The presence of noise in the training dataset can hinder ML models induction with an increase in processing time, a higher complexity, overfitting of the induced model and also harm the predictive performance (LORENA; CARVALHO, 2004).

In Zhu e Wu (2004), for supervised learning datasets, two types of noise are distinguished: attribute (or feature) and class (or label) noise. The former is present in one or more features as a result of absent, incorrect or missing values. The latter can be caused by errors or by the use of wrong information in the labeling process.

According to the literature, the removal of examples with feature noise is not as useful as label noise detection. This occurs since the values of non-noisy features can be helpful in the classification process and also because there is only one label while there are many attributes (FRENAY; VERLEYSEN, 2014). Besides, feature noise can later lead to label noise. Hence, this work will concentrate on the label noise problems. Hereinafter, label noise is also referred to as noise.

In theory, noise detection requires a verification step, in which examples marked as noisy are confirmed by a domain expert before they can be further processed as applied in Sluban, Gamberger e Lavrač (2014). Since eliminating noisy data is the common approach, it is important to differentiate these data from the noiseless data, which should be preserved as they have features that represent the knowledge required for an adequate model induction.

In real-world applications, to evaluate whether an example is noisy or not generally requires the examination of domain specialists. Nonetheless, this is not always feasible as they may not be available. Moreover, the need for consulting a specialist tends to increase the duration and cost of the preprocessing step. This problem is mitigated when artificial datasets are used, or simulated noise is injected into a dataset in a controlled way. The study and further validation of noise detection techniques and noise models influence on the learning process are simplified when a systematic addition of noise is performed.

In order to do so, it is imperative to choose the method by which the noise will be inserted into a dataset. Usually, three types of models are chosen: NCAR (SLUBAN; GAMBERGER; LAVRAČ, 2014), NAR (SAEZ et al., 2015) and/or NNAR (BOOTKRAJANG, 2016). Whatever the model employed to inject noise to a dataset, it is necessary to corrupt the examples within a given rate (percentage of noise), and, due to its stochastic nature, this injection is normally repeated a number of times for each noise level.

In the next sections, we present the background information necessary to describe our studies and analysis: in Section 2.1, label noise detection is reviewed and well-known techniques in the literature are presented. In Section 2.2, the chosen label noise detection technique for this work is explained. The main different label noise models are outlined in Section 2.3. Finally, in Section 2.4, the metrics used for noise detection evaluation are described.

2.1 DETECTION APPROACHES

In the literature, there are several techniques for label noise detection. According to Frenay e Verleysen (2014), two ways of implementing these techniques are: (1) designing classifiers that are more robust and noise-tolerant and (2) performing data cleaning by filtering instances as a preprocessing step.

While the first approach aims to implement robust models by using some available information related to the noise present in data, the preprocessing step, on the other hand, usually involves the application of noise filtering techniques to identify the presence of noise.

Therefore, we may say that the first is an algorithm-level approach, which relies on algorithms that are naturally robust to label noise, or algorithms that directly model noisy data during the learning process.

Likewise, the second is a data-level approach, in which noisy data is handled prior to the training process such as data filter implementations. The separation of noise filtering and the learning phase has the advantage of avoiding the use of polluted instances in the classifier building process. In addition, filter approaches are cheap and easy to implement (FRENAY; VERLEYSEN, 2014). Our work is included in this category.

2.1.1 Algorithm-level Approach

One way of building models to overcome noise problems is by including the noise information during the learning process. In Bootkrajang (2016), for example, it is proposed the *generalized robust Logistic Regression* (gLR) to tackle not only problems with NCAR and NAR but also with NNAR noise models. For this, it was employed the probability density function of the exponential distribution to model noises in a scenario where points that are closer to the decision boundary have a relatively higher chance of being mislabelled than those that live further away.

There are some notable model-based approaches proposed early for dealing with NCAR and NAR noise models such as *robust Kernel Fisher Discriminant* (LAWRENCE; SCHöLKOPF, 2001), in which is proposed a method that associates to each example a probability of the label being noisy, applying an *Expectation Maximization* (EM) algorithm for updating the probabilities (later improved in Li et al. (2007) for more complex datasets), and *robust kernel logistic regression* (BOOTKRAJANG; KABAN, 2014) in which the optimal hyper-parameters for the method are automatically determined using Multiple Kernel Learning and Bayesian regularisation techniques.

Recently, studies on NNAR problems have gained attention and extensions of previous works have been proposed to tackle the associated issues. In Bootkrajang (2016), for instance, a logistic regression classifier employing a noise model based on a mixture of Gaussians is proposed.

Another approach consists in modifying the learning algorithm to reduce the influence of label noise. In Biggio, Nelson e Laskov (2011), for example, a model is proposed (*Label Noise robust SVMs*) for the analysis of label noise in support vector learning in which it is developed a modification of the SVM that indirectly compensates for the noise present in data by correcting the kernel matrix of SVM with a specially structured matrix based on the information regarding the level of noise in data.

The aforementioned approaches directly model label noise during the learning process. Although the advantage of this approach is to use prior knowledge regarding a noise model and its consequences (FRENAY; VERLEYSEN, 2014), they increase the complexity of learning algorithms and can lead to overfitting, because of the additional parameters of the training data model.

Some approaches are naturally tolerant to noise and this can be used as a benefit. Ensemble methods like *bagging* have the diversity increased in the presence of noise what help to cope with mislabeled examples. The *Decision Tree* (DT) pruning process is also more robust to noisy data as it has been shown that this technique decreases the influence of label noise once it prevents data overfitting (ABELLÁN; MASEGOSA, 2010).

2.1.2 Data-level Approach

Another way of overcoming the noise problem is to improve the quality of training data before using it in the classification process. This improvement is accomplished through the application of filtering techniques that clean data by removing possible noises. Upon completion of the filtering step, which delivers a noiseless dataset, the training data is ready to be used.

In this case, noisy labels are detected and dealt in a preprocessing step and mislabeled instances can either be relabeled or simply removed (Garcia; Lorena; Carvalho, 2012). The general procedure is depicted in Figure 1.

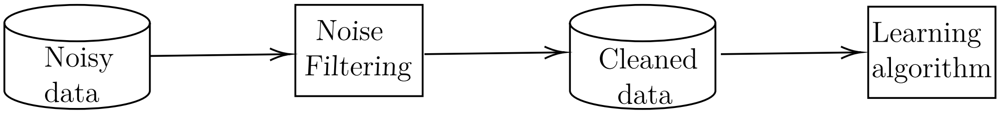

Figure 1 – General procedure for improving data quality applying noise filtering (FRENAY; VERLEYSEN, 2014).

Various studies have showed that using class NF techniques can lead to a better classification performance and also reduce classifiers complexity (Garcia; Lorena; Carvalho, 2012)(SLUBAN; GAMBERGER; LAVRAČ, 2014) (SAEZ et al., 2015). For this, there are, in the literature, many different NF approaches, i.e, different procedures are followed to decide whether a certain information should be treated as noise or not.

NF can be performed, for example, by using complexity measures, in which instances are removed when the values exceed a predefined threshold (SUN et al., 2007) (SMITH; MARTINEZ; GIRAUD-CARRIER, 2014); using the prediction of a classifier or an ensemble of classifiers, in which instances are removed when a certain number of algorithms misclassifies them (YUAN et al., 2018); partitioning approaches for removing mislabeled instances for large datasets (ZHU X., 2003)(GARCIA-GIL et al., 2019); filtering noisy instances by verifying the impact of the removal on learning process (MALOSSINI; BLANZIERI; NG, 2006); using *k-NN* algorithms to remove instances that are distant from the ones of same class (WILSON; MARTINEZ, 2000), among others.

The highlighted NF in the literature are following summarized:

- *Edited Nearest Neighbor* (ENN) (KANJ et al., 2016). This algorithm eliminates instances whose class does not match the majority of its k-nearest neighbors.

- *Classification Filter* (CF) (GAMBERGER et al., 1999). The training dataset is divided into n subsets. A set of classifiers is trained based on the union of any $n - 1$ subsets. The examples misclassified in the remaining subset are then removed from the training dataset.

- *Ensemble Filter* (EF) (SLUBAN; LAVRA, 2015a)(YUAN et al., 2018). The training dataset is classified using n-fold cross-validation with various different classifiers. Then, a vote combination is applied (usually consensus or majority) to decide which examples will be eliminated.

- *Iterative-Partitioning Filter* (IPF) (Khoshgoftaar; Rebours, 2004). Noisy instances through multiple iterations are removed. In each iteration, the training dataset is divided into n subsets, and a DT is built over each of these subsets to evaluate all the instances. Then, the misclassified instances are removed (using the consensus or majority voting scheme), and a new iteration is started.

- *Synthetic Minority Over-sampling Technique* (SMOTE)-IPF (SAEZ et al., 2015). Synthetic Minority Over-sampling Technique is combined with the IPF approach for dealing with noisy examples in imbalanced datasets.

- High Agreement Random Forest (HARF) (SLUBAN; GAMBERGER; LAVRAČ, 2014). It uses *Random Forest* (RF) classifiers for noise identification. HARF considers the rate of disagreement in the predictions from the individual trees in the forest to detect the noisy examples: if this rate is relatively high, the example is considered noisy; otherwise, it is labeled as clean.

- *Saturation Filter* (SF) (GAMBERGER; LAVRAČ, 1997)(SLUBAN; GAMBERGER; LAVRAČ, 2014). It is is based on the observation that the elimination of noisy examples reduces the *Complexity of the Least Correct Hypothesis* (CLCH) value of the training set.

- *Cross-validated Committees Filter* (CVCF) (VERBAETEN; ASSCHE, 2003). Induces classifiers in a cross-validation strategy. The number of times an example is marked as noisy reflects its reliability. If the example is marked as noisy in most of the cross-validation rounds, CVCF classifies the example as noisy.

In this work, the ensemble-based noise filtering is employed in the experiments. It applies the prediction results through an ensemble of classifiers which has been used in many related works (SLUBAN; LAVRA, 2015a)(BRODLEY; FRIEDL, 1999)(SAEZ et al., 2015)(YUAN et al., 2018). In the following section, we will further discuss this approach.

2.2 ENSEMBLE-BASED FILTERING

Label noise can be detected from the prediction of a classifier or an ensemble of classifiers. The idea is to remove instances that are misclassified by the algorithm(s), i.e, instances which observed classes are different from the true classes. This approach has been widely chosen in early and recent works (BRODLEY; FRIEDL, 1999)(ZHU X., 2003)(VERBAETEN; ASSCHE, 2003) (SLUBAN; LAVRA, 2015b)(SLUBAN; GAMBERGER; LAVRAČ, 2014)(SAEZ et al., 2015)(YUAN et al., 2018)(GUAN et al., 2018) so to overcome the problem of using a single classifier.

Using only one classifier for the noise filter brings the risk of removing too many instances. A solution is to combine the predictions of a set of different algorithms (FRENAY; VERLEYSEN, 2014). This approach improves the noise detector, once that an instance is likely to have been incorrectly labeled if distinct classifiers disagree on their predictions for the instance.

The ensemble noise filtering applies the k-fold cross-validation, i.e, in k repetitions, k-1 folds of the dataset are used for training each algorithm in the ensemble, and the remaining fold is used for validation. Then, all instances are classified by all algorithms in the pool.

There are different techniques for combining the results of the predictions. The most common are *consensus* and *majority* vote (GUAN et al., 2018)(SLUBAN; LAVRA, 2015a)(YUAN et al., 2018). Whereas majority vote classifies an instance as incorrectly labeled if a majority of the algorithms in the pool misclassifies it, the consensus vote requires that all classifiers have misclassified the instance.

The two vote techniques produce different results. As consensus requires a higher agreement of classifiers, it tends to remove a few lines of the sample. On the other hand, the majority vote may throw out too many instances. Nevertheless, few works have also investigated the influence of ensemble threshold variations, i.e., the influence on noise detection when varying the number of erroneous ensemble predictions used to identify an instance as noisy. Khoshgoftaar, Zhong e Joshi (2005) and Sabzevari, Martinez-Munoz e Suarez (2018) showed, for example, that selecting adequate values of the ensemble threshold for noise filtering is superior to using standard filtering at all noise levels.

In this work, a variation of the ensemble threshold is applied so to analyze the impact of choosing one approach over another, besides checking if any specific threshold would maximize the results in a certain context.

2.3 NOISE MODELS

Broadly speaking, class label noise can be classified into two types: random and non-random noise. The random noise occurs independently of the input features and its probability is assumed to be class-conditional and equally shared among examples of the same class. On the other hand, a non-random noise is a noise which is influenced by the input features and its probability is not necessarily equal to examples of the same class (BOOTKRAJANG, 2016).

Label noise can be generated due to many reasons such as low reliability of human experts during labeling, incomplete information, communication problems, among others (SLUBAN; LAVRA, 2015b).

In Frenay e Verleysen (2014), the authors provided a taxonomy of label noise models, which reflects the distribution of noisy instances in a dataset. The three models are shown in Figure 2. Let's considered X the vector of features, Y the true class, $\hat{Y}$ the observed label and E a binary variable telling if a labeling error occurred. Each model has a different assumption on how noise is generated:

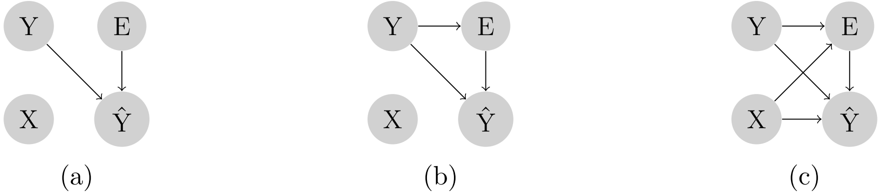

Figure 2 – Statistical taxonomy of label noise according to (FRENAY; VERLEYSEN, 2014). (a) NCAR, (b) NAR, and (c) NNAR. X denotes the vector of features, Y is the true class, $\hat{Y}$ is the observed label, and E is a binary variable telling whether a labeling error occurred. Arrows report statistical dependencies.

1. Noisy Completely at Random (NCAR): the occurrence of a mislabeled instance is independent on the instance's attributes and class. Mislabeled instances are uniformly

present across the instance space. In a binary classification problem, for example, there will exist the same proportion of mislabeled instances in both classes. In other words, as shown in Figure 2a, the occurrence of an error E is independent of the other random variables, including the true class itself (Y). For this model, the mislabeled instance probability is given by $p_e = P(E = 1) = P(Y \neq \hat{Y})$.

2. Noisy at Random (NAR): it is assumed that the probability of labeling errors depends on the instance class although it is not dependent on instance's attributes. Once mislabeling is conditional to instance classes, it allows us to model asymmetric label noise, i.e., when instances from certain classes are more prone to be mislabeled. This model could be applied, for example, to simulate mislabelling classification that is often verified in medical case–control studies where the misclassification of disease outcome may be unrelated to risk factor exposure (non-differential) (GILBERT et al., 2016). As shown in Figure 2b, E is still independent of X but it is conditioned by Y. For this model, the mislabeled instance probability is given by $p_e = P(E = 1) = \sum_{y \in Y} P(Y = y)P(E = 1|Y = y)$.

3. Noisy not at Random (NNAR): the probability of an error occurrence depends not only on the instance class but also on the instance attributes. In this case, for example, samples are more likely to be mislabeled when they are similar to instances of another class or when they are located in certain regions of the instance space. By applying this model, it is possible to simulate mislabeling near classification boundaries or in low density regions, it also can be applied for medical case-control studies where the misclassification of disease outcome may be related to risk factor exposure (differential) (GILBERT et al., 2016). As can be seen in Figure 2c, this is a more complex model, where E depends on both X and Y, i.e., labeling errors are more likely for certain classes and in certain regions of the X space.

It is usually quite difficult to identify the kind of noise present in a dataset without any background knowledge. Nevertheless, it is important to evaluate how sensitive noise detection techniques are to the noise distribution in a dataset. In this work, analysis regarding the NAR and NCAR models were performed in different scenarios.

2.4 PERFORMANCE MEASURES

Most experiments in the literature assess the efficiency of methods in detecting noise regarding accuracy (FRENAY; VERLEYSEN, 2014). A basic measure to evaluate the performance of noise detection is *precision*, which means how many instances the detector correctly identified as noisy among all instances identified as noisy by the detector:

$$\text{Precision} = \frac{\text{number of noisy cases correctly identified}}{\text{number of all noisy cases identified}}$$

In addition to the precision, another useful measure is *recall*, which calculates how many instances the detector correctly identified as noisy among all the noisy instances inserted into the dataset:

$$\text{Recall} = \frac{\text{number of noisy cases correctly identified}}{\text{number of all noisy cases in dataset}}$$

Finally, a measure that trades off *precision* versus *recall* is the *F-score*, which is the weighted harmonic mean of *precision* and *recall*:

$$\textit{F-score} = \frac{\beta \times Precision \times Recall}{Precision + Recall} \tag{2.1}$$

where $\beta^2 = \frac{1-\alpha}{\alpha}$, with $\alpha \in [0, 1]$ and $\beta^2 \in [0, \infty]$.

By setting the β parameter, it is possible to assign more importance to either precision or recall in the calculation of the *F-score*. In this work, the standard *F-score* (also referred to $F_1 score$, $\beta = 1$ and $\alpha = 1/2$) was used. It equally weights precision and recall.

2.5 CHAPTER REMARKS

In this chapter, an overall view of the main approaches of dealing with label noise problems was presented. In Section 2.1, the techniques were categorized according to their nature of implementation in two ways: algorithm-level approach, in which noise is dealt by adaptation of a classifier, and data-level approach, in which noise is dealt in a preprocessing step and is classifier-independent.

Well-known NF techniques were briefly described and the ensemble-based filtering was further discussed in Section 2.2 once this work focuses on an EF approach. The most common ensemble voting combinations, consensus and majority, were described along with their advantages and disadvantages. *Precision*, *Recal*, and *F-score* were defined as they will be applied to assess the EF performance.

Random and non-random noise types were defined in Section 2.3 and NCAR, a non-class-conditional model, and NAR, a class-conditional model, were differentiated, once both are random noise and will be employed in the experiments detailed in the proposed methodology presented in Chapter 3.

3 PROPOSED METHODOLOGY

This chapter provides details of the experimental setup adopted in our work to evaluate the ensemble noise detectors under NCAR and NAR models. In Section 3.1, the algorithms for the ensemble filter and the vote scheme approach are presented. In Section 3.2, the synthetic and real-world datasets are presented and the methodology for data generation with specific settings are described. The procedure for noise injection regarding the noise model is explained in Section 3.3. Lastly, the input variables and the experimental protocol are detailed in Section 3.4.

3.1 ENSEMBLE

The noise detection ensemble used in the experiments was generated from 10 algorithms as found in related work (SLUBAN; LAVRA, 2015a). They are from different families: decision trees, Bayesian models, neural networks, support vector machines, random forest, nearest neighbors and ruled-based methods. All the classifiers are implemented in R from specific packages as shown in Table 1.

Table 1 – Learning algorithms for classification noise filtering.

Algorithm	**R Package**
CN2 (rule learner)	RoughSets
kNN (nearest neighbor)	class
Naive Bayes	naivebayes
Random forest	randomForest
SVM (RBF Kernel)	e1071
J48	RWeka
JRip	RWeka
Multiplayer perceptron	RSNNS
Decision tree	party
SMO (linear Kernel)	RWeka

All algorithms' parameters used in the experiments were the default ones suggested in the R packages.

To combine the results from the ensemble, we employed different thresholds L of the ensemble, i.e, the proportion of algorithms used to make a decision regarding the classification of an instance. For this study, we made L varies from 10% to 100%.

In this way, when $L = 50\%$, the combination corresponds to the majority vote, i.e, if more than half (50% plus 1) of the votes from the ensemble predict a label different from the true class for an instance, it is considered mislabeled (noise). Likewise, when $L = 100\%$, the combination corresponds to the consensus vote, i.e, if all votes from the ensemble predict a label different from the true class for an instance, it is considered mislabeled.

Our goal is to analyze and check if there is a threshold that would maximize the noise detection under a specific context, i.e, taking into consideration the class imbalance ratio, noise distribution, and percentage of total noise.

3.2 DATA

Quantitative assessment of noise detection methods requires knowing which are the noisy instances beforehand. In real-world datasets, this is achieved either by expert labeling or by randomly injecting artificial noise into a dataset. While the former approach is not feasible for an extensive evaluation, the latter still has the problem of uncertainty about which instances are (originally) noisy when dealing with real-world datasets.

In order to handle this problem we divided our analysis into two parts: first, we performed several experiments on a set of synthetic datasets with a more controlled environment, and, then, we ran the experiments on a set of real-world datasets so to confirm our findings.

3.2.1 Synthetic data

To address the problem of quantitative assessment of noise detection in real-world datasets and still have reliable results, we first performed an analysis on ten (10) synthetic datasets. They are listed in Table 3 and illustrated in Figure 3[1].

In alignment with other researchers, we chose binary classification problems (the minority vs the majority class) with instances randomly distributed in the two-dimensional space. Following related works, we decided to use specific artificial datasets in order to have a more controlled environment.

The P2 dataset was generated according to Valentini (2005), and the remain datasets were generated with Mlbench[2]. All synthetic data employed in the experiments are binary (two-classes) problems.

We initially adopted the P2 dataset, where each class is defined in multiple decision regions determined by polynomial and trigonometric functions (VALENTINI, 2005). P2 is a convenient synthetic dataset for evaluation due to its complex and multimodal decision boundary. The P2 problem is illustrated in Figure 3a.

[1] Only bi-dimensional datasets are illustrated

[2] Mlbench package: rdocumentation.org/packages/mlbench

Table 3 – Synthetic data information.

Dataset	**Setting**
P2	P2(n)
2dnormals	2dnormals(n, cl=2)
circle	circle(n, d=2)
	circle(n, d=5)
ringnorm	ringnorm(n, d=2)
	ringnorm(n, d=5)
spirals	spirals(n)
	spirals(n, cycles=2)
	spirals(n, cycles=4)
cassini	cassini(n)

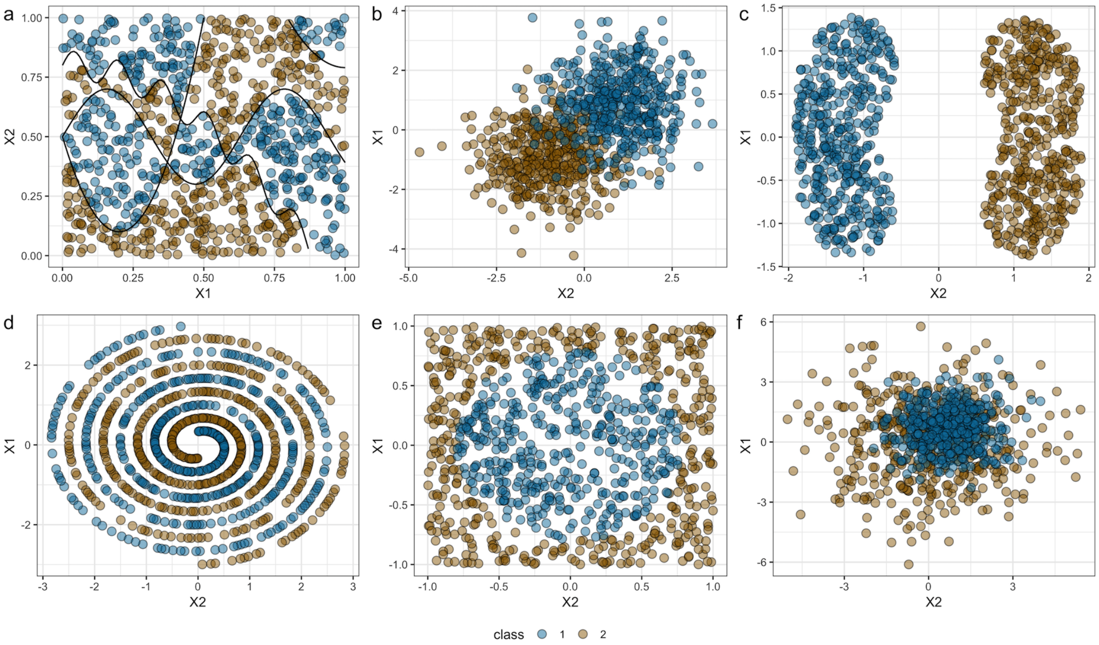

Figure 3 – Class distribution of synthetic data. a) P2 problem, b) 2dnormals(n,cl=2), c) cassini(n), d) spirals(n,cycles=4), e) circle(n, d=2), and f) ringnorm(n, d=2).

Afterward, we also included other binary synthetic datasets from the Mlbench Package (which is available in R libraries) using different input parameters (Table 3) to generate more data. In Figures 3b-f, it is possible to check the class distribution of datasets and their complexity. Some graphs for *spirals* datasets are omitted as the general behavior can be seen in Figure 3d. The *circle(n,d=5)* and *ringnorm(n,d=5)* are not bi-dimensional datasets, hence they are not presented in Figure 3.

An imbalance ratio generation step was included in the process in order to evaluate the impact of class imbalance. In this way, three different configurations of class distribution were created for each dataset. They are described as follows:

- **Configuration 1:** it was generated balanced data, i.e., data with equal instance distribution per class (50% for class 1 and 50% for class 2).
- **Configuration 2:** it was generated imbalanced data with uneven distribution of 30% of instances for class 1 and 70% for class 2.
- **Configuration 3:** it was generated imbalanced data with uneven distribution of 20% of instances for class 1 and 80% for class 2.

Although class 1 was the minority class in aforementioned cases, we highlight that the experiments were also performed by adopting class 2 as the minority class in turn. The results were practically the same compared to the ones obtained when class 1 was adopted as the minority class as it will be discussed later. In fact, both classes are comparable in terms of classification difficulty as can be observed in Figure 3.

3.2.2 Real-world data

The second part of the experiments was carried out on 20 (twenty) binary real-world dataset available at the KEEL-dataset repository (ALCALA-FDEZ A. FERNANDEZ, 2011), UCI repository (DUA; TANISKIDOU, 2017) and Open Media Library (VANSCHOREN et al., 2013). Some multi-class datasets are modified to obtain two-class imbalanced problems, defining the joint of one or more classes as positive and the remainder as negative. The list of datasets is presented in Table 4.

Unlike the process for synthetic data generation, when dealing with real-world data, we must be aware of some issues. First, it is likely that some datasets already contain noisy instances. Second, most of the real data present a certain original class *Imbalance Ratio* (IR).

In order to tackle the aforementioned issues, in our experiments, we applied a preprocessing step for data cleaning and also for imbalance ratio generation. While the former aim to remove possible existing noises, the latter aim to adequate real-world data imbalance ratio to simplify results comparison with synthetic data.

Table 4 – Real-world data information, where *Attr*, *Inst*, and *Rem* denote, respectively, the quantity of attributes, instances, and removed instances.

Dataset	**Attr.**	**Original**		**After cleaning**		
		IR	**Inst.**	**IR**	**Inst.**	**Rem.(%)**
arcene	10001	44:56	200	44:56	199	0.50
breast-c-w	10	34:66	699	35:65	673	3.72
column2C	7	32:68	310	32:68	308	0.65
credit	16	44:56	690	45:55	644	6.67
cylinder-bands	40	42:58	540	36:64	276	48.89
diabetes	9	35:65	768	31:69	720	6.25
eeg-eye-state	15	45:55	14980	45:55	14979	0.01
glass0	10	33:67	214	33:67	214	0.00
glass1	10	36:64	214	35:65	212	0.93
heart-c	14	46:54	303	45:55	289	4.62
heart-statlog	14	44:56	270	44:56	262	2.96
hill-valley	101	50:50	1212	48:52	1184	2.31
ionosphere	35	36:64	351	35:65	345	1.71
kr-vs-kp	37	48:52	3196	48:52	3194	0.06
mushroom	23	48:52	8124	38:62	5644	30.53
pima	9	35:65	768	32:68	732	4.69
sonar	61	47:53	208	46:54	206	0.96
steel-plates-fault	34	35:65	1941	35:65	1941	0.00
tic-tac-toe	10	35:65	958	35:65	958	0.00
voting	17	39:61	435	47:53	228	47.59

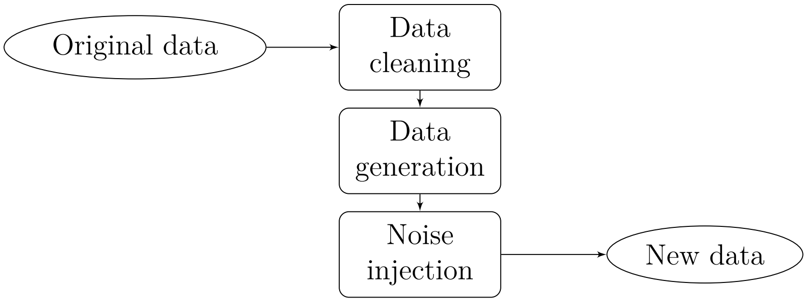

Figure 4 – Data generation process.

In this way, we chose datasets with low classes disparities as can be observed in Table 4. This made possible to generate new data from the original one with the same IR described in the previous section. The process of data generation for each dataset is summarized in Figure 4, and the details are described below:

- **Data cleaning:** A 10-fold classification was applied and the consensus vote was used to remove instances more likely to be noisy, i.e., instances misclassified by all classifiers were removed.
- **Data generation:** For generating data with new specific imbalance ratio, an undersampling process was applied as general illustrated in Figure 5. New data was generated for three different IR configurations: (1) 50:50, (2) 30:70, and (3) 20:80 as following described:
    - **Configuration 1:** it was generated balanced data, i.e., data with equal instance distribution per class (50% for class 1 and 50% for class 2). To do so, it was performed an undersampling process on the majority class so that instances were removed until reaching a balanced ratio. Taking *eeg-eye-state* dataset as an example, which has 8239 instances in the majority class and 6741 in the minority class, it would have 1498 random instances removed from majority label. In this case, the final dataset would have a total of 13482 rows with 6741 each class.
    - **Configuration 2:** it was generated imbalanced data with uneven distribution of 30% of instances for class 1 and 70% for class 2. To do so, it was applied a random undersampling process on the minority class. For the *eeg-eye-state* dataset example, the final minority class would have 3531 instances, corresponding to 30% of data, while the majority class would have 6741 instances, corresponding to 70% of data.
    - **Configuration 3:** it was generated imbalanced data with uneven distribution of 20% of instances for class 1 and 80% for class 2. Again, to do so, it was applied a random undersampling process on the minority class. For the *eeg-eye-state* dataset example, the final minority class would have 2060 instances, corresponding to 20% of data, while the majority class would have 6741 instances, corresponding to 70% of data.

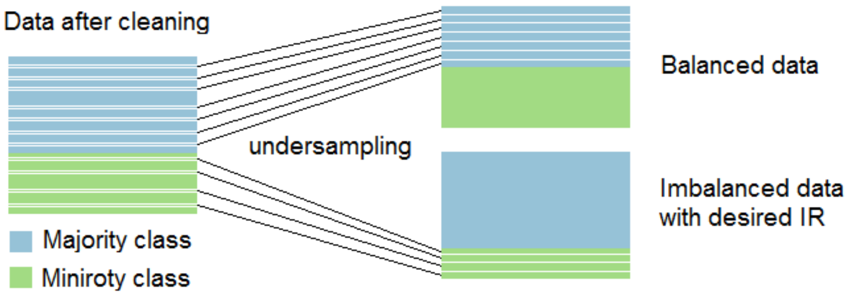

Figure 5 – Imbalance ratio generation for real data.

- **Noise injection:** The process for injecting noise was the same applied on synthetic datasets and it is described in Section 3.3.

3.3 NOISE INJECTION

For noise detection evaluation, random label noise was injected in the testing set by adopting the NAR, and the NCAR models with distinct *Noise levels in dataset* (p). In this works, p assumed 4 (four) different values: 5%, 10%, 15%, and 20%, which correspond to the proportion of noisy instances in the testing set. This was done by changing the classes labels in a certain number of instances randomly selected.

For NAR model, noise was inserted to achieve a certain *Noise Ratio* (M) of noisy instances per class:

- $M = 9/1$: for every nine noisy instances in minority class, there is one noisy instance in majority class. It means that it is more difficult to label instances in the minority class. This chosen ratio corresponds to NAR (9:1) on results discussion.

- $M = 1/9$: in turn, for each noisy instance in minority class, there are nine noisy instances in majority class. It means that the majority class is more prone to label noise. This chosen ratio corresponds to NAR (1:9) on results discussion.

This ratio was chosen in such a way as to have a great discrepancy between the noises in each class so as to be better analyzed. The exact number of noisy instances of each class is derived according to both the desired noise level p and the ratio M. Let d_n be the number of instances in the testing set. Let p be the desired noise level in the data test. Let n_1 and n_2 be the number of noisy instances in each class. Then $n_1 + n_2 = p \times d_n$ and $M = n_1/n_2$. In order to obtain such constraints, n_1 and n_2 are defined according to the following equations:

$$n_1 = \frac{M \times (d_n \times p)}{M + 1} \tag{3.1}$$

$$n_2 = \frac{d_n \times p}{M + 1} \tag{3.2}$$

For instance, suppose that we have $d_n = 1000$ instances in the testing set, that the desired noise level is $p = 0.10$ (100 noisy instances), and that we are going to inject noise with the three different models explored in this work: (1) NCAR ($M = 1$), (2) NAR 9:1 ($M = 9$), and (3) NAR 1:9 ($M = 1/9$).

By adopting the above equations for the first example, NCAR ($M = 1$), we would find $n_1 = 50$ and $n_2 = 50$. In this case, the number of noisy instances injected in the testing set are equal for both classes. This example is illustrated in Figure 6. Notice that NCAR is a special case of NAR.

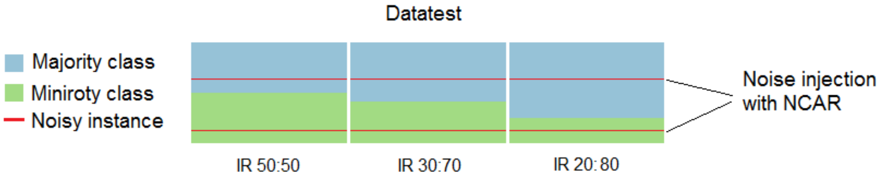

Figure 6 – Noise injection with NCAR model in data with different imbalance ratios.

Likewise, when applying previous equations for the second example, NAR 9:1 ($M = 9$), we would find $n_1 = 90$ (90 noisy instances in minority class) and $n_2 = 10$ (10 noisy instances in majority class). In this case, the number of noisy instances injected in the testing set are greater for the minority class. This example is illustrated in Figure 7.

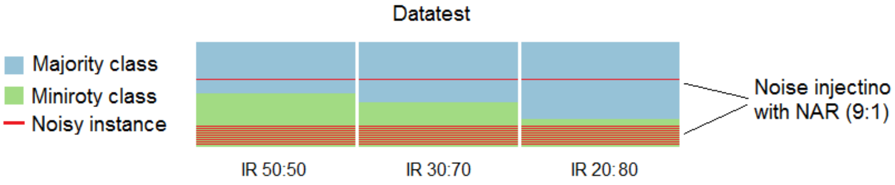

Figure 7 – Noise injection with NAR 9:1 model - grater noise ratio in minority class - in data with different imbalance ratios.

Lastly, for the third example, NAR 1:9 ($M = 1/9$), we would find $n_1 = 10$ (10 noisy instances in the minority class) and $n_2 = 90$ (90 noisy instances in the majority class) as results of the previous equations. In this case, the number of noisy instances injected in the testing set are greater for the majority class as illustrated in Figure 8.

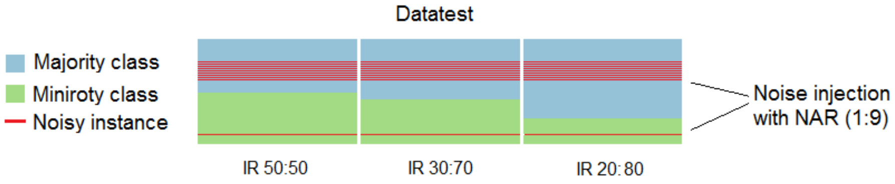

Figure 8 – Noise injection with NAR 1:9 model - grater noise ratio in majority class - in data with different imbalance ratios.

3.4 EXPERIMENTAL PROTOCOL

In this section, it is described the steps followed so to have the outputs necessary to perform the analysis discussed in Sections 4.1 and 4.2.

The combination of the main input values used in the experiments is presented in Table 6. The main input parameters are: class imbalance ratio (IR), total percentage of noisy instances in the data test (p), and noise ratio (M).

Table 6 – Experiment setup

Imbalance Ratio (IR)	Total percentage of				Noise Ratio (M)
Class 1 : Class 2	noise in testing set (p)				Class 1 : Class 2
50 : 50	5%	10%	15%	20%	NCAR NAR (1 : 9) NAR (9 : 1)
30 : 70	5%	10%	15%	20%	NCAR NAR (1 : 9) NAR (9 : 1)
20 : 80	5%	10%	15%	20%	NCAR NAR (1 : 9) NAR (9 : 1)

The general process followed according to steps described in previous sections is presented in Figure 9.

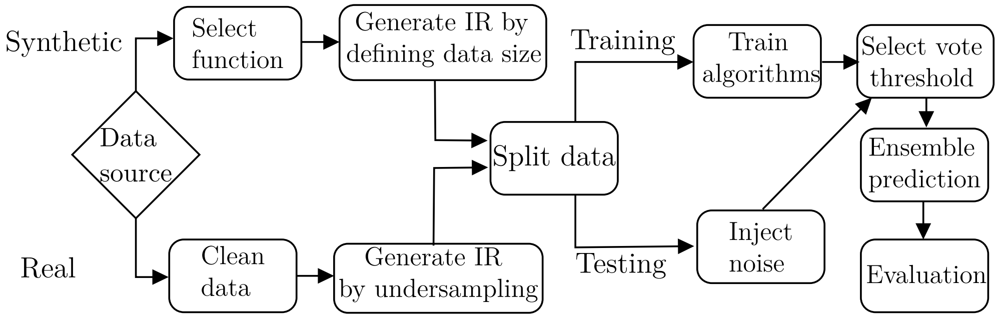

Figure 9 – General experimental protocol.

The general flow were executed for each input combination shown in Table 6. As can be observed, the only difference between synthetic and real data analysis is the data generation approach.

For a more detail analysis and protocol replication, the process is also outlined in Algorithm 1 and Algorithm 2 for synthetic and real data respectively. Each one uses specific parameters as input to generate a certain scenario for analysis. The algorithms use the following taxonomy:

- *IR*: corresponds to one of the three class imbalance ratio shown in Table 6.
- *p*: corresponds to one of the four percentage of noise (to be injected in testing set) shown in Table 6.
- *M*: corresponds to one of the three noise ratio (noise distribution between classes) shown in Table 6.
- *dataset*: corresponds to each dataset listed in Sections 3.2.1 and 3.2.2 for synthetic and real data respectively.
- *number_rows*: defines the total of instances in synthetic datasets. For this work, *number_rows* had a fixed value of 2000.
- *threshold*(L): assumes the values described in Section 3.1.
- *classifiers*: are the classifiers presented in Section 3.1.

The procedure outlined in Algorithm 1 is detailed as follows:

Line 1 A dataset is generated according to one of the datasets in Table 3, the desired number of rows (instances), and the class imbalance ratio.

Algorithm 1 Experimental procedure for synthetic data

```
Input: IR, M, p, dataset, classifiers, threshold (L)
Output: performance measures
1: $dat \leftarrow generateData(dataset, number\_rows, IR)$
2: $training, testing \leftarrow split(dat, 70\%, 30\%)$ {Split data in training and testing}
3: $testing\_n \leftarrow injectNoise(testing, M, p)$
4: for c in classifiers do
5:     $model \leftarrow train(training, c)$
6:     $predictions \leftarrow classify(model, testing\_n)$
7: end for
8: $ensemble\_prediction \leftarrow voting(predictions, L)$
9: $measures \leftarrow calculate(ensemble\_prediction)$
```

Line 2 The generated dataset is split into training (70%) and testing (30%) data.

Line 3 The percentage of noise *p* is injected into the data test according to the noise ratio *M*.

Line 5 Each classifier is trained with the training data.

Line 6 Then, each classifier is used to classify the testing data.

Line 8 With all predictions, an ensemble vote is applied using a proportion *L* of algorithms.

Line 9 *F-score*, *precision*, and *recall* are then calculated.

Algorithm 2 Experimental procedure for real data

```
Input: IR, M, p dataset, classifiers, threshold (L)
Output: performance measures
 1: $clean \leftarrow dataCleasing(dataset, classifiers)$
 2: $dat \leftarrow generateData(clean, IR)$
 3: $training, testing \leftarrow split(dat, 70\%, 30\%)$ {Split data in training and testing}
 4: $testing\_n \leftarrow injectNoise(testing, M, p)$
 5: for c in classifiers do
 6:     $model \leftarrow train(training, c)$
 7:     $predictions \leftarrow classify(model, testing\_n)$
 8: end for
 9: $ensemble\_prediction \leftarrow voting(predictions, L)$
10: $measures \leftarrow calculate(ensemble\_prediction)$
```

The procedure outlined in Algorithm 2 is detailed as follows:

Line 1 A data cleaning process using all classifiers (consensus vote) is applied to remove possible noise from data (Table 4).

Line 2 A new dataset is generated from cleaned data with the desired class imbalance ratio (as described in Section 3.2.2).

Line 3 The generated dataset is split into training (70%) and testing (30%) data.

Line 4 The percentage of noise p is injected into the data test according to the noise ratio M.

Line 6 Each classifier is trained with the training data.

Line 7 Then, each classifier is used to classify the testing data.

Line 9 With all predictions, an ensemble vote is applied using a proportion L of algorithms.

Line 10 *F-score*, *precision*, and *recall* are then calculated.

Due to the stochastic nature of noise injection, this insertion is normally repeated a number of times for each noise level (SLUBAN; GAMBERGER; LAVRAČ, 2014) (ZHU X., 2003). In this work, for each combination of parameters, the procedures in Algorithm 1 and 2 was repeated 100 times and the ensemble noise detector was evaluated in terms of the average results of *Precision*, *Recall*, and *F1-Measure*.

4 RESULTS

In this chapter, the findings from the experiments described in Section 3.4 are presented and examined. The discussion is divided into two main parts: the results on synthetic data in Section 4.1, and the results on real data in Section 4.2.

The analysis is performed considering the metrics presented in Section 2.4 for assessing the ensemble filter (Section 2.2) performance under following noise models: NCAR and NAR (Section 2.3). The behavior examination of noise detection is conducted applying a different percentage of noise (Section 3.3) in several datasets with a variation of the class imbalance ratio (Section 3.2). Lastly, an analysis of the impact on noise detection when changing the ensemble vote threshold (Section 3.1) is performed.

The experiments were performed on all datasets for all combination of parameters as described in Section 3.4. The results for each one can be found in Appendix A for synthetic data and in Appendix B for real data.

4.1 RESULTS ON SYNTHETIC DATA

The results shown in this section were obtained from the steps outlined in Algorithm 1. To facilitate the analysis, the discussion is carried out putting into perspective each input parameter that resulted in a specific scenario. In this way, we first examine the noise detection under balanced and imbalanced datasets in Section 4.1.1. Then, we explore the impact of different noise level per class in Section 4.1.2. Lastly, in Section 4.1.3, we analyze noise detection under different noise ensemble thresholds.

4.1.1 Balanced vs Imbalanced Datasets

As discussed earlier, we generated datasets and repeatedly injected random noise into them. We analyzed the experiments by applying three different noise distributions in order to investigate how NCAR and NAR have an influence on the results. The outcome is shown in Figures 10 and 11. They present the *F-score* performance vs percentage of noise p in data testing. In the columns, it is possible to observe the imbalance ratios, while, in the rows, each dataset can be checked.

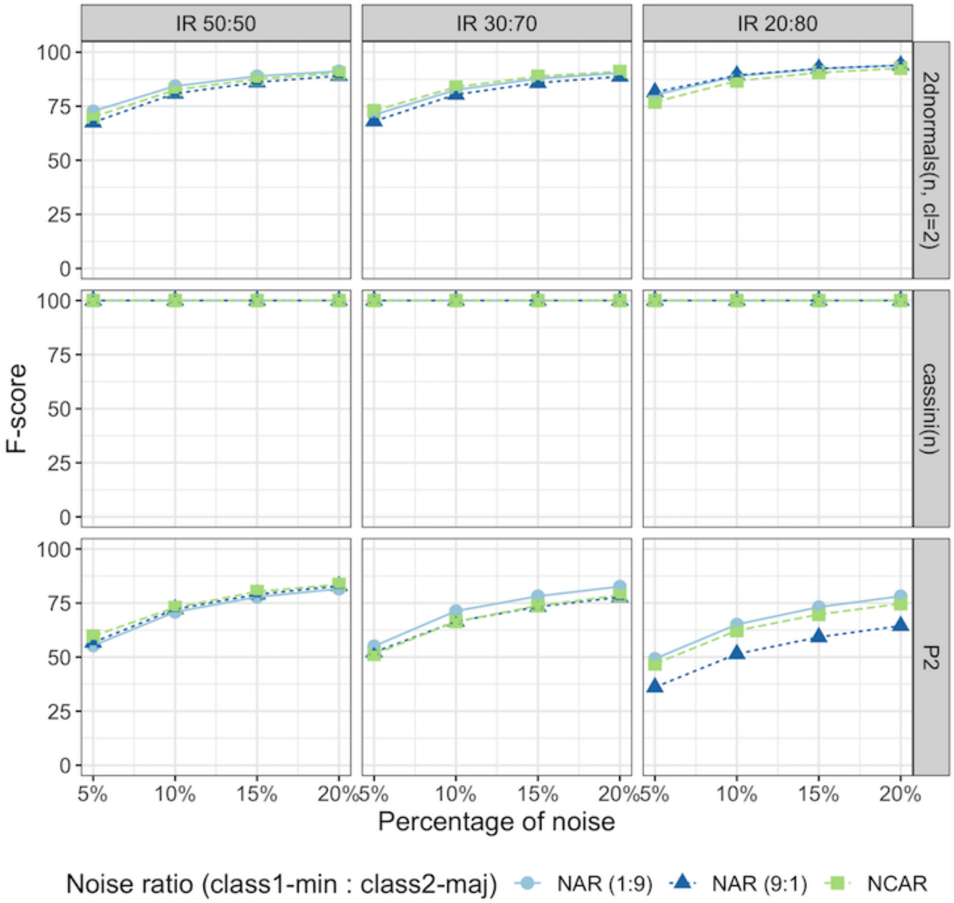

Figure 10 – *F-score* performance for majority vote on *2dnormals(n,cl=2)*, *P2*, and *cassini(n)* datasets.

Observing the first column (IR 50:50) for each dataset, i.e, when dealing with balanced datasets, we can see, from the overlapping lines, that NCAR and NAR have practically the same impact on noise detection. There is no significant difference in noise detection performance. From our experiments with synthetic data, only the *spirals(n,cycles=2)* dataset (Figure 30) had partially a different result.

In light of this, it is possible to infer that the amount of noise in a class with regards to the other class is not relevant in the scenario of balanced classes. In fact, if we take into consideration that algorithms are equally trained regarding the two classes, there will be no tendency in mislabeling one over another.

Nevertheless, it is known that most real-world datasets are more likely to have classes unevenly distributed. Imbalanced datasets are usually challenging since instances in the minority group are, in general, prone to be misclassified. The performance of ensemble noise detectors may be affected as well. This leads us to another scenario where we investigated the behavior of NCAR and NAR in imbalanced datasets. The results can be verified in the second and third columns of Figures 10 and 11. In the second column (IR of 30:70), we have datasets with 30% of instances in the minority class, whereas, in the third column (IR of 20:80), the datasets with 20% of instances in minority class.

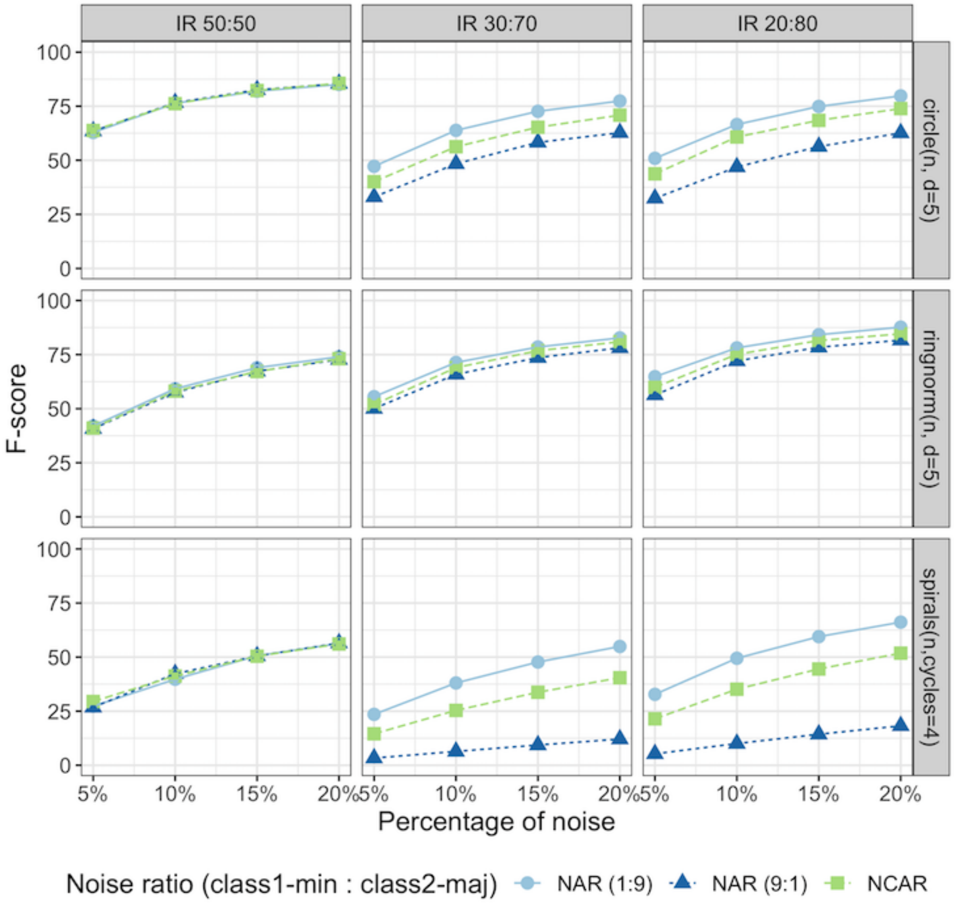

Figure 11 – *F-score* performance for majority vote on *circle(n,d=5)*, *ringnorm(n,d=5)*, and *spirals(n,cycles=4)* datasets.

Unlike the observed results for balanced class distributions, ensemble detection can have different behaviors considering both NCAR and NAR models when dealing with imbalanced datasets. This can be easily checked by visually comparing the *F-score* lines for an IR of 50:50 in contrast with the other IRs.

For most datasets, class imbalance harmed the performance of noise detection under the NCAR model. To better access this behavior, the *F-score* variation was calculate and summarized in Table 8. This variation was obtained by calculating the difference between the *F-score* in imbalanced data (IR of 20:80) and the *F-score* in balanced data (IR 50:50). In other words, in Table 8, it is shown the variation produced on *F-score* measure when the IR varies from a balanced dataset to an imbalanced data of 20:80 ratio. The negative numbers denote a decrease in noise detection, whereas, positive numbers denote an improvement. For example, in Figure 11, for 15% of injected noise, the average *F-score* under NCAR for *circle(n,d=5)* dataset is 82.6 in balanced datasets, then, it decreases to 67.5 for a class imbalance ratio of 20:80. This corresponds to a total variation of -15.1 as shown in Table 8.

Table 8 – F-score variation vs class imbalance ratio in synthetic data when class 1 is the minority class.

Datasets	**F-score variation when IR goes from 50:50 to 20:80**											
	5% of noise			**10% of noise**			**15% of noise**			**20% of noise**		
	NAR (9:1)	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
2dnormals(n,cl=2)	12.8	9.2	5.7	8.4	5.1	3.6	6.1	4.2	2.7	4.4	3.2	2.0
cassini(n)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
circle(n,d=2)	-13.2	-12.2	-11.8	-8.8	-7.5	-6.9	-8.2	-6.0	-4.9	-5.8	-4.9	-4.0
circle(n,d=5)	-31.1	-20.8	-14.6	-30.6	-18.0	-11.5	-28.5	-15.1	-8.6	-25.5	-12.9	-6.7
P2	-9.8	-4.2	10.9	-11.3	-3.2	9.4	-10.4	-4.2	7.3	-9.2	-2.2	6.5
ringnorm(n,d=2)	-23.7	-1.6	7.4	-34.8	-1.5	8.1	-38.2	-2.4	7.7	-40.0	-2.6	6.8
ringnorm(n,d=5)	-6.3	2.5	2.3	-7.0	1.7	3.5	-7.1	1.4	2.4	-6.2	1.0	2.7
spirals(n)	-18.1	-9.0	-5.1	-14.5	-7.5	-3.1	-13.4	-7.6	-2.5	-14.2	-8.3	-2.1
spirals(n,cycles=2)	-75.9	-58.3	-39.4	-75.3	-51.0	-32.2	-72.3	-42.7	-26.6	-70.4	-37.1	-20.4
spirals(n,cycles=4)	-17.7	-2.7	10.1	-25.8	-3.9	11.8	-30.7	-2.6	13.0	-32.7	-2.6	13.6

In this same context (column 15% of noise and NCAR), it was also verified a decrease in noise detection, when IR is increased, for *circle(n,d=2)*, *P2*, *circle(n,d=2)*, *ringnorm(n,d=2)*, *ringnorm(n,d=5)*, *spirals(n)*, *spirals(n,cycles=2)* and *spirals(n,cycles=2)* problems. Observing the other percentages (5%, 10% and 20% of noise columns), the same overall negative variation was verified. This behavior makes more evident the damage caused in detection when applying NCAR in a imbalanced dataset.

The ensemble performance in identifying mislabeled instances decreases due to the majority class influence. In fact, classifiers are biased to predict more instances as being from the majority class. In such a case, many instances from the minority class were incorrectly labeled as the majority were judged as being correctly classified by the ensemble. Hence, the ensemble failed in detecting many minority instances as noisy, which strongly harmed the *Recall* measure (results for *Recall* measure can be assessed in Appendix A).

As shown, this behavior is also verified for the other percentages of noise in data although with less or more impact. For instance, taking the *spirals(n,cycles=2)* dataset under NCAR model at each level of noise, we can see that the negative variation of *F-score* decreases from -57.7 (at 5% of noise) to -36.8 (at 20% of noise). In other words, when the percentage of noise in data test is higher in a more imbalanced class scenario, noise detection is improved. This is reasonable once the classifiers are considered of high quality as they are trained from noiseless datasets in these experiments. In this way, more noise injected in the majority class (which implies noise not only in the boundaries areas) are easily detected.

Furthermore, we also changed the minority and majority classes, i.e, *class 1* corresponding to the majority class (while *class 2* corresponding to the minority class) in order to to evaluate if the negative influence also would occur in such a scenario. As shown in Table 9, the same behavior is verified in most datasets.

Nevertheless, we found two exceptions: *2dnormals* (also in Table 8) and *ringnorms* datasets which presented positive variations. For example, when *class 1* was the minority class, the *F-score* of *ringnorm(n,d=2)*, at 15% of noise and under NCAR, had a negative variation of −2.4 (Table 8). When *class 1* became the majority class, the *F-score*, under the same settings, had a positive variation of 27.8 (Table 9).

This led us to investigate the way data are spread in the space for those datasets when the IR gets higher. Class distribution for both datasets is shown in Figures 12 for *2dnormals* and in 13 for *ringnorms*.

Table 9 – F-score variation vs class imbalance ratio in synthetic data when class 2 is the minority class.

Datasets	**F-score variation when IR goes from 50:50 to 20:80**											
	5% of noise			**10% of noise**			**15% of noise**			**20% of noise**		
	NAR (9:1)	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
2dnormals(n,cl=2)	6.1	11.4	1.9	4.2	6.2	1.2	3.3	5.2	1.1	2.3	3.9	0.9
cassini(n)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
circle(n,d=2)	-15.0	-15.7	-10.0	-10.7	-9.1	-6.0	-8.8	-7.6	-4.0	-6.4	-6.8	-3.1
circle(n,d=5)	-43.5	-28.1	-18.9	-43.9	-26.6	-15.7	-40.8	-23.0	-12.3	-37.6	-20.4	-9.8
P2	-9.8	-4.2	10.9	-11.3	-3.2	9.4	-10.4	-4.2	7.3	-9.2	-2.2	6.5
ringnorm(n,d=2)	24.0	38.0	33.3	21.4	33.7	27.9	17.8	27.8	23.9	11.9	23.9	20.2
ringnorm(n,d=5)	5.2	13.3	15.8	4.5	13.1	13.5	3.0	10.7	11.7	2.0	8.4	10.1
spirals(n)	-24.3	-15.2	-5.9	-22.5	-13.4	-4.0	-18.1	-13.4	-3.4	-22.7	-13.6	-2.9
spirals(n,cycles=2)	-73.8	-57.7	-39.1	-70.6	-50.6	-32.0	-65.7	-42.4	-26.4	-63.1	-36.8	-20.3
spirals(n,cycles=4)	-17.6	-3.0	9.8	-25.7	-4.2	10.5	-30.6	-3.0	12.1	-32.5	-2.9	12.6

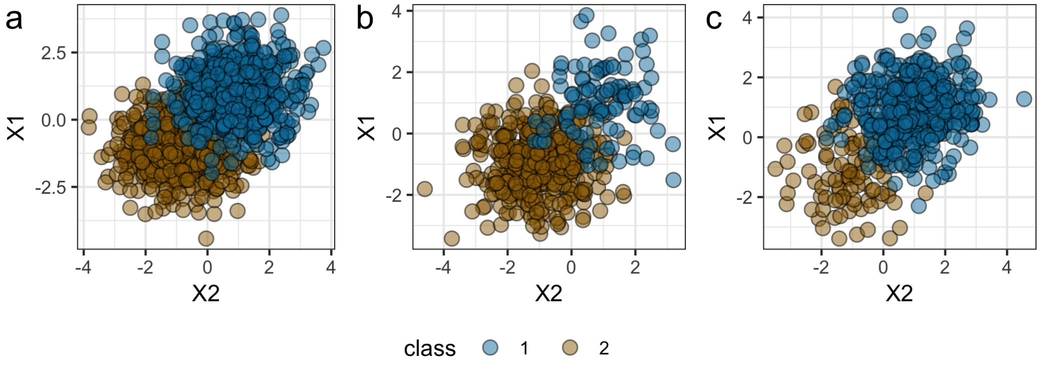

Figure 12 – Data distribution of the *2dnormals* dataset: a) IR = 50:50, b) IR = 20:80 and class 1 = minority class, and c) IR = 20:80 and class 2 = minority class.

From Figures 12 and 13, we can rationalize why noise detection under the NCAR had increased instead of decreasing in such cases. First, data are more distinguished in *2dnormals* with a higher IR, i.e, data from each class are more separated from each other. This, in turn, makes the classification process easier and, consequently, the ensemble noise detection.

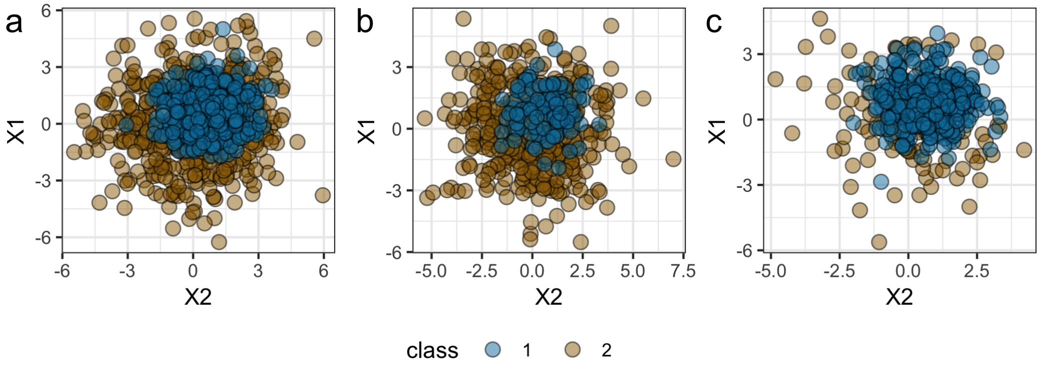

Figure 13 – Data distribution of the *ringnorm* dataset: a) IR = 50:50, b) IR = 20:80 and class 1 = minority class, and c) IR = 20:80 and class 2 = minority class.

Second, for *ringnorms* dataset, there were two different behaviors for data distribution depending on which class was the minority one: 1) *class 1* as the minority presents an even smaller inner circle without a separability boundary from the surround data; 1) *class 2* as the minority, on the contrary, has a bigger inner circle with a more defined boundary line between classes. This explicates why the *ringnorm* datasets had different results. Negative in the first scenario and positive in the second. This alerts us for the intrinsic influence of datasets particularities on the outcome, which is an inherent issue in ML.

4.1.2 Noise Level in the Minority vs Majority Class

Class imbalance also affected the noise detection performance under the NAR model, but in different ways depending on which class is noisier. In the great majority of datasets, a negative impact was verified when the minority class was noisier than the majority class. On the other hand, noise detection performance had a minor impact or the result was even improved in most datasets when the majority class was the noisiest one.

When there is a higher noise ratio in the minority class, NAR(9:1), the harm on the ensemble filter performance is worse compared to NCAR as shown in Figures 10 and 11 for IRs of 30:70 and 20:80 as NAR(9:1) lines are usually under NCAR lines.

The impact on performance is higher with a greater class imbalance ratio. For example, under NAR(9:1) model, for P2 dataset at 15% of noise level, *F-score* decreased from 72.1 when IR is 30:70 to 65.0 when IR is 20:80. This is also verified for other datasets, as shown in Tables 8 and 9, *F-score* under NAR(9:1) suffers a higher variation in contrast to NCAR.

In turn, when there is a higher noise ratio in the majority class, NAR(1:9), the noise detection performance is improved or less impacted in comparison to NCAR and NAR(9:1). This is observed from the positive numbers or smaller variation on NAR(1:9) column in comparison to the other models in Tables 8 and 9 and also from the upper lines in Figures 10 and 11 when IR gets higher.

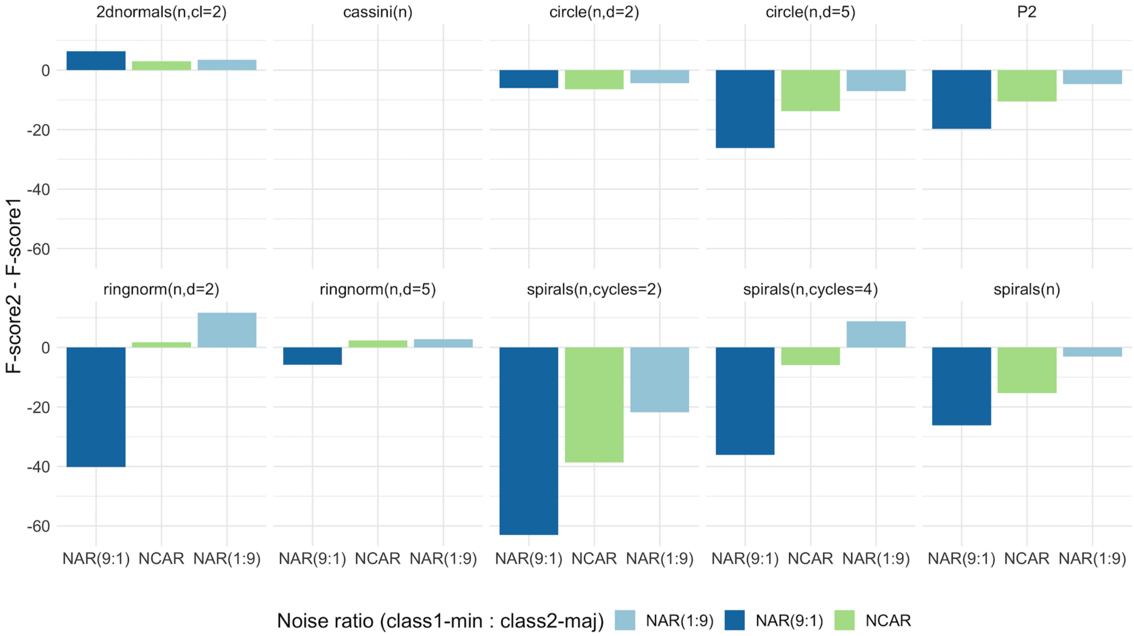

Figure 14 – Variation in *F-score* performance when IR increases from 50:50 (F-score1) to 20:80 (F-score2) in presence of a noise level of 15%.

The prevalence in the number of instances in the majority class makes the detection process easier in the presence of noisier instances in this class. The noise detection still failed in the minority class but it does not impact performance since there are fewer noisy instances in this class for $M = 1/9$. Based on these findings, we can suppose that previous work that assumed the NCAR model may be over or underestimating the performance of noise detectors, which will be affected by the class imbalance ratio and noise level at each class.

Figure 14 shows a visual representation of Table 8 which presents the general behavior of noise detection performance under the three noise models configuration when class imbalance ratio goes from a balanced dataset to a dataset with IR of 20:80. In Figure 14, bars situated below zero denote a decrease in detection performance while the opposite implies an improvement.

As can be seen in Figure 14, there is a pattern in the way noise detection behaves according to each noise model. Some exceptions were found as explained in the previous section. In all experiments, for the *cassini(n)* dataset, we did not verify any pattern discussed so far which can be attributed to the low level of difficulty in classification (see Figure 3).

4.1.3 Noise Detection vs Ensemble Vote Thresholds

In this work, we also performed experiments taking into consideration the proportion L of algorithms in the ensemble to have a final decision on noise detection. The main idea is to verify in which circumstances a different threshold would respond better in terms of noise detection performance in contrast with the most commonly used approaches in the literature.

For this, we varied L from 10% to 100% and grouped the results by the highest *F-scores*. The results are shown in Table 10.

Table 10 – Best ensemble threshold. #min denotes a greater noise ratio in minority class (NAR 9:1), #maj denotes a greater noise ratio in majority class (NAR 1:9), and #c denotes the noise ratio equally distributed (NCAR).

Datasets	**Noise %**	**Best ensemble threshold**								
		IR of 50:50			**IR of 30:70**			**IR of 20:80**		
		#min	**#c**	**#maj**	**#min**	**#c**	**#maj**	**#min**	**#c**	**#maj**
2dnormals (n,cl=2)	5	10	10	10	10	10	10	9	10	10
	10	10	10	10	9	10	10	9	10	10
	15	9	10	9	9	10	9	9	9	10
	20	9	9	9	8	9	9	4	8	9
circle (n,d=5)	5	8	8	7	6	9	9	5	10	10
	10	7	7	7	5	8	9	3	6	10
	15	7	7	6	4	7	9	3	5	9
	20	7	6	6	3	6	8	3	4	9
P2	5	8	8	8	7	7	10	5	7	10
	10	8	7	8	6	7	10	4	7	9
	15	7	7	7	5	7	9	3	5	9
	20	7	7	7	4	6	9	2	4	9
ringnorm (n,d=5)	5	9	9	10	8	9	10	7	9	10
	10	8	9	9	7	8	10	4	8	10
	15	8	8	8	6	8	9	4	7	9
	20	7	8	8	5	7	9	3	6	9
spirals (n,cycles=4)	5	8	8	8	1	10	10	1	10	10
	10	8	7	8	1	10	10	1	10	10
	15	7	7	8	1	10	10	1	1	10
	20	7	7	8	1	1	10	1	1	10

*Where values 1, 2, ... , 10 correspond to 10%, 20%, ..., 100%.

For better visualization, the values shown in Table 10 goes from 1 (one) to 10 (ten), where, 1 corresponds to $L = 10\%$, 10 corresponds to $L = 100\%$ and so forth. In this setting, if the best threshold found is $L = 2$, for example, it means that the highest noise detection performance was achieved when 20% of the algorithms were used in the pool. In this way, majority and consensus vote are the best choices when $L = 5$ and $L = 10$ respectively.

The values in Table 10 show many thresholds that would deliver better results than consensus and majority vote under a specific context. In order to analyze the gain in choosing a different threshold, we plotted every *F-score* result of each L for all datasets (see Appendix A).

In Figure 15, for example, the results for *circle(n,d=5)* dataset are shown. In this case, for an IR of 20:80 with 15% of noise under the NAR (9:1) model, the *F-score* would be 63.0 and 14.80 if applied a majority and consensus vote respectively. On the other hand, if a threshold of $L = 3$ is applied, which corresponds to the best L (Table 10), the resulting *F-score* would be equal to 70.85.

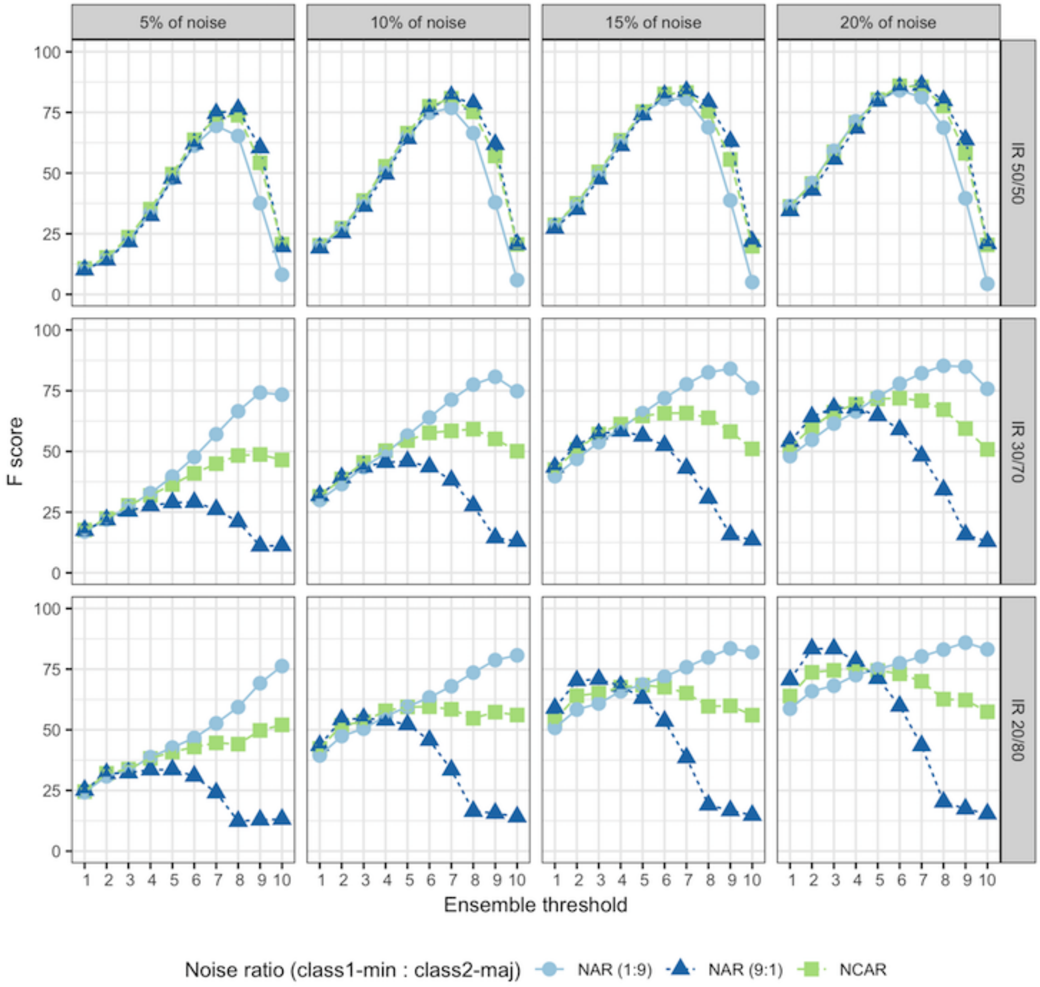

Figure 15 – Noise detection *F-score* on *circle(n,d=5)* dataset under different ensemble vote thresholds (where 1 = 10%, 2 = 20%,..,10 = 100%).

Although these findings are part of a preliminary investigation, interestingly aspects were found from analysis on the plotted graphs for all data (Appendix A). These aspects were similar for most datasets, therefore, the following discussion will be conducted from Figure 15. For better analysis, we focused on one variable at a time while examining the behavior of noise detection.

4.1.3.1 Varying Class Imbalance Ratio (IR)

In general, we observed that, for NCAR and NAR(9:1) models, when the IR gets higher (from a 50:50 to 20:80), the threshold to produce the best noise detection tends to have a smaller number. What may indicate that, for this context, only specialized algorithms are able to detect properly whether an instance is noisy or not. Conversely, for the case when the majority class is the noisiest one, NAR (1:9) model, we verified that, as IR gets higher, a higher threshold produces better noise detection. In this case, as the noise may be more spread in the space, a diversity pool might improve the ensemble filter performance.

This aforementioned pattern can be verified in Figure 15 when we look at one column at a time (which indicates the percentage of noise). Looking at NAR (9:1) curve, the peak points of each curve (which indicates the best threshold) are located on the left side of the graphs (which indicates a smaller threshold). Conversely, the peak points of NAR (1:9) curve are located on the right side of the graphs which indicates a greater threshold. Interestingly, in the case noise are equally distributed between classes (NCAR model), a center-tendency is observed. In this case, choosing an L around the majority vote is likely to result in satisfactory noise detection performance.

4.1.3.2 Varying the Total Percentage of Noise

Although the amount of noise in data impacts the performance in detection, it does not seem to have significant influence (compared to IR variable) on the best threshold choice. If we look at one row at a time (which indicates the IR) in Figure 15, we can verify that the pattern in the three curves is practically the same for different levels of noise when data are balanced.

Nevertheless, it is observed that, if the total amount of noise in the dataset is increased, the best threshold has a slight tendency to have a smaller number for higher values of IR. This can also be visualized in Table 10 for each noise model. Although the threshold has the same behavior of getting lower as the percentage of noise gets higher, we can see a greater impact when the minority class is the noisiest one. In Figure 15, for an IR of 30:70, the best threshold for NCAR goes down from 9 to 6, and from 9 to 8 or NAR(1:9) model when the noise percentage varies from 5% to 20%. For the NAR (9:1) model, in turn, this decrease is from 6 to 3. This behavior is also found for an IR of 20:80, but in a more pronounced way.

4.1.4 Statistical tests

The Friedman test (FRIEDMAN; RAFSKY, 1979) was used in order to compare the impact of all three noise models over the 108 problems (9[1] datasets X 3 IR's X 4 different percentages of noise). Since we are comparing the effect of three different noise models on ensemble detection, the degree of freedom is 2. The level of significance was set to $\alpha = 0.05$, i.e., 95% confidence. The Friedman test shows that there is a significant difference on the detection noise under the three models. The *p-value* obtained for each problem is presented in Table 12

[1] Cassini dataset was removed due to its 100% precision.

Table 12 – Friedman test results of each problem. Non-significantly difference ($\alpha > 0.05$) are marked with *.

Datasets	IR	Friendman test result p-value			
		5%	10%	15%	20%
2dnormals(n,cl=2)	50:50	0.0000	0.0000	0.0000	0.0000
	30:70	0.0009	0.0001	0.0004	0.0008
	20:80	0.0000	0.0000	0.0000	0.0000
circle(n,d=2)	50:50	0.0164	0.0118	0.0039	0.0167
	30:70	0.0003	0.0032	0.0013	0.0000
	20:80	0.0821*	0.0110	0.0000	0.0305
circle(n,d=5)	50:50	0.0446	0.0055	0.0177	0.0050
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
P2	50:50	0.0004	0.0000	0.0000	0.0001
	30:70	0.0450	0.0202	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
ringnorm(n,d=2)	50:50	0.0006	0.0009	0.0010	0.0003
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
ringnorm(n,d=5)	50:50	0.1225*	0.0015	0.0068	0.1588*
	30:70	0.0082	0.3904*	0.0482	0.0072
	20:80	0.0000	0.0000	0.0000	0.0000
spirals(n,cycles=2)	50:50	0.1309*	0.0842*	0.4907*	0.1496*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
spirals(n,cycles=4)	50:50	0.6483*	0.3558*	0.7408*	0.1588*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
spirals(n)	50:50	0.2941*	0.0155	0.0811*	0.0045
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000

From all problems, only 14 did not show statistical differences on the noise detection. 85% of those 14 cases occurred on balanced datasets. In this way, apart of observing variation on detection when the IR is 50:50, the tests indicate there is no significant difference in such cases. In other words, when dealing with balanced datasets, it is not possible to reject the hypothesis that NAR and NCAR produce equivalent influence on the ensemble detector performance. On the other hand, as expected from the experiments, Friedman test showed there is a different impact on noise detection according to the noise model applied when a IR is present.

Additionally, statistical comparisons in each of the problems were also performed.

Wilcoxon's signed ranks statistical test (DEMSAR, 2006) was applied to compare the impact on noise detection of each noise model. The Wilcoxon's signed ranks is a non-parametric pairwise test that aims to detect significant differences between two sample means, that is, the behavior of noise detection under the two noise models verified in each comparison. The test was applied with level of significance set to $\alpha = 0.05$ and the pairwise comparison was performed over all problems.

Table 14 – Wilcoxon test results when there is 15% noise in data. W/T/L = wins/ties/losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	5/0/4	5/0/4	8/0/1	8/0/1	9/0/0	8/0/1	8/0/1	8/0/1
		p-value	0.407	0.286	**0.013**	**0.013**	**0.009**	**0.013**	**0.018**	**0.013**
	NCAR	**W/T/L**		5/0/4	8/0/1	8/0/1	9/0/0	6/0/3	7/0/2	8/0/1
		p-value		0.906	**0.018**	**0.018**	**0.009**	0.097	0.058	**0.018**
	NAR(1:9)	**W/T/L**			6/0/3	7/0/2	7/0/2	4/0/5	4/0/5	4/0/5
		p-value			0.097	**0.033**	**0.024**	0.813	0.722	1.000
30:70	**NAR(9:1)**	**W/T/L**				9/0/0	8/0/1	6/0/3	8/0/1	8/0/1
		p-value				**0.009**	**0.013**	0.477	**0.013**	**0.024**
	NCAR	**W/T/L**					7/0/2	0/0/9	1/0/8	7/0/2
		p-value					0.058	**0.009**	**0.013**	0.155
	NAR(1:9)	**W/T/L**						0/0/9	0/0/9	2/0/7
		p-value						**0.009**	**0.009**	**0.033**
20:80	**NAR(9:1)**	**W/T/L**							8/0/1	9/0/0
		p-value							**0.013**	**0.009**
	NCAR	**W/T/L**								9/0/0
		p-value								**0.009**

The tests were performed for each different percentage of noise. As the results were equivalent (independently of the amount of noise), the following discussion will be regarding the noise percentage of 15% shown in Table 14. The remain results can be accessed in Appendix A.3.

In Table 14, a pairwise comparison on the noise detection results under each noise model is presented. The W/T/L denote the wins (better performance on noise detection), ties (equivalent performance on noise detection), and losses (worse performance on noise detection), produced by the noise models on the columns in comparison to the ones on the rows. For instance, the ensemble detector on problems under NAR(1:9) with 30:70 IR (column) performed 7 times better and 2 times worse (7/0/2) in comparison to the detection on problems under NAR(1:9) with 50:50 IR (row). For this same example, the *p-value* $= 0.024$ implies there is a significant difference in the results.

Focusing on problems under same IR, as discussed previously, the models seems not to have significant effect on detection on balanced problems as similar values of 5/0/4 for W/T/L and the *p-values* > 0.05 show. For the case of 30:70 IR, NAR(9:1) - more noise in minority class, harmed detection as its performance was worse 9 times when compared to detection under NCAR, and 8 times when compared to detection under NAR(1:9) - more noise in majority class. Lastly, when increasing IR (20:80), tests shows that noise detection is significantly improved under NAR(1:9) in comparison to the other noise models.

Results from statistical tests are aligned to the hypothesis raised in the previous section for imbalanced problems. In this type of problem, the ensemble detector performs better under the NAR(1:9) model than under NCAR and NAR(9:1) models; and it performs better under NCAR than NAR(9:1). For balanced problems, statistical tests did not show significant differences.

4.2 RESULTS ON REAL-WORLD DATA

The results shown in this section were obtained from the steps outlined in Algorithm 2. In the same way as in the previous section, the discussion is carried out putting into perspective each input parameter that resulted in a specific scenario in order to facilitate the analysis and comparison. Therefore, we first examine the noise detection under balanced and imbalanced datasets in Section 4.2.1. Then, we explore the impact of different noise level per class in Section 4.2.2. Lastly, in Section 4.2.3, we analyze noise detection under different noise ensemble thresholds.

4.2.1 Balanced vs Imbalanced Datasets

After generating datasets with specific IRs (Section 3.2.2), we repeatedly injected noise on all datasets described in Table 4. In Figures 16 and 17, we gathered some results to make discussion easier but the complete list of performance graphs can be assessed in Appendix B.

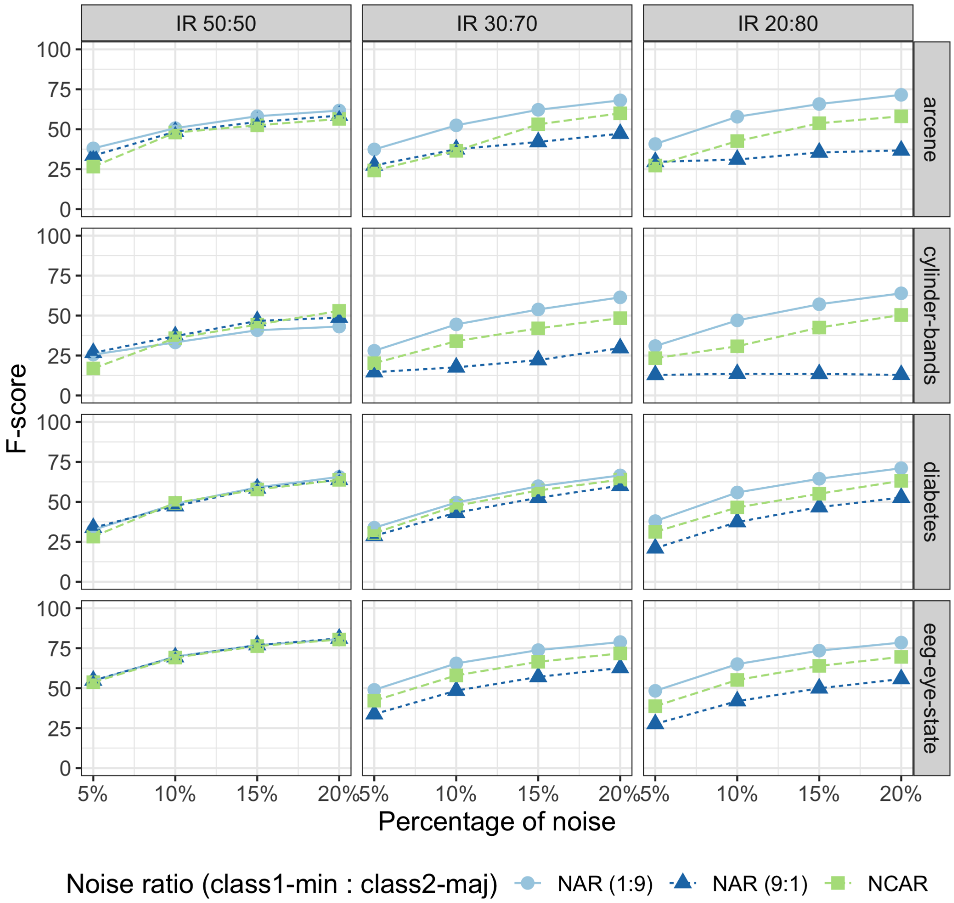

Figure 16 – *F-score* performance for majority vote on *arcene*, *cylinder-bands*, *diabetes*, and *eeg-eye-state* datasets.

Interestingly, as can be observed in the first column of Figures 16 and 17 and intuitively infer, the general behavior found on synthetic datasets was also verified in most real data. Likewise, the noise distribution per class seems not to be relevant in the scenario of balanced data (IR 50:50), that is, NCAR and NAR produce almost the same effect on noise detection. However, we found exceptions on *arcene* and *cylinder-bands* (Figure 16) with a slight discrepancy in results for NCAR and NAR models. This was also observed for *column2C*, *glass0*, *glass1*, and *sonar* datasets especially with 5% of noise level (Appendix B).

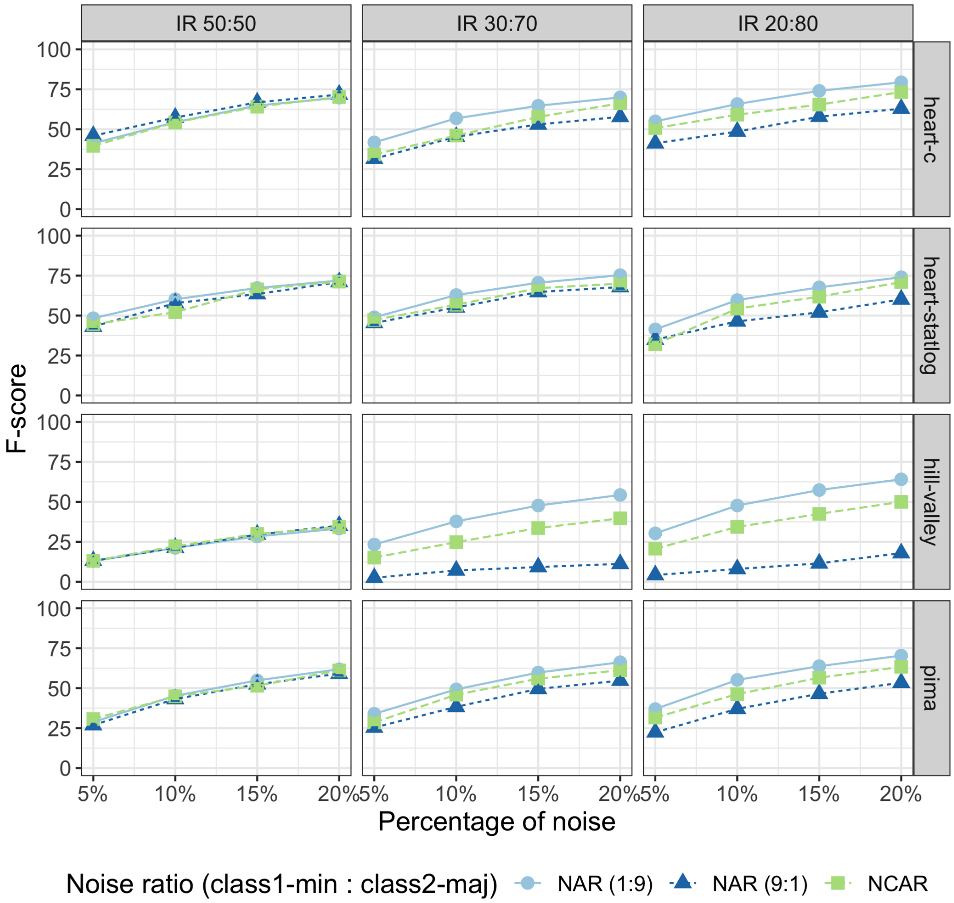

Figure 17 – *F-score* performance for majority vote on *heart-c*, *heart-statlog*, *hill-valley*, and *pima* datasets.

Although, a decrease in noise detection under NCAR model was expected from results on synthetic data, the numbers we found on real data experiments did not present a consistent pattern to confirm such behavior. On column NCAR of Table 15, which shows what happens to the *F-score* when the dataset varies from a balanced dataset to an imbalanced dataset (negative and positive number denote, respectively, a decrease and increase in noise detection), we can observe that there is no agreement or tendency in the results. Apparently, when the noise is evenly distributed in an imbalance dataset, the particularities and difficulties of the problem are more crucial in noise detection performance than the IR.

Table 15 – F-score variation vs class imbalance ratio in real data.

Datasets	**F-score variation when IR goes from 50:50 to 20:80**											
	5% of noise			**10% of noise**			**15% of noise**			**20% of noise**		
	NAR (9:1)	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
arcene	-3.6	1.1	2.8	-17.4	-5.4	7.1	-19.2	1.3	7.6	-21.6	1.7	9.8
breast-cancer-wisconsin	-1.1	-1.4	4.7	-0.4	0.0	3.8	-0.9	-0.6	2.2	-0.4	-0.1	1.7
column2C	-13.6	4.1	-2.8	-10.5	-2.1	-0.4	-15.6	1.7	1.8	-13.3	-2.4	0.9
credit	-5.8	-1.2	2.3	-8.6	0.6	1.6	-7.0	0.0	1.4	-6.6	-0.3	2.0
cylinder-bands	-13.8	6.6	5.4	-23.6	-5.1	13.8	-33.0	-1.8	16.2	-35.8	-2.6	20.8
diabetes	-12.8	3.0	5.5	-10.0	-2.9	7.2	-11.8	-2.6	5.4	-11.1	-0.6	5.4
eeg-eye-state	-27.2	-15.0	-6.3	-27.5	-14.0	-4.8	-27.0	-12.4	-3.4	-25.4	-10.9	-2.2
glass0	-1.5	1.9	-3.0	-0.5	8.8	7.9	-5.1	3.5	6.5	-4.5	3.3	8.9
glass1	-7.6	-6.2	-0.8	-18.1	3.1	8.8	-21.6	2.2	8.9	-27.0	-1.8	11.9
heart-c	-4.7	11.1	13.7	-8.8	4.9	11.4	-9.0	1.3	9.2	-8.8	3.1	9.6
heart-statlog	-8.2	-12.6	-7.0	-11.4	2.3	-0.5	-11.4	-4.6	0.4	-10.9	-0.3	2.1
hill-valley	-8.8	7.6	17.4	-13.3	12.1	26.5	-18.1	12.6	29.0	-17.2	15.7	30.8
ionosphere	-4.6	-5.3	3.6	-4.1	-2.8	3.3	-6.4	1.5	1.1	-4.7	0.1	1.9
kr-vs-kp	-8.7	-6.8	-7.4	-5.9	-3.9	-3.6	-4.7	-3.0	-3.0	-3.9	-2.1	-2.0
mushroom	-0.1	-0.1	-0.1	-0.0	-0.0	-0.1	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
pima	-4.5	0.9	8.2	-6.1	1.4	9.9	-5.9	5.0	8.9	-5.7	2.3	8.5
sonar	14.1	18.6	15.0	13.4	13.4	19.4	9.0	13.7	17.8	6.3	10.0	16.5
steel-plates-fault	0.0	1.0	0.4	-0.0	0.4	0.3	0.0	0.4	0.2	-0.0	0.1	0.1
tic-tac-toe	-22.1	-4.9	5.0	-26.1	-5.9	5.4	-26.6	-5.2	5.1	-27.1	-5.4	5.4
voting	-6.8	-13.6	-4.1	-5.8	-7.6	-2.5	-4.4	-6.2	-2.0	-4.7	-4.8	-1.4

4.2.2 Noise Level in the Minority vs Majority Class

Noise detection was impacted by class imbalance under the NAR model as expected from the findings on synthetic data. In the same way, *F-score* is degraded when the minority class is noisier than the majority class and it is improved or has an attenuated impact when the majority class is the noisiest one. This can be verified in the greater number of datasets as shown in Table 15 where we can observe a pattern of negative numbers on NAR(9:1) column and positive numbers on NAR(1:9) column for different levels of noise.

Like in the experiments on synthetic data, Figures 18 and 19 show a better visualization of the general behavior of noise detection under NAR model when IR increases. The results are aligned to the ones found previously. This can be observed by the bigger and negative bar on the left side of the graphs in contrast to a smaller and positive bar on the right.

The aforementioned behavior was found in all data with the exception of *sonar* dataset. This may imply that the noise model characteristics and class imbalance ratio, in general, have more influence on the noise detection than the nature of the classification problem for the majority of the problems.

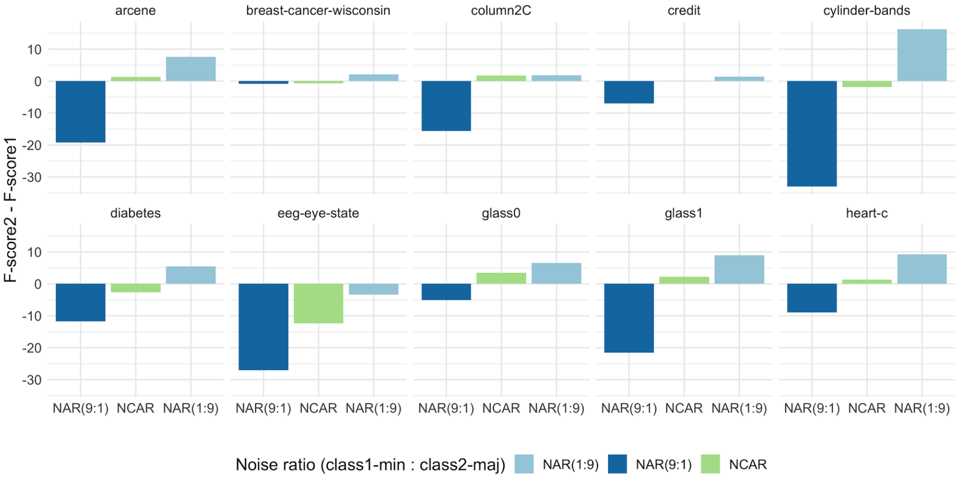

Figure 18 – Variation in *F-score* performance when IR increases from 50:50 (F-score1) to 20:80 (F-score2) in presence of a noise level of 15% - parte 1.

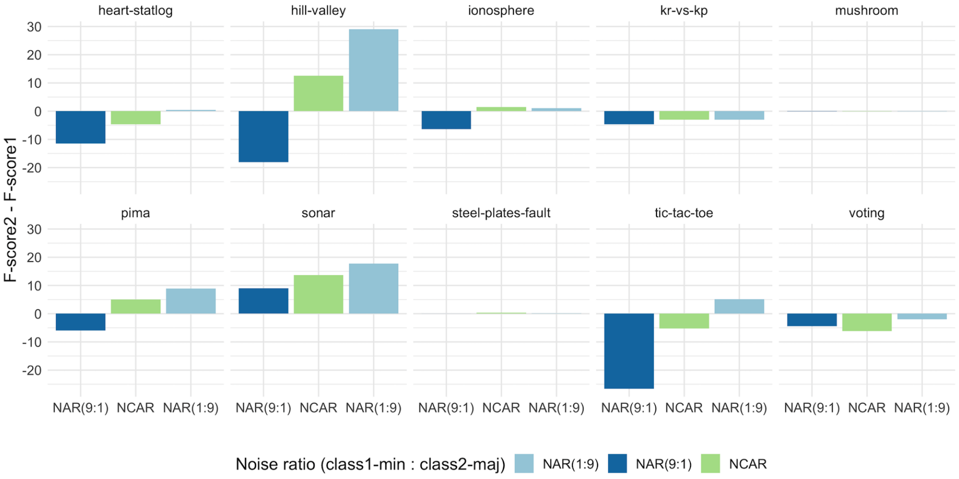

Figure 19 – Variation in *F-score* performance when IR increases from 50:50 (F-score1) to 20:80 (F-score2) in presence of a noise level of 15% - parte 2.

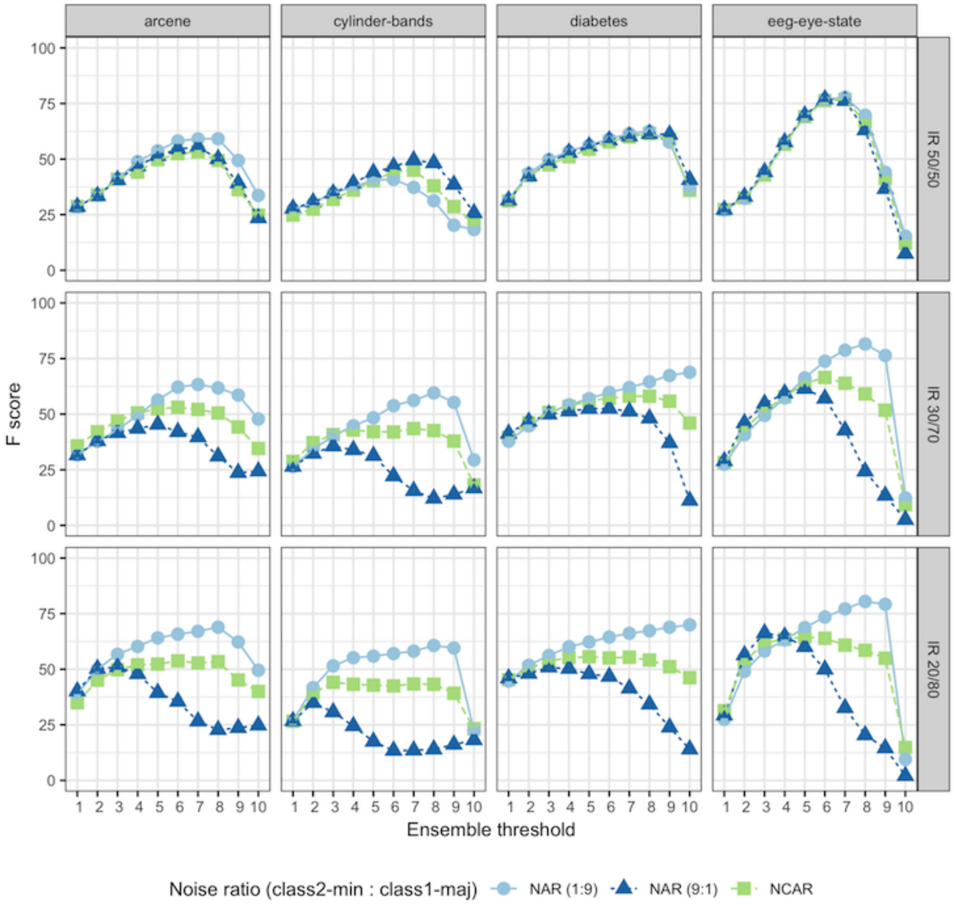

Figure 20 – *F-score* on *arcene*, *cylinder-bands*, *diabetes*, and *eeg-eye-state* datasets under different ensemble vote thresholds (where 1 = 10%, 2 = 20%,..,10 = 100%) in presence of 15% of noise.

4.2.3 Noise Detection vs Ensemble Vote Thresholds

Similarly to the experiments performed on synthetic data, we also evaluated the noise detection under different ensemble vote thresholds on real data. The findings are aligned with previous results for the majority of datasets. Results were gathered in Figures 20 and 21.

In Section 4.1.3, we divided the analysis under two variables: imbalance ratio and percentage of noise. As the latter did not present much influence on the results, here we will focus on the former. In this way, we gathered the main results in Figures 20 and 21, which show the noise detection performance for each threshold at 15% of noise level under NAR and NCAR models. The complete result can be assessed in Appendix B.

As can be seen, most data present the same behavior: better noise detection is achieved with smaller threshold values under the NAR (9:1) model, and with higher threshold values under the NAR (1:9) model when IR is increased. Under the NCAR model, threshold

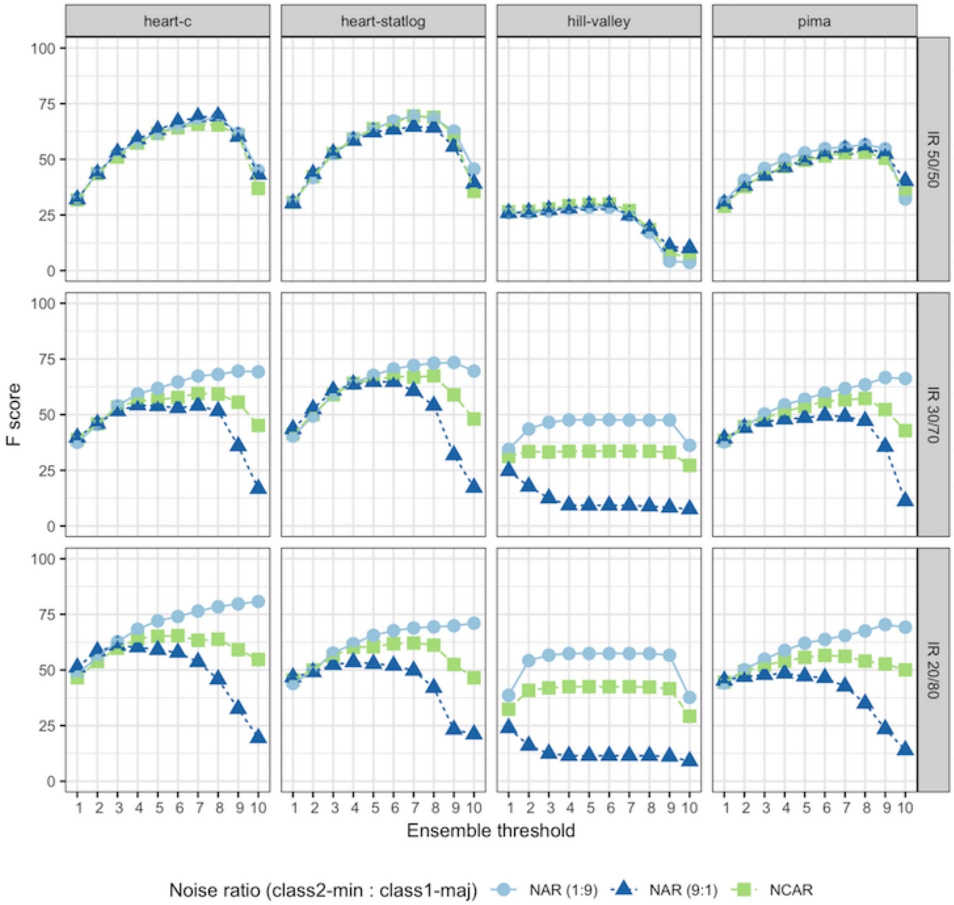

Figure 21 – *F-score* on *heart-c*, *heart-statlog*, *hill-valley*, and *pima* datasets under different ensemble vote thresholds (where 1 = 10%, 2 = 20%,..,10 = 100%) in presence of 15% of noise

values close to $L = 5$ (majority vote) return higher *F-score* results.

The above behavior is verified in a more or less pronounced way depending on the dataset. For example, in Figure 20 for *arcene* dataset, the best threshold under the NAR(9:1) model is $L = 7$ with 55.4 of performance when IR is 50:50, $L = 5$ with 45.3 when IR is 30:70, and $L = 3$ with 51.0 for an IR of 20:80. Under the same setting, for *pima* dataset in Figure 21, the best threshold under the NAR(9:1) model is $L = 8$ with 55.2 of performance when IR is 50:50, $L = 6$ with 49.5 when IR is 30:70, and $L = 4$ with 48.5 for an IR of 20:80. Values are different but the general behavior is the same.

Results on real data validate, for the majority of cases, the findings on synthetic data regarding differences in the way noise model can influence the noise detection under a specific context.

4.2.4 Statistical tests

The Friedman test (FRIEDMAN; RAFSKY, 1979) was also performed in order to compare the impact of all three noise models over the 228 problems (19^2 datasets X 3 IR's X 4 different percentages of noise). Since we are comparing the effect of three different noise models on ensemble detection, the degree of freedom is 2. The level of significance was set to $\alpha = 0.05$, i.e., 95% confidence. All *p-values* obtained can be accessed in the Appendix B.3 in Tables 21 and 23. In order to facilitate analysis, a summary of the results is presented in Table 16.

The Friedman test shows that there is a significant difference on the detection noise for the three models in certain contexts. As shown in Table 16, from the 152 imbalanced data problems analyzed, 77.63% (118/152) presented a significant difference on the detection results. When considering only the problems with 20:80 IR, this number is equal to 88.16%. On the other hand, when it comes to balanced datasets (76 of cases), only 18.42% are significant different. These results are aligned with the hypothesis discussed in the previous section. The choice of a noise generation model are more likely to have impact on detection results in data-imbalance problems.

Table 16 – Summary of Friedman test results on each problem.

IR	Cases with significant difference								Total per IR	
	5% of noise		**10% of noise**		**15% of noise**		**20% of noise**			
50:50	6/19	31.6%	1/19	5.3%	6/19	31.6%	1/19	5.3%	14/76	18.4%
30:70	11/19	57.9%	14/19	73.7%	12/19	63.2%	14/19	73.7%	51/76	67.1%
20:80	15/19	78.9%	16/19	84.2%	18/19	94.7%	18/19	94.7%	67/76	88.2%
Total per noise	32/57	56.1%	31/57	54.4%	36/57	63.2%	33/57	57.9%		

The influence of the amount of noise on ensemble detection was also tested. From the 57 problems for each different percentage of noise, approximately half of the cases presented significant difference (56.1% for 5% of noise, 54.4% for 10%, 63.2% for 15% and 57.9% for 20%). In this way, the amount of noise in data seems not be as relevant as the IR on noise detection under different noise models.

A second statistical analysis was also conducted in a pairwise fashion in order to verify if the noise models significantly improve/harm the noise detection under certain contexts. To that end, the Wilcoxon non-parametric signed rank test with the level of significance $\alpha = 0.05$ was used over all problems.

The tests were performed for each different percentage of noise. As the results were equivalent (independently of the amount of noise), the following discussion will be regarding the noise percentage of 15% shown in Table 17. The remain results can be accessed in Appendix B.3.

[2] Mushroom dataset was removed due to its 100% precision.

Table 17 – Wilcoxon test on real problems when there is 15% noise in data. W \T \L = wins\ties\losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	5/0/14	11/1/7	16/0/3	17/0/2	15/0/4	17/0/2	17/0/2	16/0/3
		p-value	0.073	0.360	**0.001**	**0.000**	**0.001**	**0.001**	**0.001**	**0.001**
	NCAR	**W/T/L**		15/0/4	11/0/8	10/0/9	9/0/10	13/0/6	8/0/11	12/0/7
		p-value		**0.038**	0.481	0.952	0.324	0.409	0.952	0.153
	NAR(1:9)	**W/T/L**			6/0/13	4/0/15	4/0/15	3/0/16	3/0/16	3/0/16
		p-value			**0.021**	**0.005**	**0.007**	**0.003**	**0.002**	**0.004**
30:70	**NAR(9:1)**	**W/T/L**				17/0/2	17/0/2	16/0/3	16/0/3	17/0/2
		p-value				**0.000**	**0.000**	**0.012**	**0.001**	**0.000**
	NCAR	**W/T/L**					17/0/2	4/0/15	9/0/10	15/0/4
		p-value					**0.000**	**0.004**	0.856	**0.004**
	NAR(1:9)	**W/T/L**						1/0/18	4/0/15	7/0/12
		p-value						**0.000**	**0.001**	**0.035**
20:80	**NAR(9:1)**	**W/T/L**							18/0/1	19/0/0
		p-value							**0.000**	**0.000**
	NCAR	**W/T/L**								19/0/0
		p-value								**0.000**

In Table 17, a pairwise comparison on the noise detection results under each noise model is presented. Once again, the W/T/L denote the wins (better performance on noise detection), ties (equivalent performance on noise detection), and losses (worse performance on noise detection), produced by the noise models on the columns in comparison to the ones on the rows. For instance, the ensemble detector on problems under NAR(1:9) with 30:70 IR (column) performed 15 times worse and 4 times better (4/0/15) in comparison to the detection on problems under NAR(1:9) with 50:50 IR (row). For this same example, the *p-value* = 0.007 implies there is a significant difference in the results.

Focusing on problems under same IR, as discussed previously, only one case showed significant difference. Under balanced data, NAR(1:9) produced a positive impact on noise detection, performing 15 times out of 19 better than the detection under NCAR although with a *p-value* = 0.038. This same behavior was found when in presence of 5% of noise (Table 18) but not for 10% of noise (Table 19) and 20% of noise (Table 20). On the other hand, for the case of 30:70 IR, NAR(9:1) - more noise in minority class, harmed detection as its performance was worse 17 times when compared to the detection under NCAR and NAR(1:9) - more noise in majority - with a really small *p-value*. This was also verified when data was exposed to different percentage of noise. Lastly, when increasing IR (20:80), tests showed noise detection is significantly improved under NAR(1:9) in comparison to the other noise models as the ensemble detector performed better in all problems (19/0/0).

4.3 CHAPTER REMARKS

In this chapter, the results obtained from the execution of the methodology proposed in Section 3.4 were presented. More than 100 problems were accessed for synthetic data analysis and more than 200 for real-data analysis. The problems were created as to contemplate a combination of three main inputs: class imbalance ratio, amount of noise, and noise distribution per class.

For each dataset, three different class imbalance ratio were created: 50:50 (balanced), 30:70, and 20:80 (imbalanced). Noise was injected in 5%, 10%, 15% and 20% of the data, and distributed applying three different models: (1) NCAR - noise injected equally between classes, (2) NAR, by injecting a high proportion of label noise in the majority class, and (3) NAR, by injecting a high proportion of label noise in the minority class. The three models were named (1) NCAR, (2) NAR(1:9), and (3) NAR(9:1), to simplify reference.

On each problem combination, the ensemble noise detector was evaluated using the *F-score* as the main variable. The majority vote scheme was chosen to combine algorithms predictions and, then, other thresholds for ensemble voting were also analyzed. Results indicated different effect on noise detection according to the context and noise model applied.

On imbalance-class problems, the noise detection performance is significantly better under NAR(1:9) model in comparison to NAR and NAR(9:1) independently on the amount of noise in data. The ensemble detector also performs better under NCAR model than NAR(9:1). In other words, more presence of noise in minority class, makes noise detection more difficult. On balanced-problems, no significant results were found, although noise detection presents similar behavior independently on the model used.

When varying the ensemble threshold, experiments show that a smaller number of voting algorithms delivers better noise detection under NAR(1:9) model, and that a higher threshold produces better performance under NAR(9:1). Finally, under NCAR models, the majority vote performs better.

5 CONCLUSIONS

Many studies have focused their attention on data quality issues due to its importance in ML applications and also due to the known fact that real-world datasets frequently contain noise (FRENAY; VERLEYSEN, 2014).

Noise can be present in data in its attributes and also in its classes (ZHU; WU, 2004). This work is focused on class noise (also label noise) behavior. For this type of problem, *classification noise filtering* approach is usually applied so to remove data irregularities prior to the learning step. The most common filtering approach consists of using the predictions of an ensemble of algorithms so that instances are removed upon a wrong classification (BRODLEY; FRIEDL, 1999)(SLUBAN; LAVRA, 2015a)(GUAN et al., 2018).

In order to evaluate Noise Filters, simulated noise is usually injected into a dataset, and analysis are performed on the detection results (GARCIA et al., 2019). In (FRENAY; VERLEYSEN, 2014), three different label noise generation models are presented: (1) NCAR, in which the probability of an instance being noisy is random, (2) NAR, the probability of an instance being noisy depends on its label, and (3) NNAR, the probability of an instance being noisy depends also on its attributes.

Although there are many approaches to model different noise behaviors, in many previous works (SLUBAN; LAVRA, 2015a) (BRODLEY; FRIEDL, 1999) (SAEZ et al., 2015) (GARCIA et al., 2019), one type of noise is chosen over another without considering the different impacts of each one. Also, in spite of the *majority* and *consensus* being the most common ensemble voting schemes used for filtering noise, studies (KHOSHGOFTAAR; ZHONG; JOSHI, 2005)(SABZEVARI; MARTINEZ-MUNOZ; SUAREZ, 2018) have shown that selecting adequate values for the ensemble threshold can lead to superior results.

In this work, it was presented an empirical study focused on an ensemble-based noise detector and its performance under three different noise generation models: NCAR, where noise is equally distributed among class; NAR model, where the majority class is noisier than the minority class, and NAR, where the minority class is the noisiest one. Detection performance versus injected noise model relation was assessed through performance measures (F-score, Precision, Recall) considering different inserted noise ratios and imbalance class configurations. The impact produced on the filtering performance was also evaluated under different ensemble thresholds.

In the next sections, some conclusions, considerations and future works are presented.

5.1 CONTRIBUTIONS

The main contributions of this work consist of the following findings:

- **Noise detection is not affected by the noise model in balanced data.** In this scenario, no major change in detection was observed under the three noise models applied. NCAR and both NAR models presented equivalent detection rates independently of the amount of noise in data.

- **Noise detection is harmed when the minority class is noisier than the majority.** When dealing with imbalanced class and choosing the NAR model applying more noise in the minority class, the noise detection was harmed in overall problems (synthetic and real).

- **Noise detection is improved when the majority class is noisier than the minority.** When dealing with imbalanced class and choosing the NAR model applying more noise in the majority class, the noise detection was improved in most problems.

- **High-imbalance class increases the impact on noise detection.** In our studies, class imbalance was increased from a 30:70 IR to a 20:80 IR in every problem. Increasing the ratio intensified the results found when NAR was applied. In other words, the noise detection was even worse when minority class was the noisiest one and the opposite was verified for the majority class.

- **Noise equally distributed does not influence the noise detection.** When noise is evenly distributed between classes (NCAR model), no consistent results on noise detection were found throughout the problems when dealing with imbalanced data.

- **Noise detection under NCAR or NAR(9:1) is improved when applying a smaller ensemble threshold**. The experiments performed with different ensemble thresholds showed that the *majority* and *consensus* voting schemes are not always the best options. Better noise detection was achieved for both models (NCAR and NAR 9:1 - when there is more noise in minority class) if less than 50% of the algorithms are selected.

- **Noise detection under NAR(1:9) is improved when applying a higher ensemble threshold**. In the experiments, it was observed that better noise detection is achieved for NAR 1:9 (when there is more noise in majority class) if more than 50% of the algorithms are selected.

5.2 FUTURE WORKS

Although interesting behaviors were found in this work, this research is a preliminary study and important aspects are open to be investigated in a more comprehensive context in future works. Following are some activities that can be undertaken:

- Include other noise filtering techniques (as those ones described in Chapter 2) in order to check if the results are also verified for other noise detection approaches.
- NNAR model should also be analyzed in contrast to the noise models studied in this research.
- Apply a new set of algorithms for the ensemble or add more algorithms or use different combinations to verify if results are not method-dependent.
- Study ways to model the different scenarios discussed in this work in order to find the best ensemble threshold to improve ensemble-based noise filters.
- Expand the work for multi-class problems.

5.3 PUBLICATION

The partial findings of this study resulted in the following publication:

> MOURA de, K. G.; PRUDENCIO, R. B. C.; CAVALCANTI, G. D. C. “Ensemble Methods for Label Noise Detection Under the Noisy at Random Model”, 7th Brazilian Conference on Intelligent Systems (BRACIS), Sao Paulo, 2018, pp. 474-479.

REFERENCES

ABELLÁN, J.; MASEGOSA, A. R. Bagging decision trees on data sets with classification noise. In: LINK, S.; PRADE, H. (Ed.). *Foundations of Information and Knowledge Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 248–265. ISBN 978-3-642-11829-6.

ALCALA-FDEZ A. FERNANDEZ, J. L. J. D. S. G. L. S. F. H. J. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, v. 17, n. 2-3, p. 255–287, 2011. Disponível em: <https://sci2s.ugr.es/keel/datasets.php>.

BIGGIO, B.; NELSON, B.; LASKOV, P. Support vector machines under adversarial label noise. In: HSU, C.-N.; LEE, W. S. (Ed.). *Proceedings of the Asian Conference on Machine Learning*. South Garden Hotels and Resorts, Taoyuan, Taiwain: PMLR, 2011. (Proceedings of Machine Learning Research, v. 20), p. 97–112. Disponível em: <http://proceedings.mlr.press/v20/biggio11.html>.

BOOTKRAJANG, J. A generalised label noise model for classification in the presence of annotation errors. *Neurocomputing*, v. 192, p. 61 – 71, 2016. ISSN 0925-2312. Advances in artificial neural networks, machine learning and computational intelligence. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231216002551>.

BOOTKRAJANG, J.; KABAN, A. Learning kernel logistic regression in the presence of class label noise. *Pattern Recognition*, v. 47, n. 11, p. 3641 – 3655, 2014. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320314001927>.

BRODLEY, C. E.; FRIEDL, M. A. Identifying mislabeled training data. In: *Artifficial Intelligence Review*. [S.l.: s.n.], 1999. p. 131–167.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 01 2006.

DUA, D.; TANISKIDOU, E. K. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>.

FRENAY, B.; VERLEYSEN, M. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, v. 25, n. 5, p. 845–869, May 2014. ISSN 2162-237X.

FRIEDMAN, J. H.; RAFSKY, L. C. Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *Ann. Statist.*, The Institute of Mathematical Statistics, v. 7, n. 4, p. 697–717, 07 1979. Disponível em: <https://doi.org/10.1214/aos/1176344722>.

GALAR A. FERNANDEZ, E. B. T. H. B. S. M.; HERRERA, F. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. In: *IEEE Trans. on Syst., Man, and Cybernetics, Part C (Applications and Reviews)*. [S.l.: s.n.], 2012. p. 463–484.

GAMBERGER, D.; BOSKOVIC, R.; LAVRAC, N.; GROSELJ, C. Experiments with noise filtering in a medical domain. In: *Proc. of 16 th ICML*. [S.l.]: Morgan Kaufmann, 1999. p. 143–151.

GAMBERGER, D.; LAVRAČ, N. Conditions for occam's razor applicability and noise elimination. In: SOMEREN, M. van; WIDMER, G. (Ed.). *Machine Learning: ECML-97*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. p. 108–123. ISBN 978-3-540-68708-5.

GARCIA-GIL, D.; LUENGO, J.; GARCIA, S.; HERRERA, F. Enabling smart data: Noise filtering in big data classification. *Information Sciences*, v. 479, p. 135 – 152, 2019. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025518309460>.

GARCIA, L. P.; LEHMANN, J.; CARVALHO, A. C. de; LORENA, A. C. New label noise injection methods for the evaluation of noise filters. *Knowledge-Based Systems*, v. 163, p. 693 – 704, 2019. ISSN 0950-7051. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950705118304829>.

Garcia, L. P. F.; Lorena, A. C.; Carvalho, A. C. P. L. F. A study on class noise detection and elimination. In: *2012 Brazilian Symposium on Neural Networks*. [S.l.: s.n.], 2012. p. 13–18. ISSN 2375-0235.

GILBERT, R.; MARTIN, R. M.; DONOVAN, J.; LANE, J. A.; HAMDY, F.; NEAL, D. E.; METCALFE, C. Misclassification of outcome in case–control studies: Methods for sensitivity analysis. *Statistical Methods in Medical Research*, v. 25, n. 5, p. 2377–2393, 2016. PMID: 25217446. Disponível em: <https://doi.org/10.1177/0962280214523192>.

GUAN, D.; WEI, H.; YUAN, W.; HAN, G.; TIAN, Y.; AL-DHELAAN, M.; AL-DHELAAN, A. Improving label noise filtering by exploiting unlabeled data. *IEEE Access*, v. 6, p. 11154–11165, 2018. ISSN 2169-3536.

HAN, J.; KAMBER, M.; PEI, J. Data mining concepts and techniques, third edition. In: . Waltham, Mass.: Morgan Kaufmann Publishers, 2012. ISBN 0123814790. Disponível em: <http://www.amazon.de/Data-Mining-Concepts-Techniques-Management/dp/0123814790/ref=tmm_hrd_title_0?ie=UTF8&qid=1366039033&sr=1-1>.

KANJ, S.; ABDALLAH, F.; DENŒUX, T.; TOUT, K. Editing training data for multi-label classification with the k-nearest neighbor rule. *Pattern Analysis and Applications*, v. 19, n. 1, p. 145–161, Feb 2016. ISSN 1433-755X. Disponível em: <https://doi.org/10.1007/s10044-015-0452-8>.

Khoshgoftaar, T. M.; Rebours, P. Generating multiple noise elimination filters with the ensemble-partitioning filter. In: *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.* [S.l.: s.n.], 2004. p. 369–375.

KHOSHGOFTAAR, T. M.; ZHONG, S.; JOSHI, V. Enhancing software quality estimation using ensemble-classifier based noise filtering. *Intell. Data Anal.*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 9, n. 1, p. 3–27, jan. 2005. ISSN 1088-467X. Disponível em: <http://dl.acm.org/citation.cfm?id=1239046.1239048>.

LAWRENCE, N. D.; SCHöLKOPF, B. Estimating a kernel fisher discriminant in the presence of label noise. In: *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 306–313. ISBN 1-55860-778-1. Disponível em: <http://dl.acm.org/citation.cfm?id=645530.655665>.

LI, Y.; WESSELS, L. F.; RIDDER, D. de; REINDERS, M. J. Classification in the presence of class noise using a probabilistic kernel fisher method. *Pattern Recognition*, v. 40, n. 12, p. 3349 – 3357, 2007. ISSN 0031-3203. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320307002166>.

LORENA, A. C.; CARVALHO, A. C. P. L. F. d. Evaluation of noise reduction techniques in the splice junction recognition problem. *Genetics and Molecular Biology*, scielo, v. 27, p. 665 – 672, 00 2004. ISSN 1415-4757. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-47572004000400031&nrm=iso>.

MALOSSINI, A.; BLANZIERI, E.; NG, R. T. Detecting potential labeling errors in microarrays by data perturbation. *Bioinformatics*, v. 22, n. 17, p. 2114–2121, 06 2006. ISSN 1367-4803. Disponível em: <https://dx.doi.org/10.1093/bioinformatics/btl346>.

SABZEVARI, M.; MARTINEZ-MUNOZ, G.; SUAREZ, A. A two-stage ensemble method for the detection of class-label noise. *Neurocomputing*, v. 275, p. 2374 – 2383, 2018. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0925231217317265>.

SAEZ, J. A.; LUENGO, J.; STEFANOWSKI, J.; HERRERA, F. Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, v. 291, p. 184–203, 01 2015.

SLUBAN, B.; GAMBERGER, D.; LAVRAČ, N. Ensemble-based noise detection: noise ranking and visual performance evaluation. *Data Mining and Knowledge Discovery*, v. 28, n. 2, p. 265–303, Mar 2014. ISSN 1573-756X. Disponível em: <https://doi.org/10.1007/s10618-012-0299-1>.

SLUBAN, B.; LAVRA, N. Relating ensemble diversity and performance: a study in class noise detection. In: *Neurocomputing*. [S.l.: s.n.], 2015. p. 120–131.

SLUBAN, D. G. B.; LAVRA, N. Advances in class noise detection. In: *Proc. of the 19th European Conference on Artificial Intelligence*. [S.l.: s.n.], 2015. p. 1105–1106.

SMITH, M. R.; MARTINEZ, T.; GIRAUD-CARRIER, C. An instance level analysis of data complexity. *Machine Learning*, v. 95, n. 2, p. 225–256, May 2014. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/s10994-013-5422-z>.

SUN, J.; ZHAO, F.; WANG, C.; CHEN, S. Identifying and correcting mislabeled training instances. In: *Future Generation Communication and Networking (FGCN 2007)*. [S.l.: s.n.], 2007. v. 1, p. 244–250. ISSN 2153-1447.

VALENTINI, G. An experimental bias-variance analysis of svm ensembles based on resampling techniques. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 35, n. 6, p. 1252–1271, Dec 2005. ISSN 1083-4419.

VANSCHOREN, J.; RIJN, J. N. van; BISCHL, B.; TORGO, L. Openml: Networked science in machine learning. *SIGKDD Explorations*, ACM, New York, NY, USA, v. 15, n. 2, p. 49–60, 2013. Disponível em: <http://doi.acm.org/10.1145/2641190.2641198>.

VERBAETEN, S.; ASSCHE, A. V. Ensemble methods for noise elimination in classification problems. In: WINDEATT, T.; ROLI, F. (Ed.). *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 317–325. ISBN 978-3-540-44938-6.

WILSON, D. R.; MARTINEZ, T. R. Reduction techniques for instance-based learning algorithms. *Machine Learning*, v. 38, n. 3, p. 257–286, Mar 2000. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1007626913721>.

YUAN, W.; GUAN, D.; MA, T.; KHATTAK, A. M. Classification with class noises through probabilistic sampling. *Information Fusion*, v. 41, p. 57 – 67, 2018. ISSN 1566-2535. Disponível em: <http://www.sciencedirect.com/science/article/pii/S156625351730221X>.

ZHU, X.; WU, X. Class noise vs. attribute noise: A quantitative study. In: *Artifficial Intelligence Review*. [S.l.: s.n.], 2004. p. 177–210.

ZHU X., W. X. . C. Q. Eliminating class noise in large datasets. In: *In 20th International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2003. p. 920–927.

APPENDIX A – RESULTS ON SYNTHETIC DATA

A.1 PERFORMANCE MEASURES

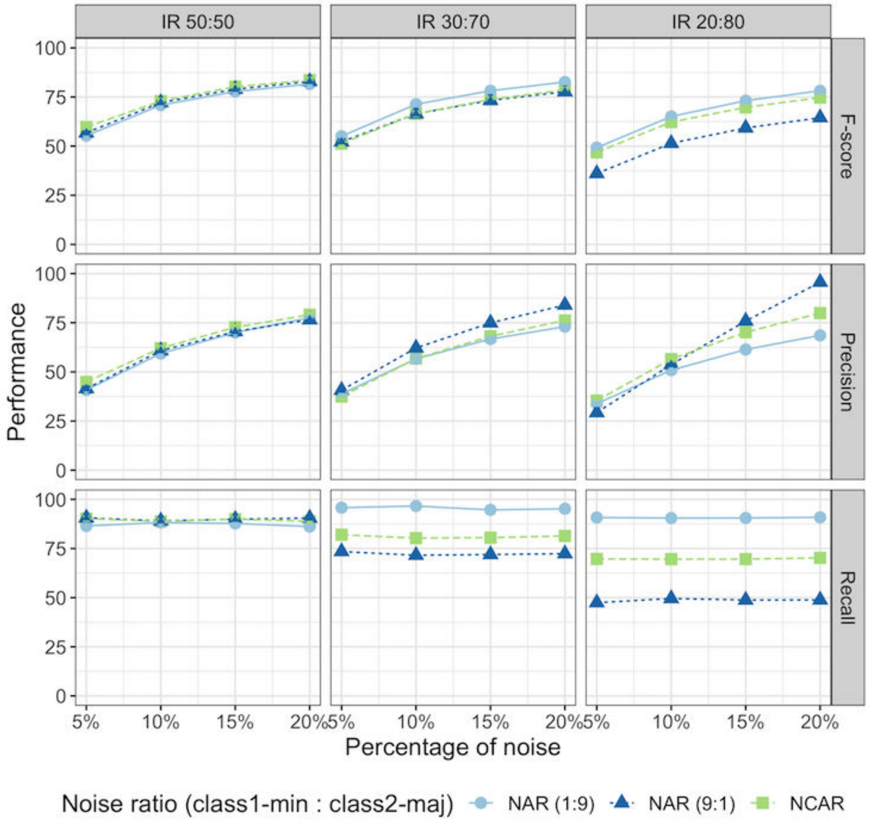

Figure 22 – Noise detection performance for majority vote on *P2* problem when class 1 is the minority class.

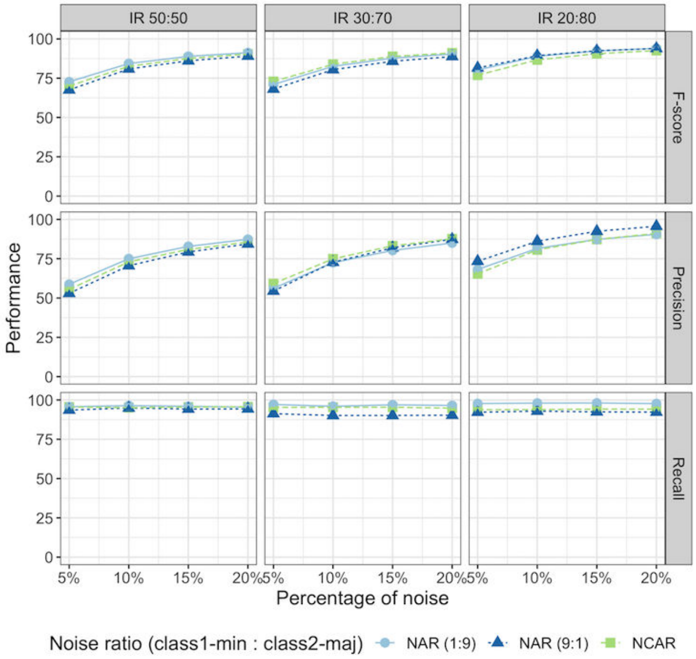

Figure 23 – Noise detection performance for majority vote on *2dnormals(n,cl=2)* dataset when class 1 is the minority class.

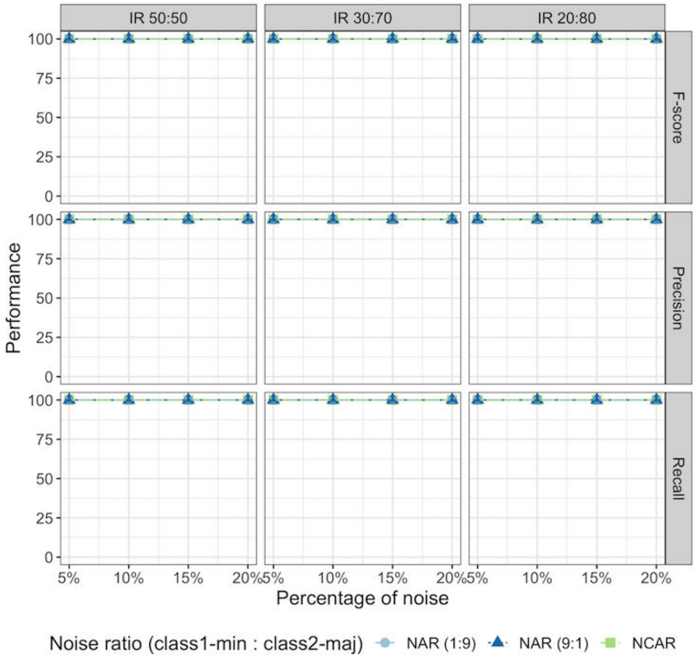

Figure 24 – Noise detection performance for majority vote on *cassini(n)* dataset when class 1 is the minority class.

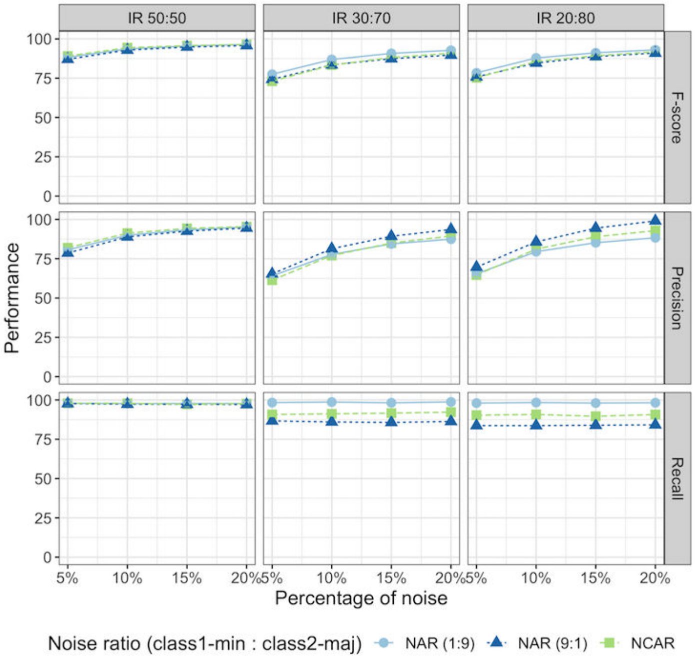

Figure 25 – Noise detection performance for majority vote on *circle(n,d=2)* dataset when class 1 is the minority class.

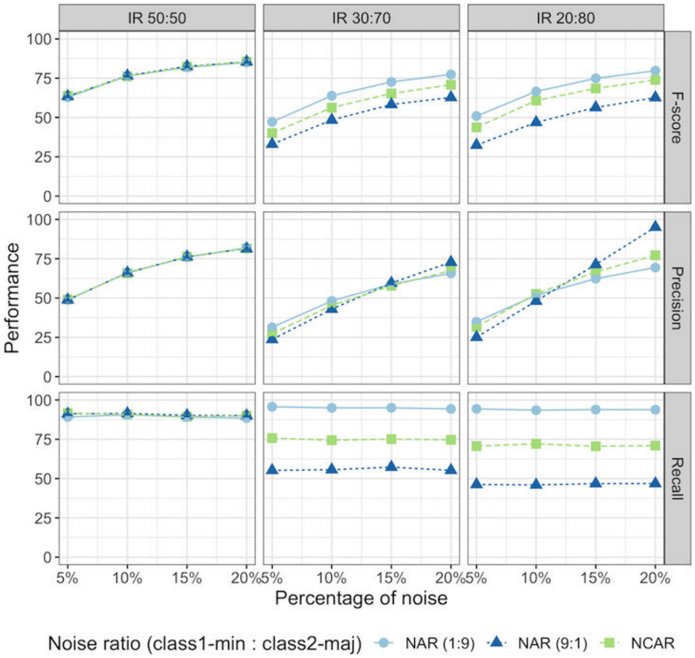

Figure 26 – Noise detection performance for majority vote on *circle(n,d=5)* dataset when class 1 is the minority class.

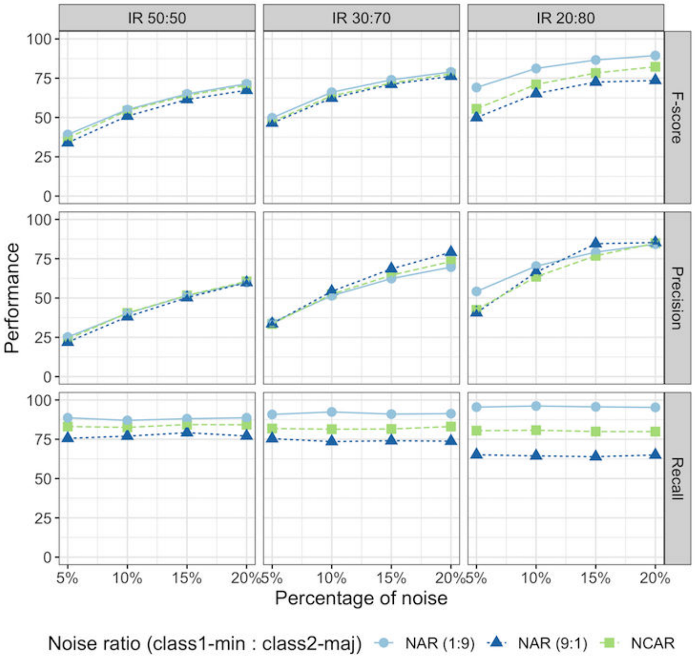

Figure 27 – Noise detection performance for majority vote on *ringnorm(n,d=2)* dataset when class 1 is the minority class.

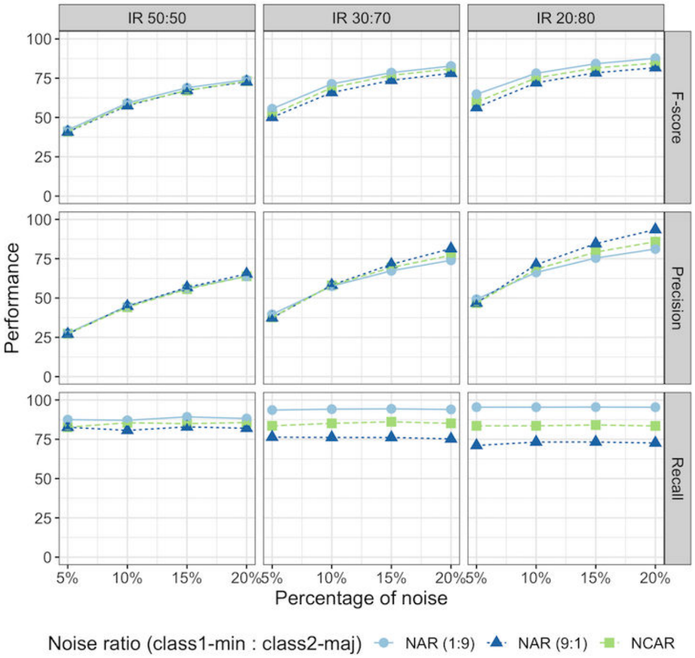

Figure 28 – Noise detection performance for majority vote on *ringnorm(n,d=5)* dataset when class 1 is the minority class.

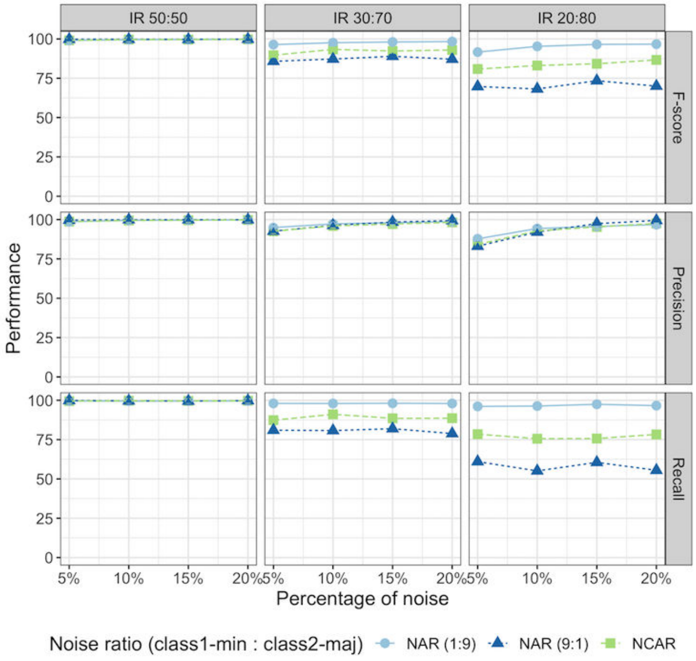

Figure 29 – Noise detection performance for majority vote on *spirals(n)* dataset when class 1 is the minority class.

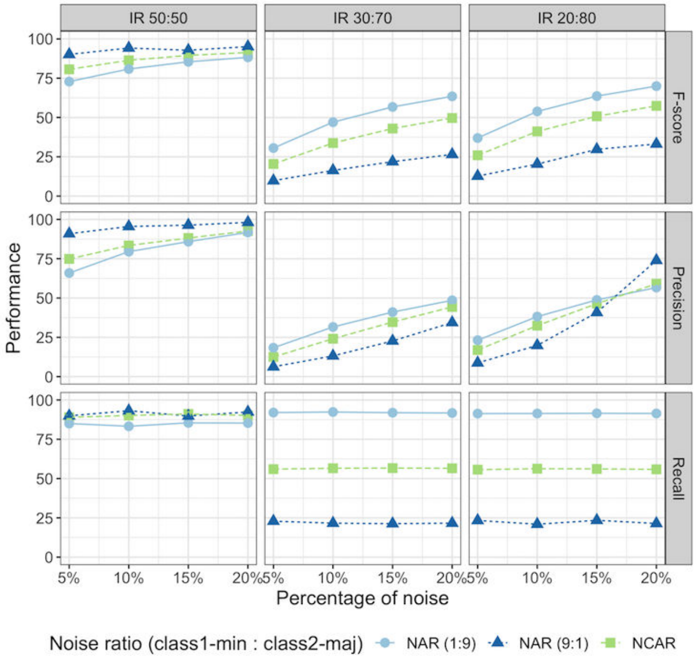

Figure 30 – Noise detection performance for majority vote on *spirals(n,cycles=2)* dataset when class 1 is the minority class.

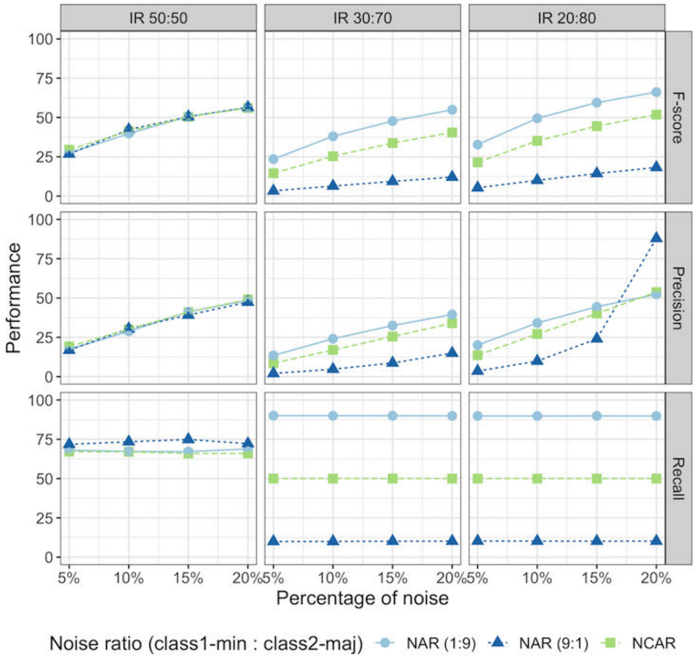

Figure 31 – Noise detection performance for majority vote on *spirals(n,cycles=4)* dataset when class 1 is the minority class.

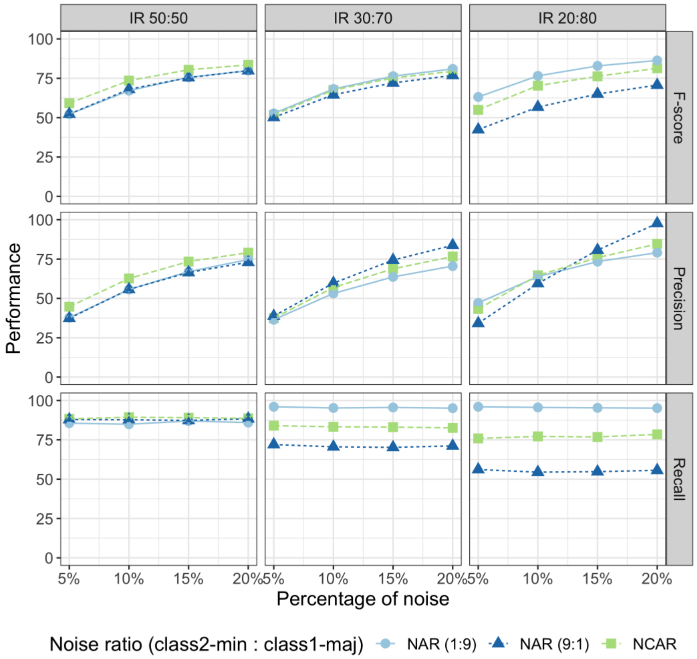

Figure 32 – Noise detection performance for majority vote on *P2* problem when class 2 is the minority class.

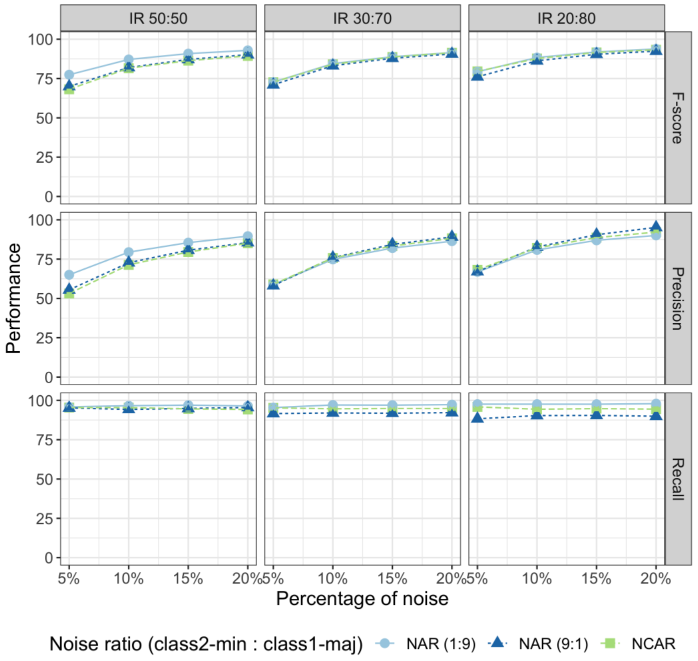

Figure 33 – Noise detection performance for majority vote on *2dnormals(n,cl=2)* dataset when class 2 is the minority class.

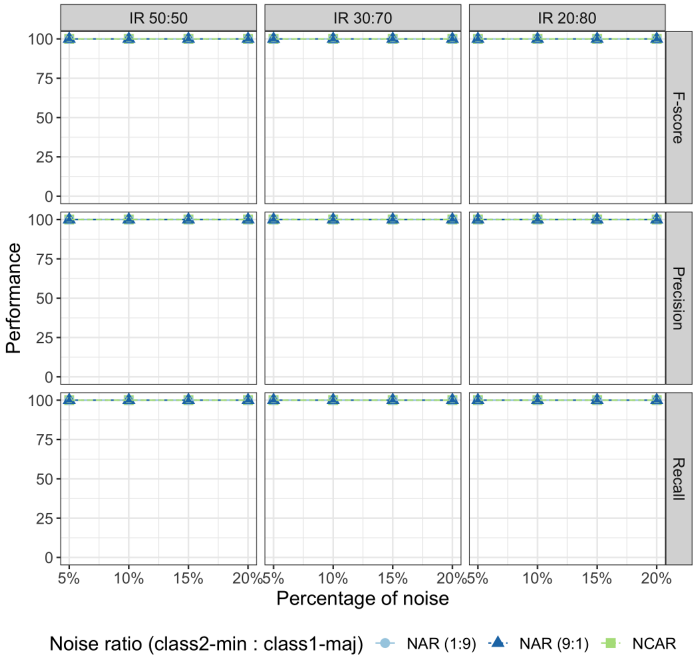

Figure 34 – Noise detection performance for majority vote on *cassini(n)* dataset when class 2 is the minority class.

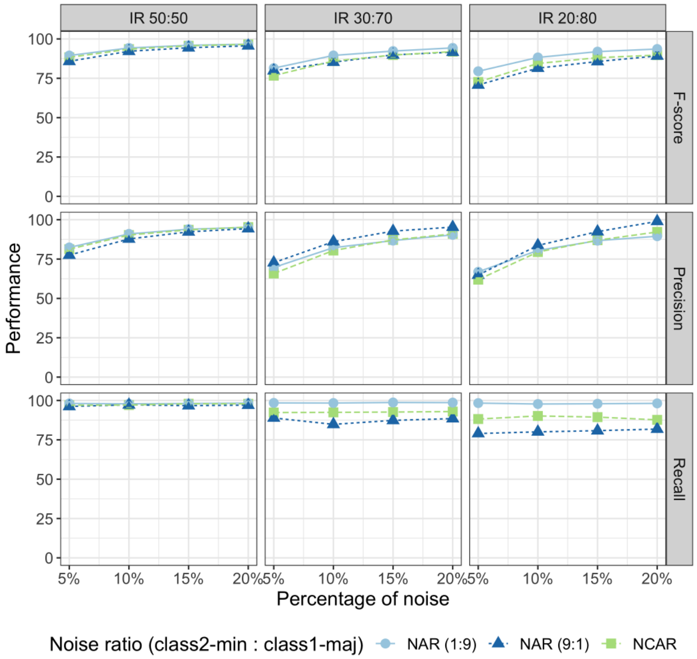

Figure 35 – Noise detection performance for majority vote on *circle(n,d=2)* dataset when class 2 is the minority class.

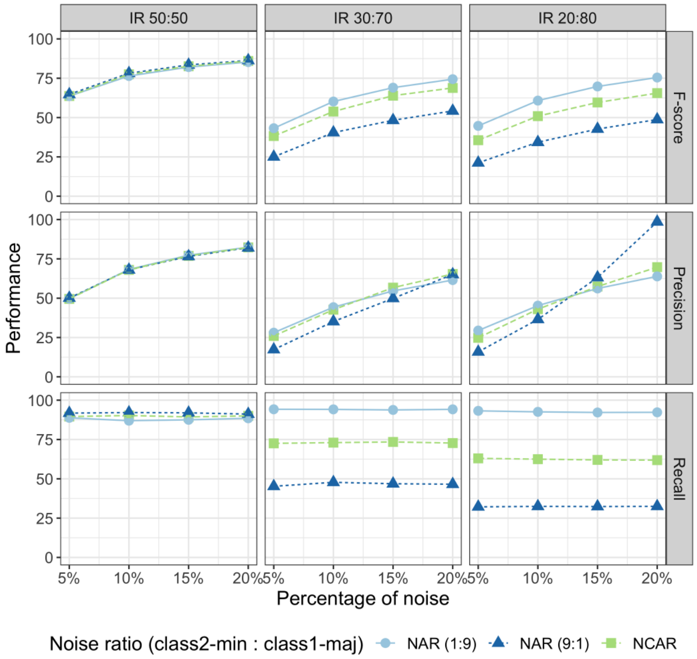

Figure 36 – Noise detection performance for majority vote on *circle(n,d=5)* dataset when class 2 is the minority class.

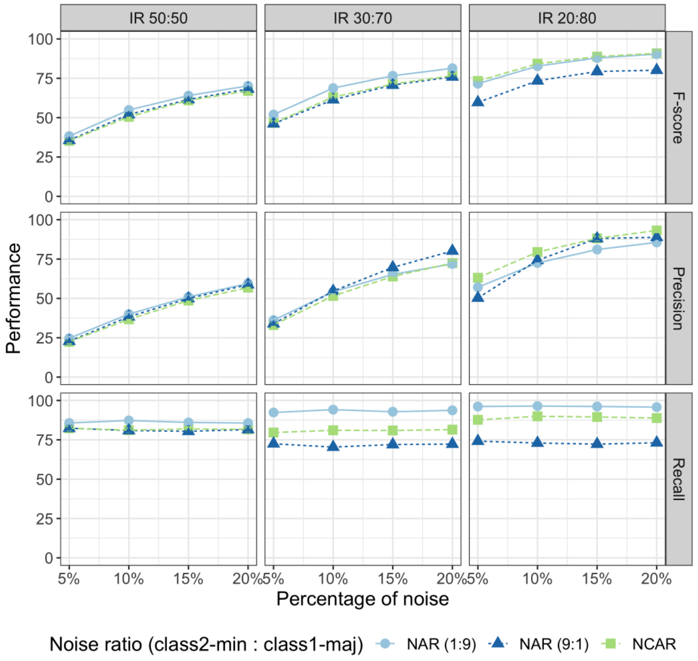

Figure 37 – Noise detection performance for majority vote on *ringnorm(n,d=2)* dataset when class 2 is the minority class.

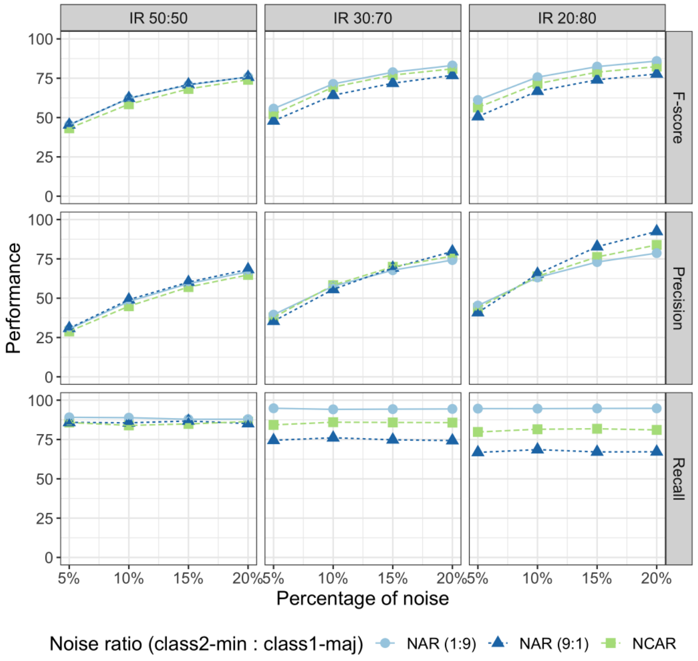

Figure 38 – Noise detection performance for majority vote on *ringnorm(n,d=5)* dataset when class 2 is the minority class.

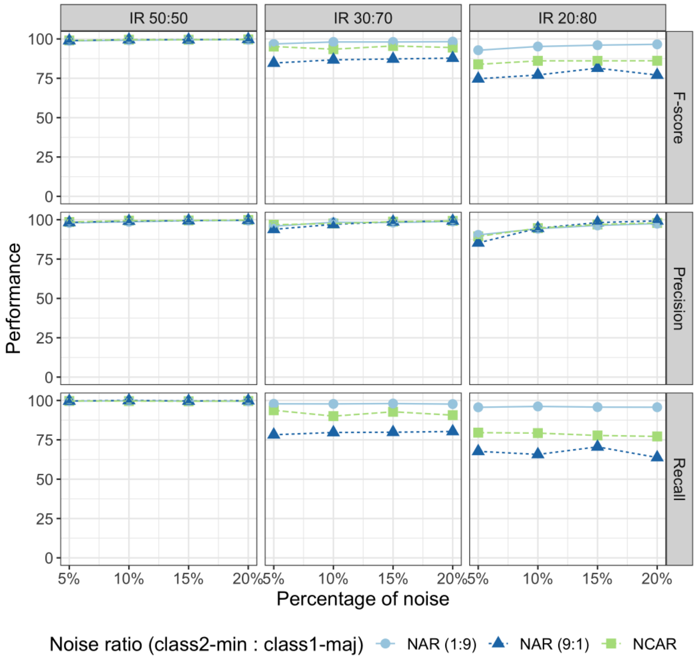

Figure 39 – Noise detection performance for majority vote on *spirals(n)* dataset when class 2 is the minority class.

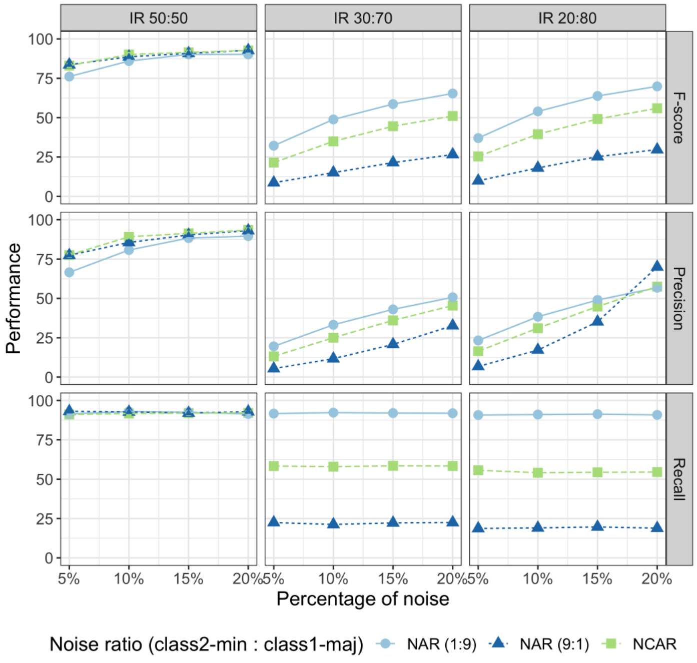

Figure 40 – Noise detection performance for majority vote on *spirals(n,cycles=2)* dataset when class 2 is the minority class.

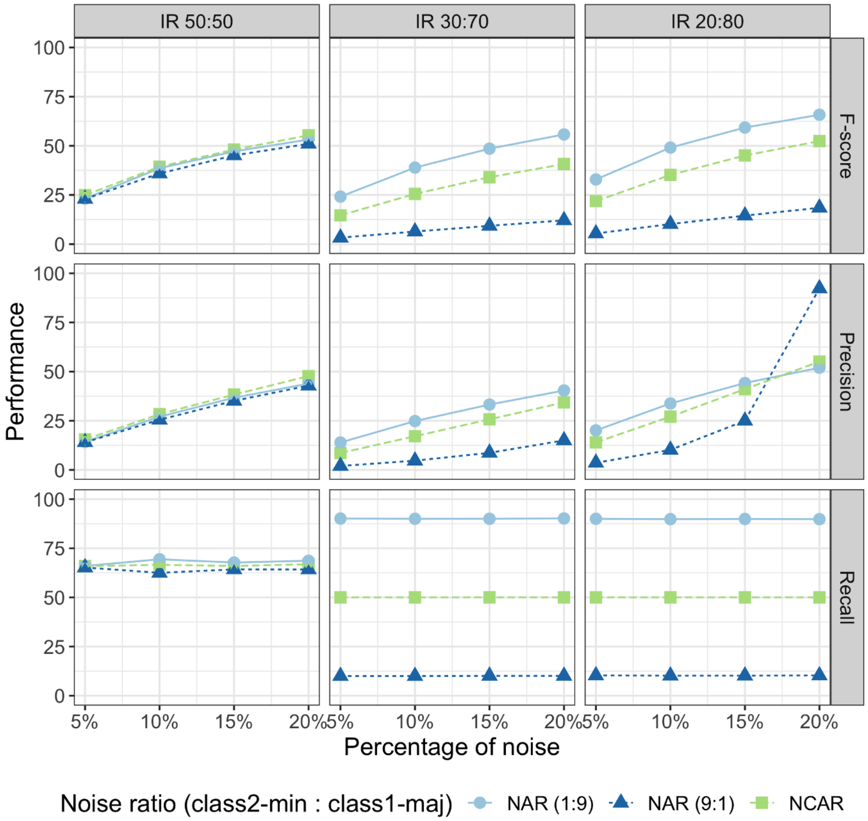

Figure 41 – Noise detection performance for majority vote on *spirals(n,cycles=4)* dataset when class 2 is the minority class.

A.2 ENSEMBLE VOTE THRESHOLD

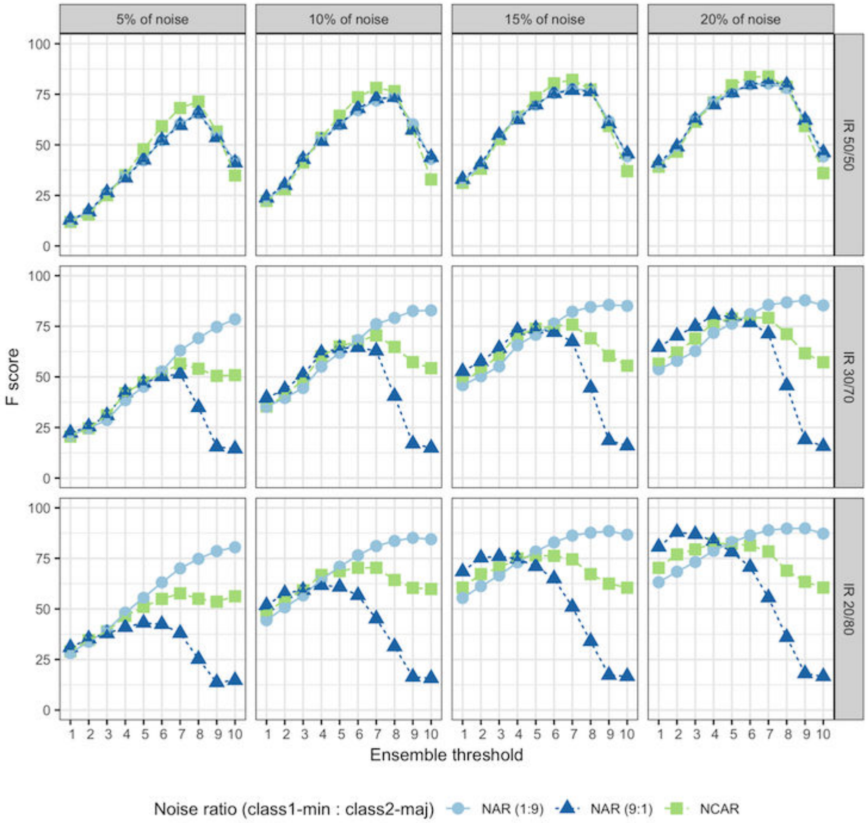

Figure 42 – *F-score* on *P2* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

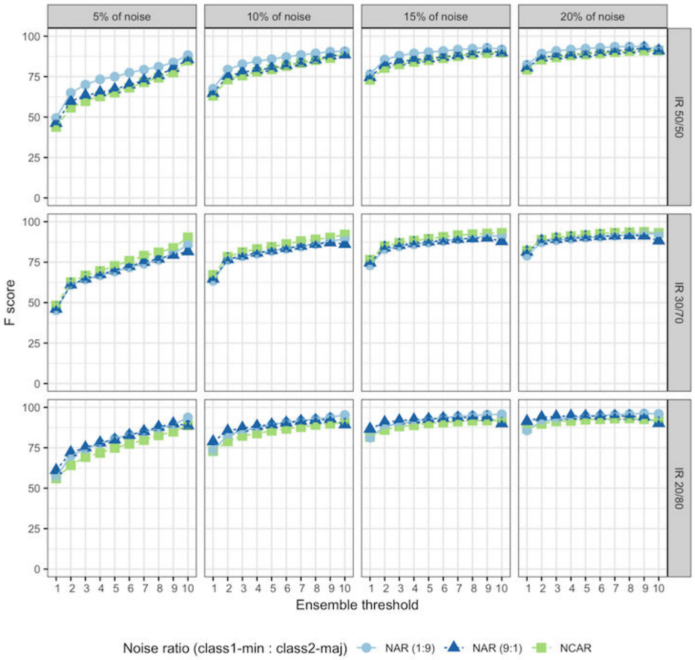

Figure 43 – *F-score* on *2dnormals(n,cl=2)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

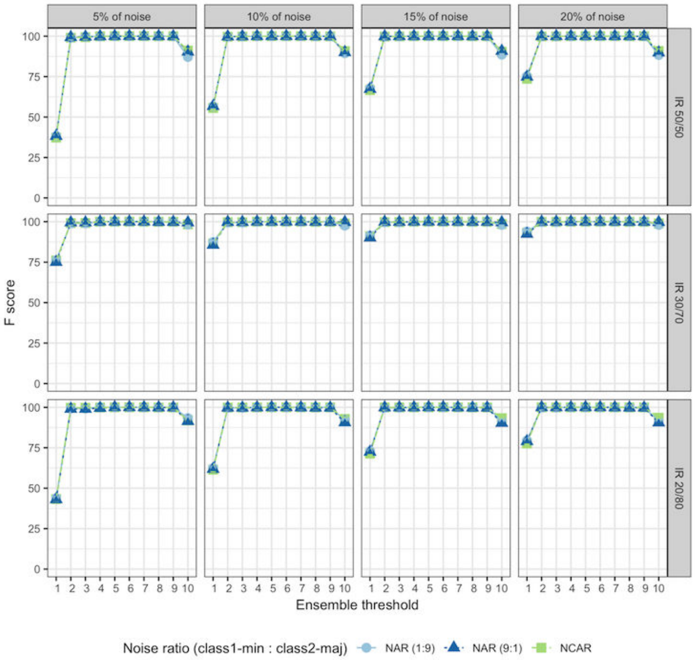

Figure 44 – *F-score* on *cassini(n)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

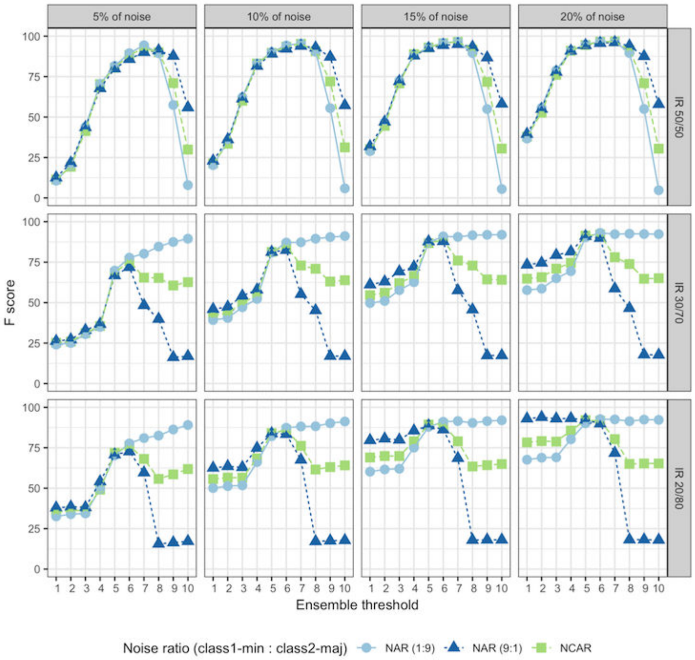

Figure 45 – *F-score* on *circle(n,d=2)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

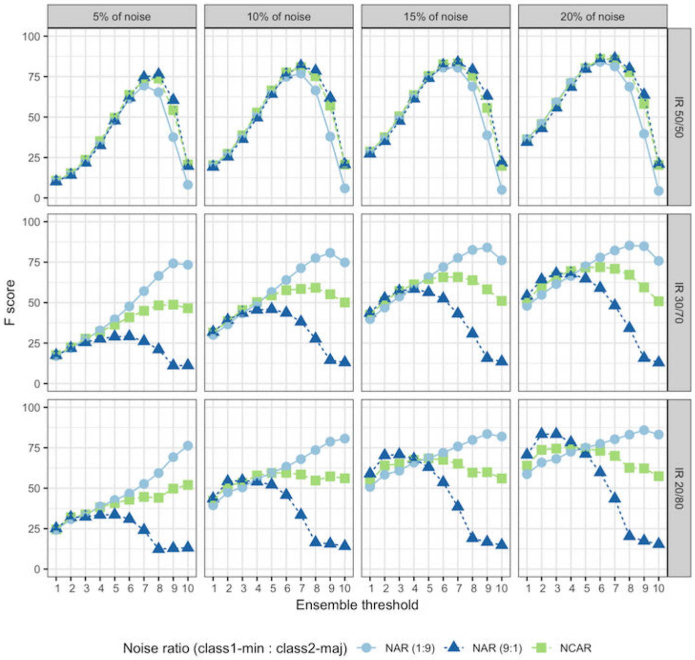

Figure 46 – *F-score* on *circle(n,d=5)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

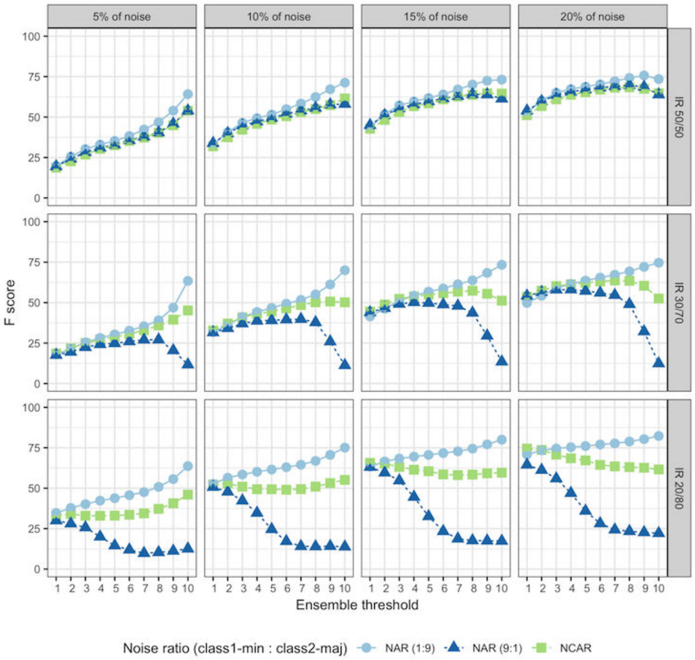

Figure 47 – *F-score* on *ringnorm(n,d=2)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

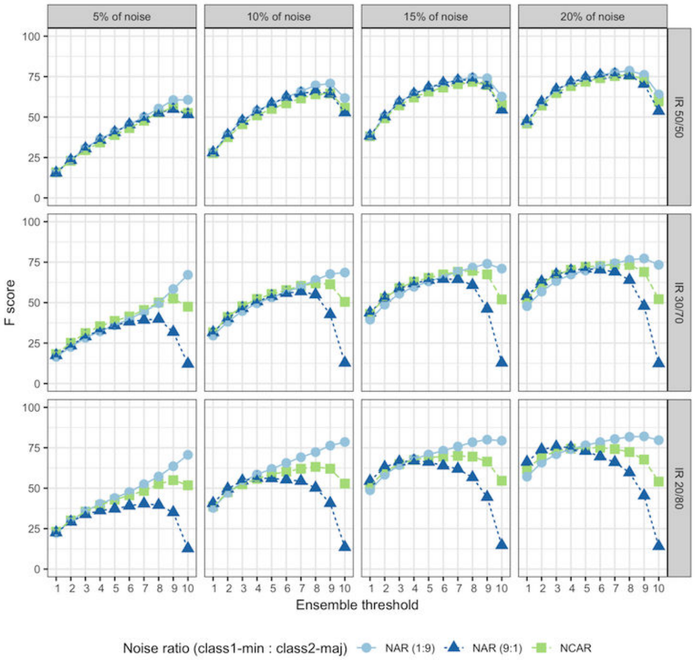

Figure 48 – *F-score* on *ringnorm(n,d=5)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

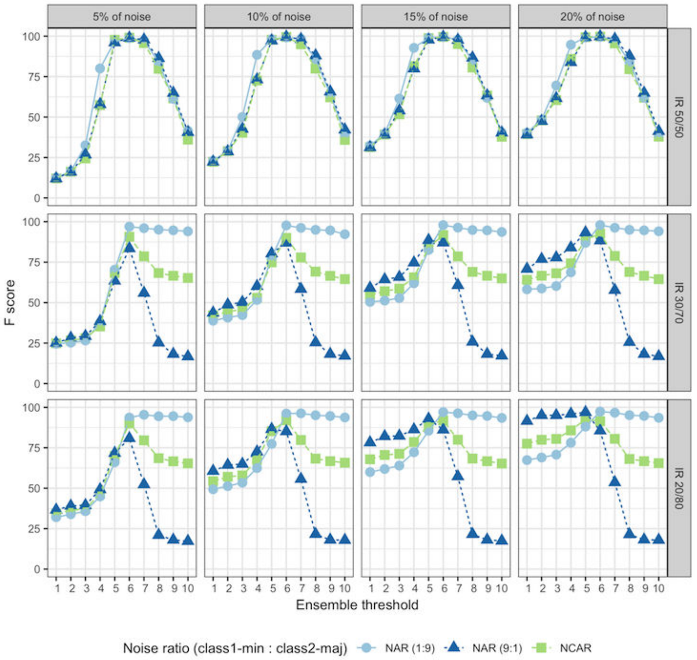

Figure 49 – *F-score* on *spirals(n)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

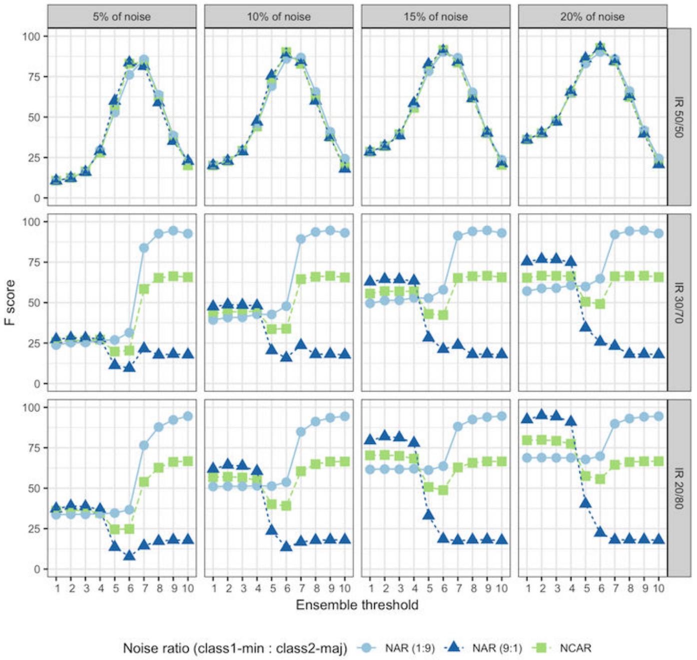

Figure 50 – *F-score* on *spirals(n,cycles=2)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

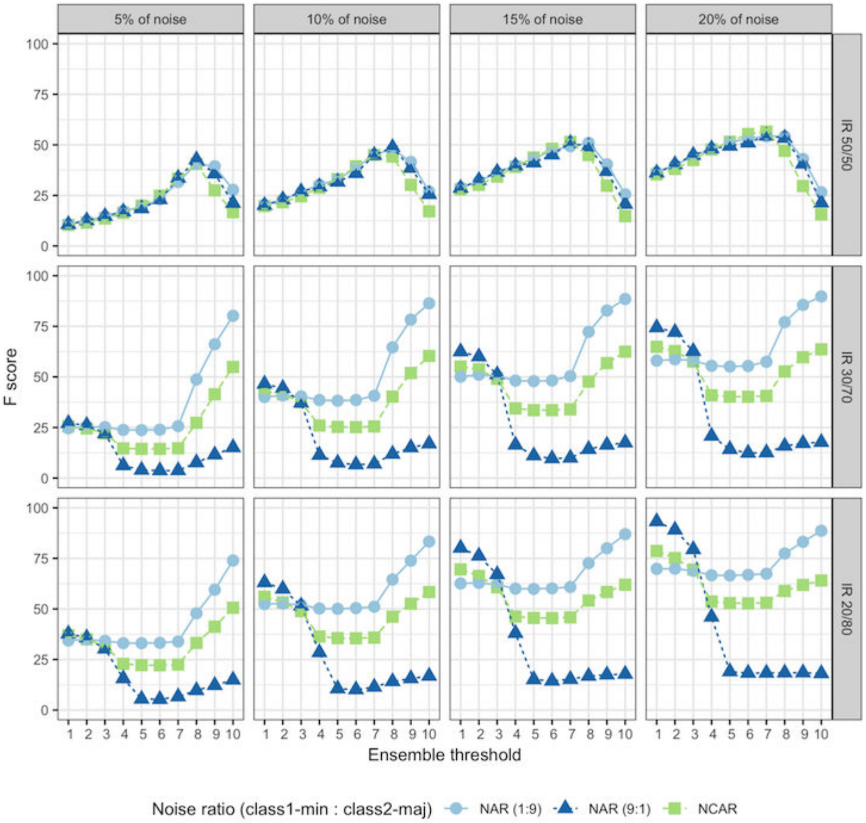

Figure 51 – *F-score* on *spirals(n,cycles=4)* dataset under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

A.3 STATISTICAL TESTS

Table 18 – Wilcoxon test on synthetic problems when there is 5% of noise in data. W/T/L = wins/ties/losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	5/0/4	4/0/5	8/0/1	8/0/1	9/0/0	8/0/1	8/0/1	8/0/1
		p-value	0.554	0.906	**0.018**	**0.013**	**0.009**	**0.024**	**0.018**	**0.013**
	NCAR	**W/T/L**		4/0/5	8/0/1	8/0/1	9/0/0	6/0/3	7/0/2	7/0/2
		p-value		0.906	**0.033**	**0.033**	**0.009**	0.155	0.097	0.058
	NAR(1:9)	**W/T/L**			6/0/3	8/0/1	7/0/2	4/0/5	4/0/5	4/0/5
		p-value			0.058	**0.044**	**0.024**	1.000	0.722	0.722
30:70	**NAR(9:1)**	**W/T/L**				9/0/0	8/0/1	4/0/5	8/0/1	8/0/1
		p-value				**0.009**	**0.013**	0.722	**0.024**	**0.033**
	NCAR	**W/T/L**					7/0/2	0/0/9	1/0/8	5/0/4
		p-value					0.076	**0.009**	**0.018**	0.636
	NAR(1:9)	**W/T/L**						0/0/9	0/0/9	3/0/6
		p-value						**0.009**	**0.009**	0.058
20:80	**NAR(9:1)**	**W/T/L**							8/0/1	9/0/0
		p-value							**0.018**	**0.009**
	NCAR	**W/T/L**								9/0/0
		p-value								**0.009**

Table 19 – Wilcoxon test on synthetic problems when there is 10% of noise in data. W/T/L = wins/ties/losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	5/0/4	4/0/5	8/0/1	8/0/1	9/0/0	8/0/1	8/0/1	8/0/1
		p-value	0.554	0.554	**0.013**	**0.013**	**0.009**	**0.018**	**0.018**	**0.013**
	NCAR	**W/T/L**		4/0/5	8/0/1	8/0/1	8/0/1	7/0/2	7/0/2	8/0/1
		p-value		1.000	**0.018**	**0.024**	**0.013**	0.097	0.076	**0.033**
	NAR(1:9)	**W/T/L**			6/0/3	8/0/1	8/0/1	4/0/5	4/0/5	4/0/5
		p-value			0.076	**0.024**	**0.018**	0.906	1.000	0.906
30:70	**NAR(9:1)**	**W/T/L**				9/0/0	8/0/1	5/0/4	8/0/1	8/0/1
		p-value				**0.009**	**0.013**	0.722	**0.024**	**0.024**
	NCAR	**W/T/L**					7/0/2	0/0/9	0/0/9	6/0/3
		p-value					0.076	**0.009**	**0.009**	0.343
	NAR(1:9)	**W/T/L**						0/0/9	0/0/9	2/0/7
		p-value						**0.009**	**0.009**	**0.044**
20:80	**NAR(9:1)**	**W/T/L**							8/0/1	9/0/0
		p-value							**0.018**	**0.009**
	NCAR	**W/T/L**								9/0/0
		p-value								**0.009**

Table 20 – Wilcoxon test on synthetic problems when there is 20% of noise in data. W/T/L = wins/ties/losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	4/0/5	5/0/4	8/0/1	8/0/1	9/0/0	8/0/1	8/0/1	8/0/1
		p-value	0.813	0.554	**0.013**	**0.013**	**0.009**	**0.013**	**0.018**	**0.013**
	NCAR	**W/T/L**		4/0/5	8/0/1	8/0/1	9/0/0	6/0/3	7/0/2	8/0/1
		p-value		0.722	**0.018**	**0.018**	**0.009**	0.124	0.058	**0.033**
	NAR(1:9)	**W/T/L**			6/0/3	7/0/2	7/0/2	4/0/5	4/0/5	4/0/5
		p-value			0.155	**0.024**	**0.033**	0.722	0.722	0.906
30:70	**NAR(9:1)**	**W/T/L**				9/0/0	8/0/1	5/0/4	8/0/1	8/0/1
		p-value				**0.009**	**0.013**	0.554	**0.013**	**0.024**
	NCAR	**W/T/L**					7/0/2	0/0/9	2/0/7	6/0/3
		p-value					**0.044**	**0.009**	0.058	0.155
	NAR(1:9)	**W/T/L**						0/0/9	0/0/9	3/0/6
		p-value						**0.009**	**0.009**	0.058
20:80	**NAR(9:1)**	**W/T/L**							8/0/1	9/0/0
		p-value							**0.018**	**0.009**
	NCAR	**W/T/L**								9/0/0
		p-value								**0.009**

APPENDIX B – RESULTS ON REAL DATA

B.1 PERFORMANCE MEASURES

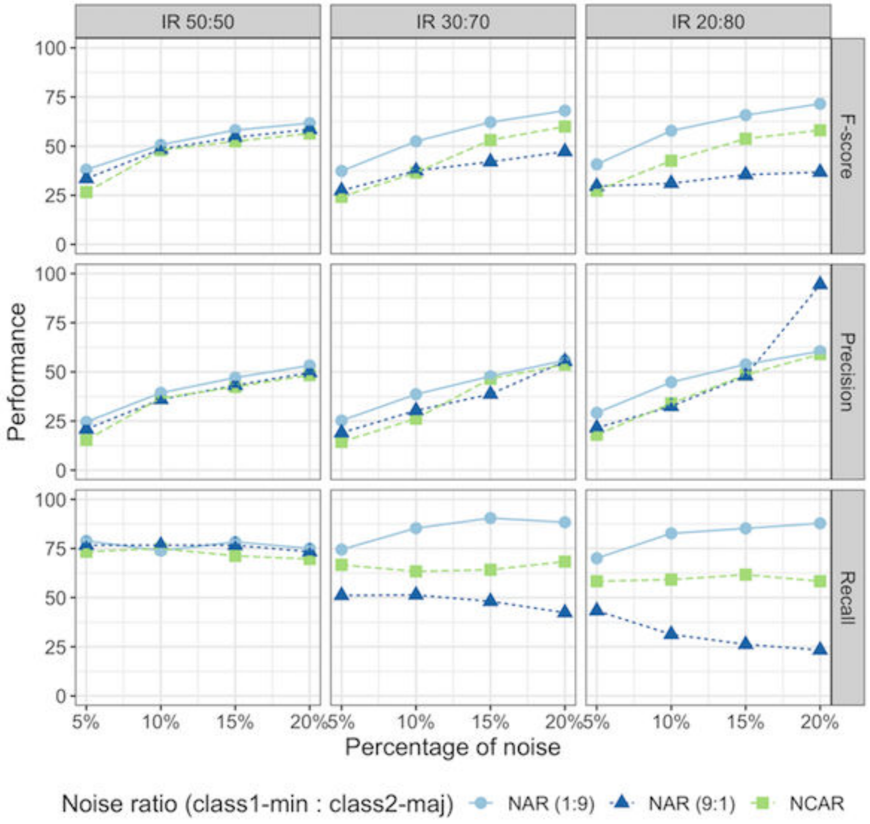

Figure 52 – Noise detection performance for majority vote on *arcene* dataset.

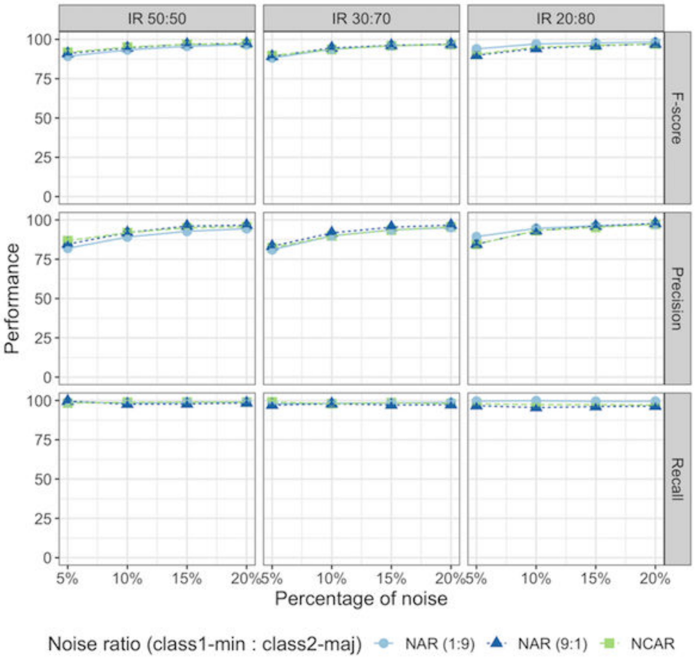

Figure 53 – Noise detection performance for majority vote on *breast-cancer-wisconsin* dataset.

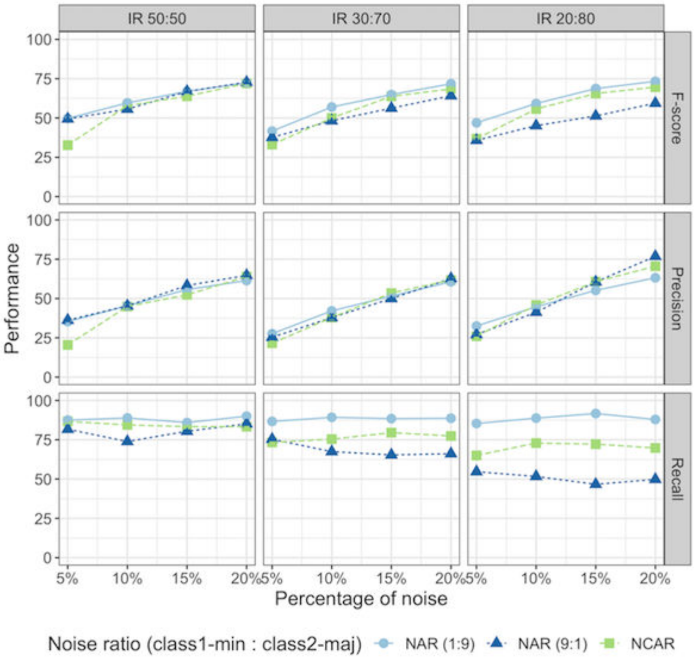

Figure 54 – Noise detection performance for majority vote on *column2C* dataset.

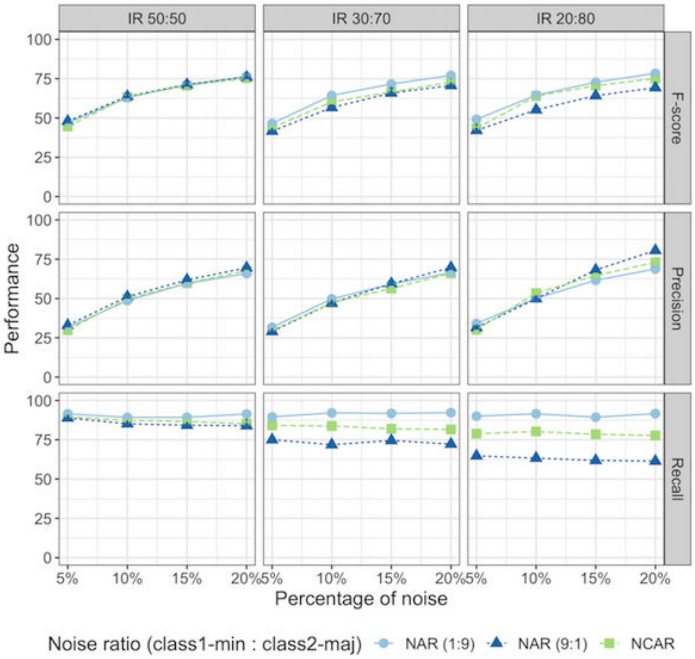

Figure 55 – Noise detection performance for majority vote on *credit* dataset.

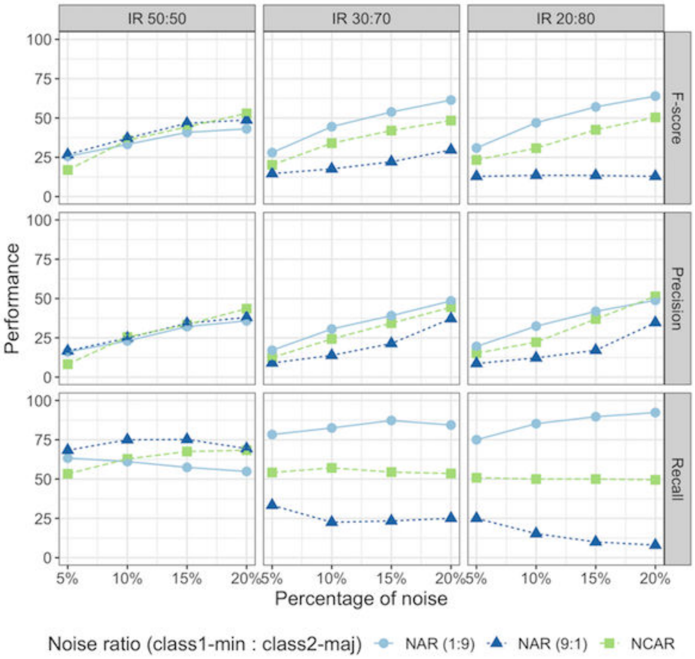

Figure 56 – Noise detection performance for majority vote on *cylinder-bands* dataset.

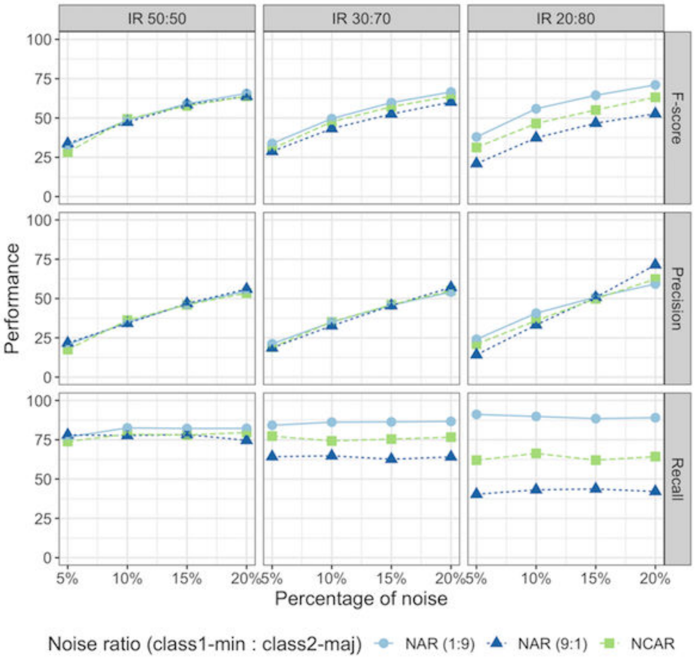

Figure 57 – Noise detection performance for majority vote on *diabetes* dataset.

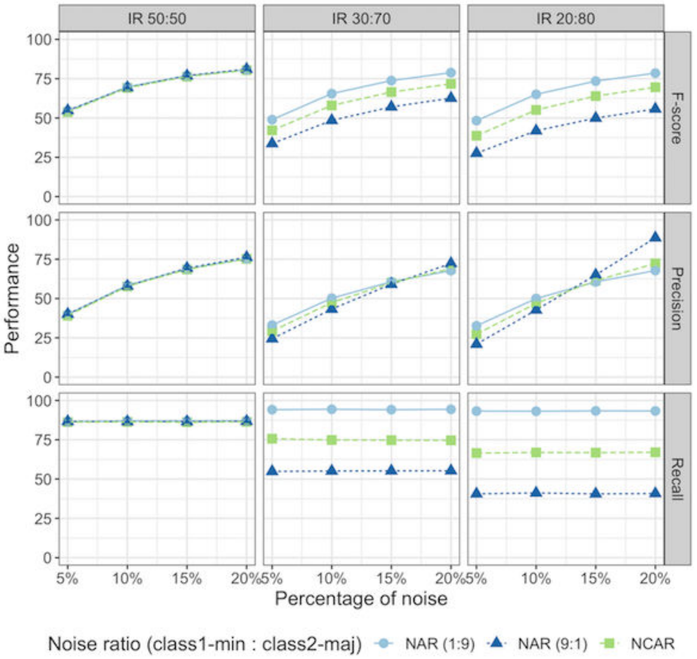

Figure 58 – Noise detection performance for majority vote on *eeg-eye-state* dataset.

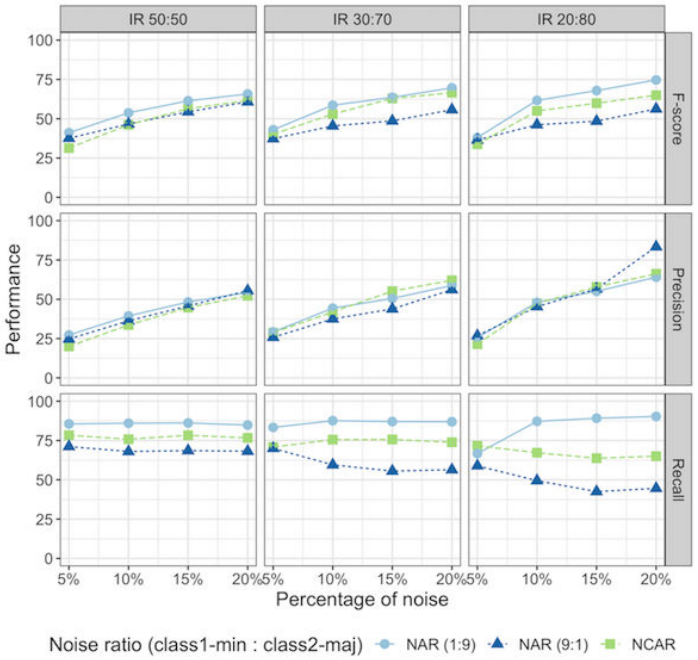

Figure 59 – Noise detection performance for majority vote on *glass0* dataset.

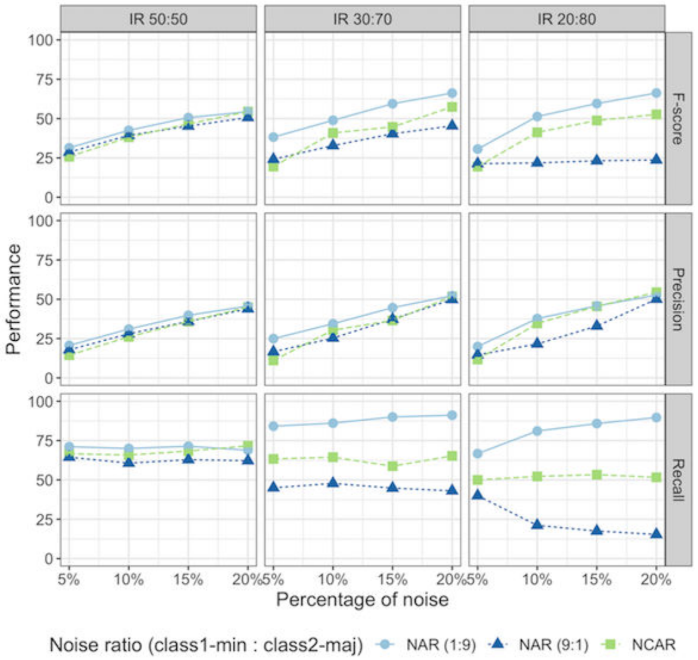

Figure 60 – Noise detection performance for majority vote on *glass1* dataset.

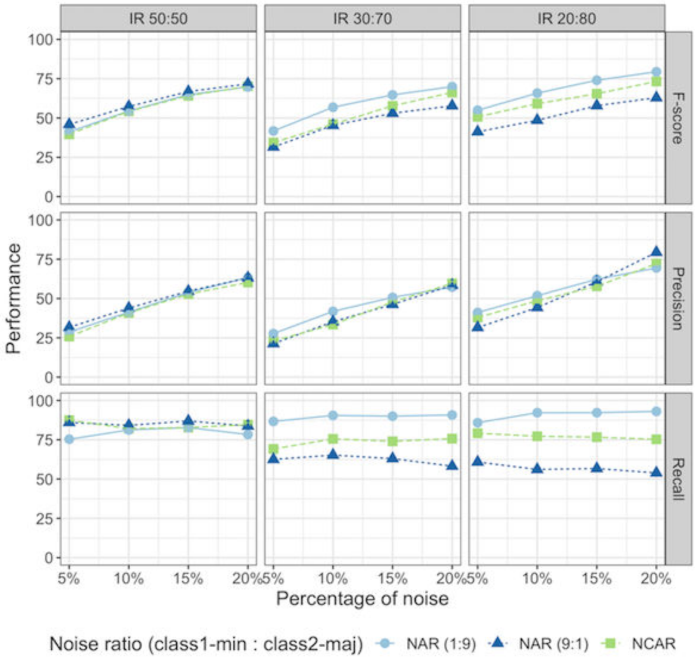

Figure 61 – Noise detection performance for majority vote on *heart-c* dataset.

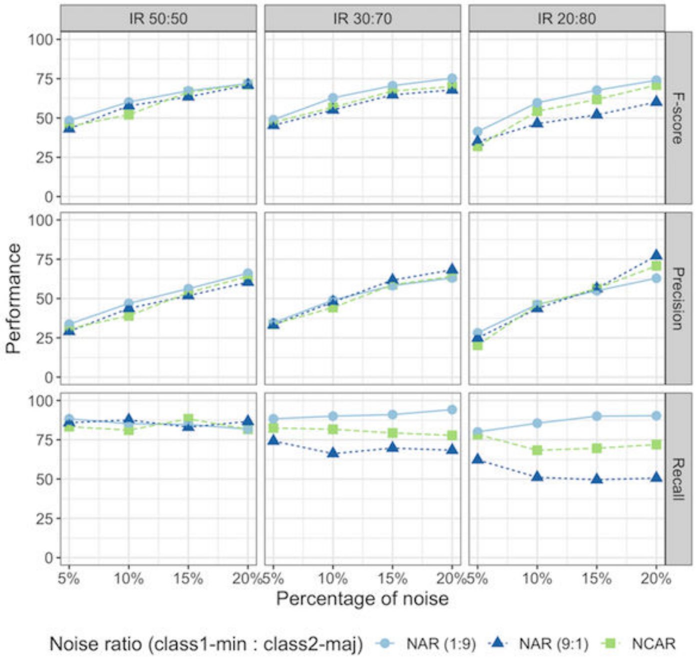

Figure 62 – Noise detection performance for majority vote on *heart-statlog* dataset.

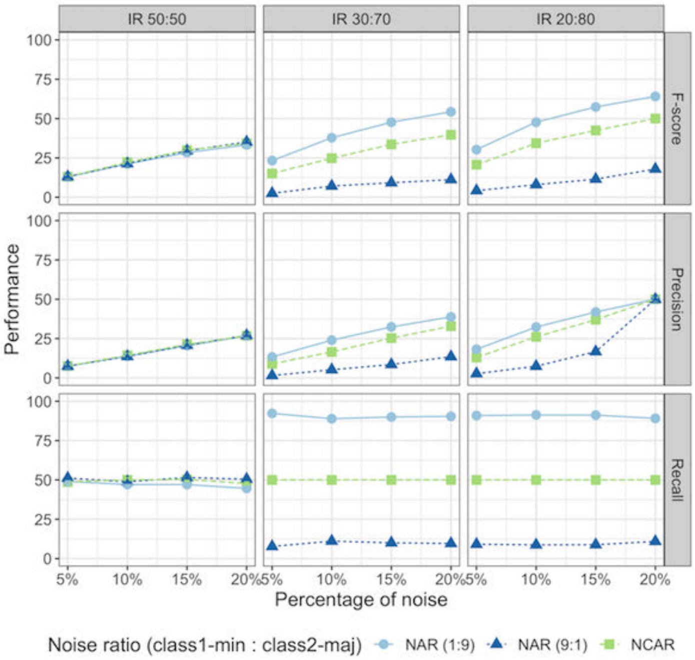

Figure 63 – Noise detection performance for majority vote on *hill-valley* dataset.

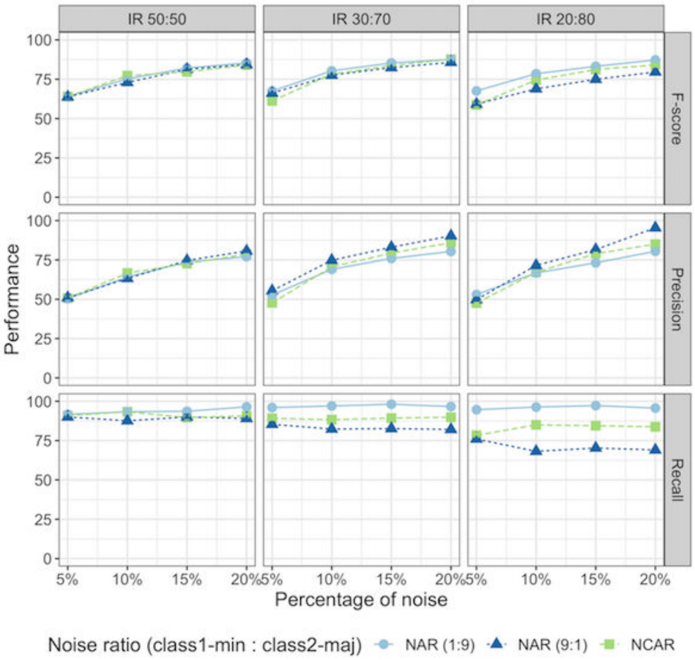

Figure 64 – Noise detection performance for majority vote on *ionosphere* dataset.

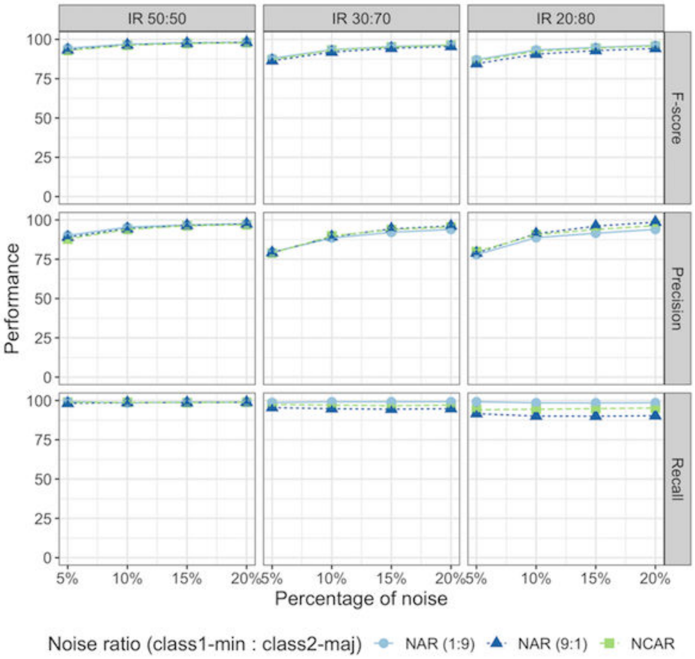

Figure 65 – Noise detection performance for majority vote on *kr-vs-kp* dataset.

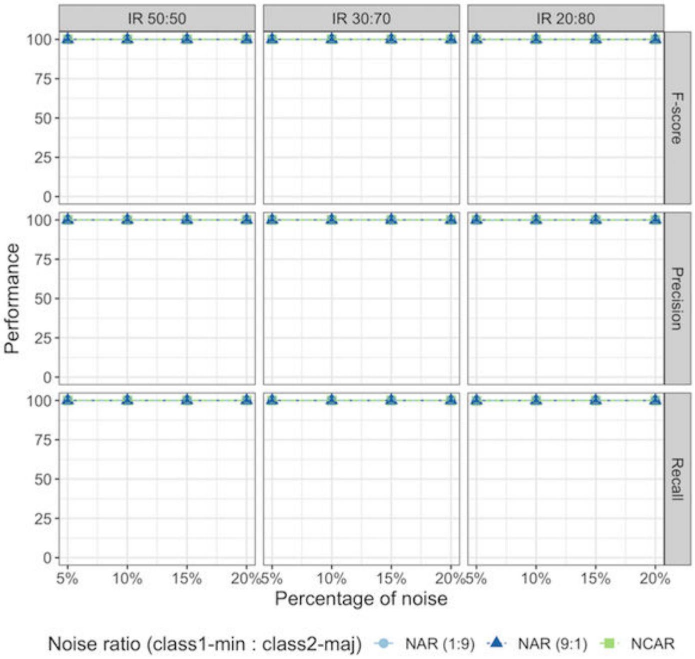

Figure 66 – Noise detection performance for majority vote on *mushroom* dataset.

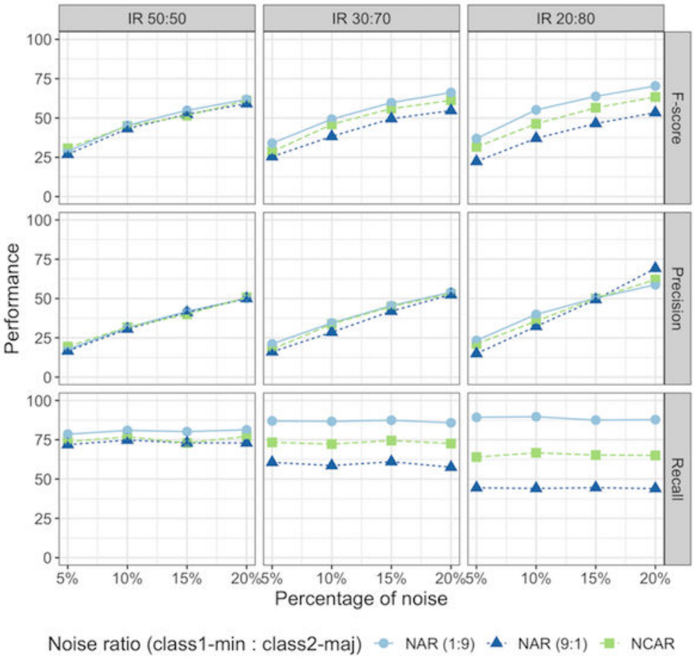

Figure 67 – Noise detection performance for majority vote on *pima* dataset.

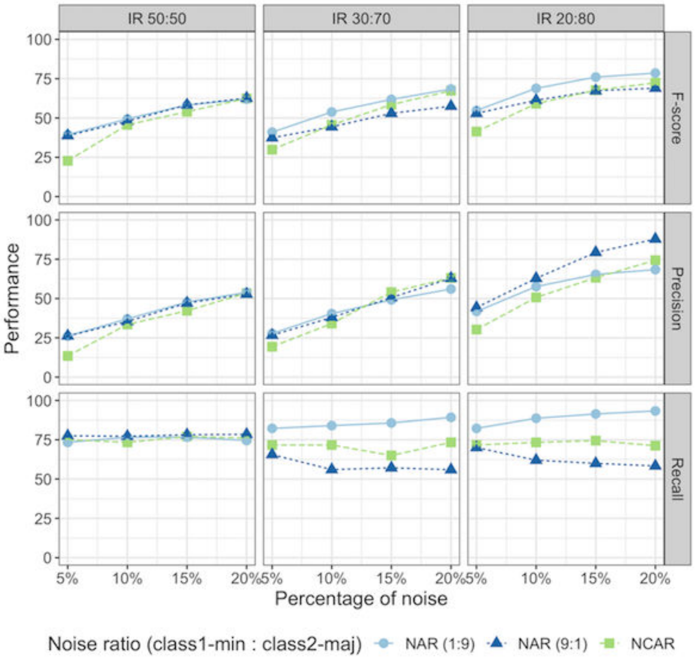

Figure 68 – Noise detection performance for majority vote on *sonar* dataset.

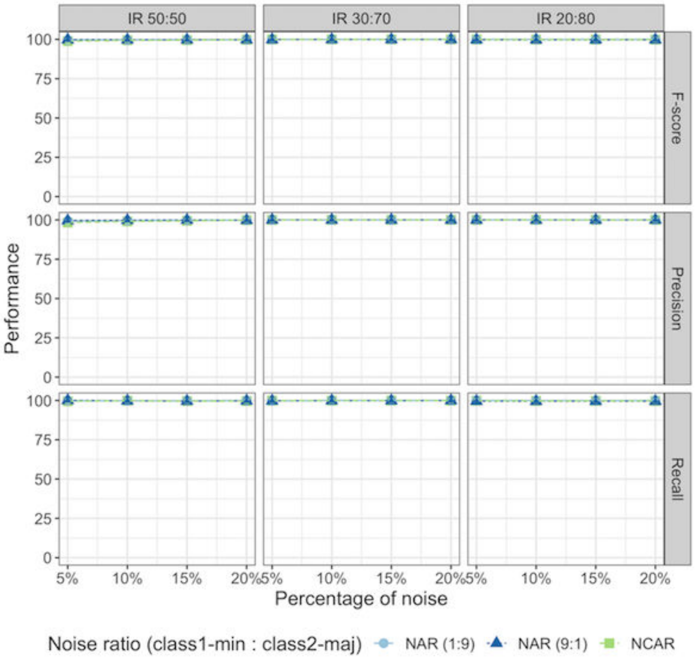

Figure 69 – Noise detection performance for majority vote on *steel-plates-fault* dataset.

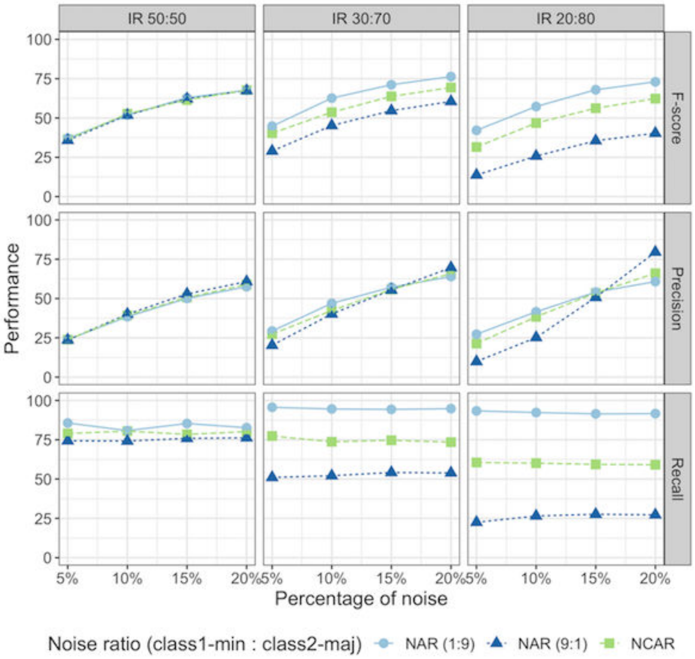

Figure 70 – Noise detection performance for majority vote on *tic-tac-toe* dataset.

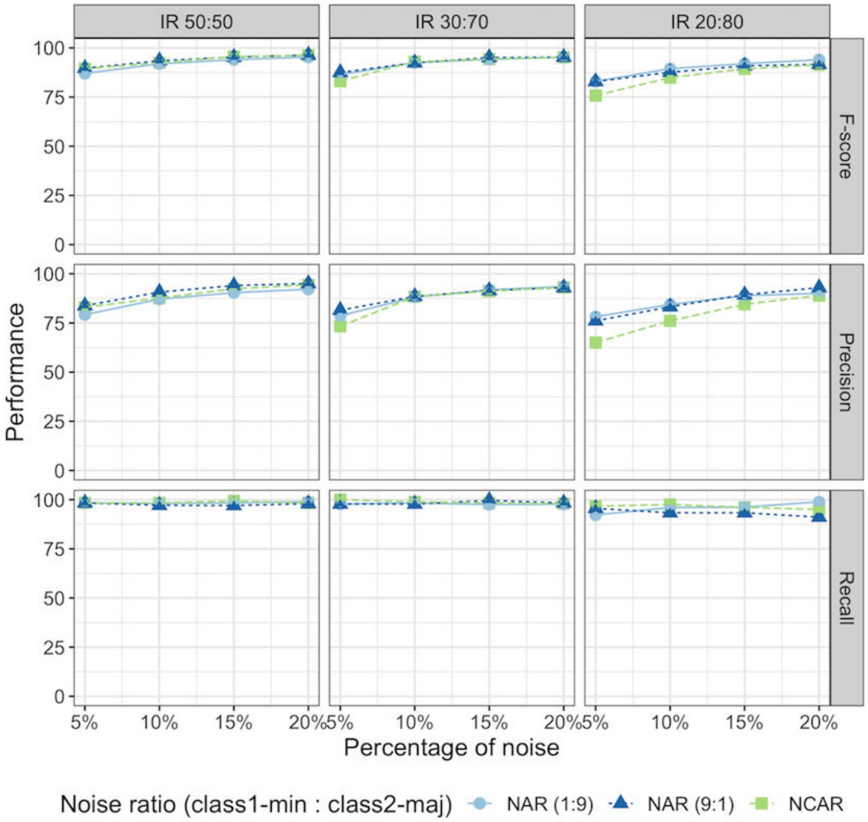

Figure 71 – Noise detection performance for majority vote on *voting* dataset.

B.2 ENSEMBLE VOTE THRESHOLD

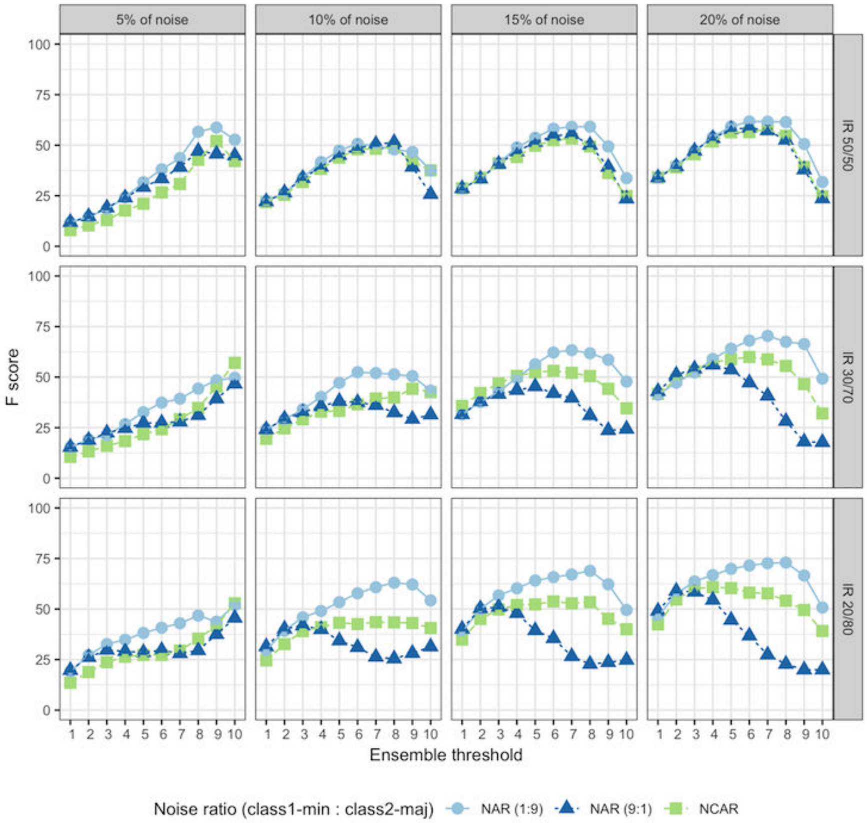

Figure 72 – *F-score* on *arcene* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

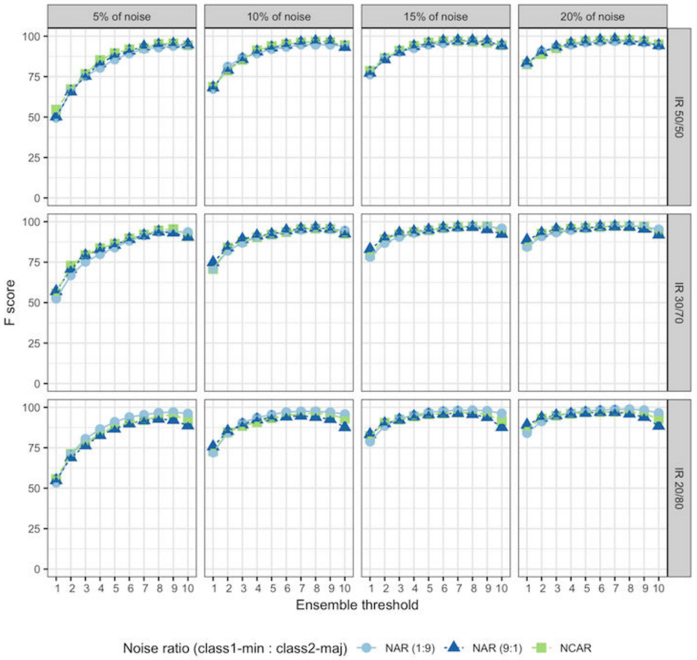

Figure 73 – *F-score* on *breast-cancer-wisconsin* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

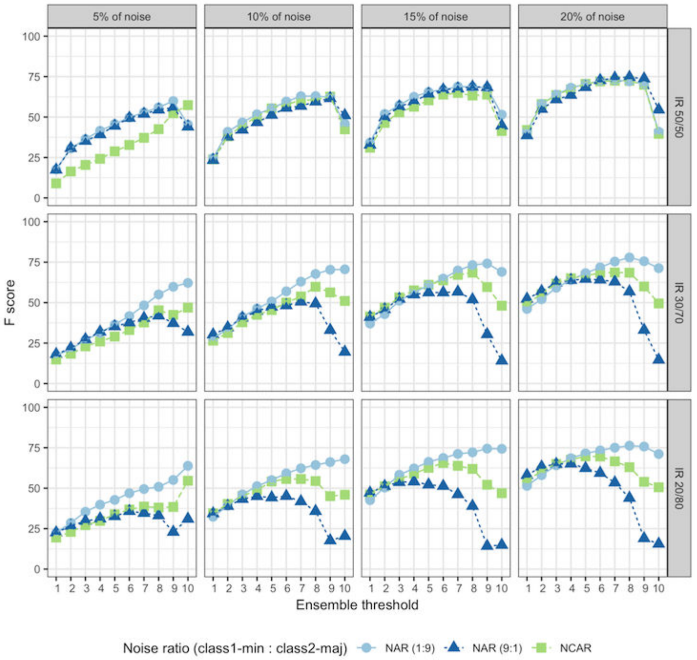

Figure 74 – *F-score* on *column2C* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

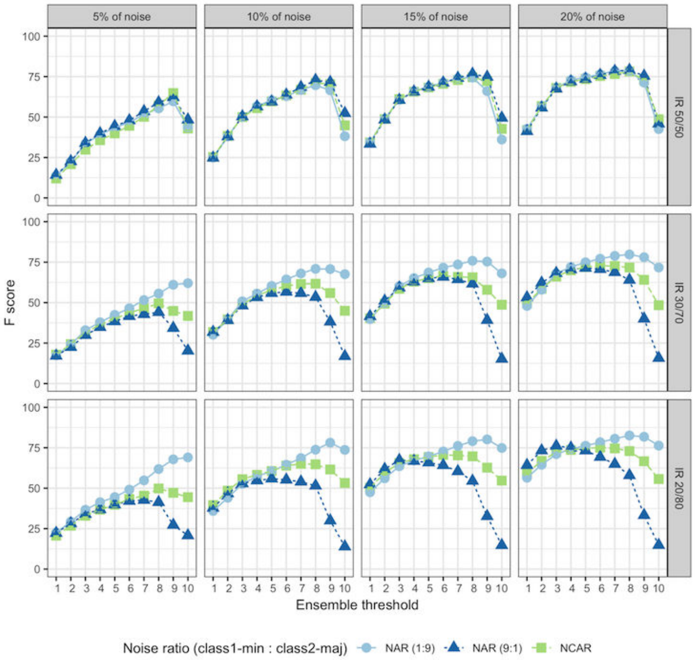

Figure 75 – *F-score* on *credit* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

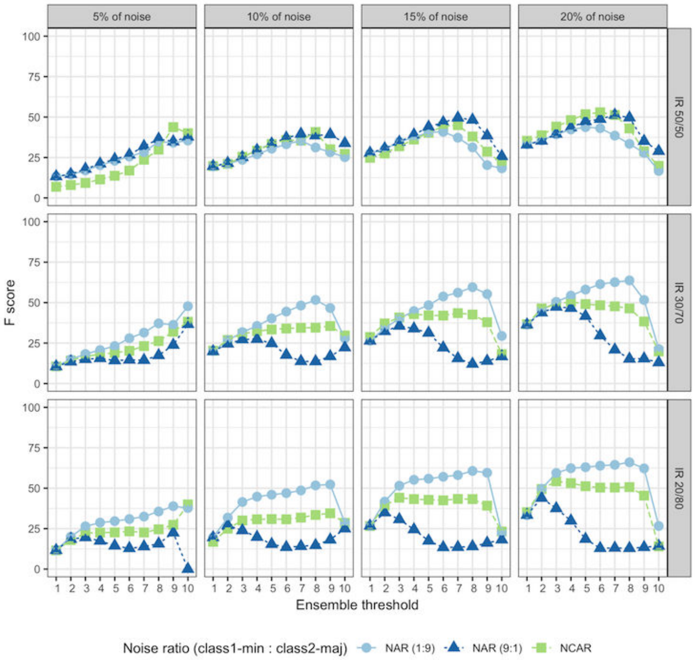

Figure 76 – *F-score* on *cylinder-bands* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

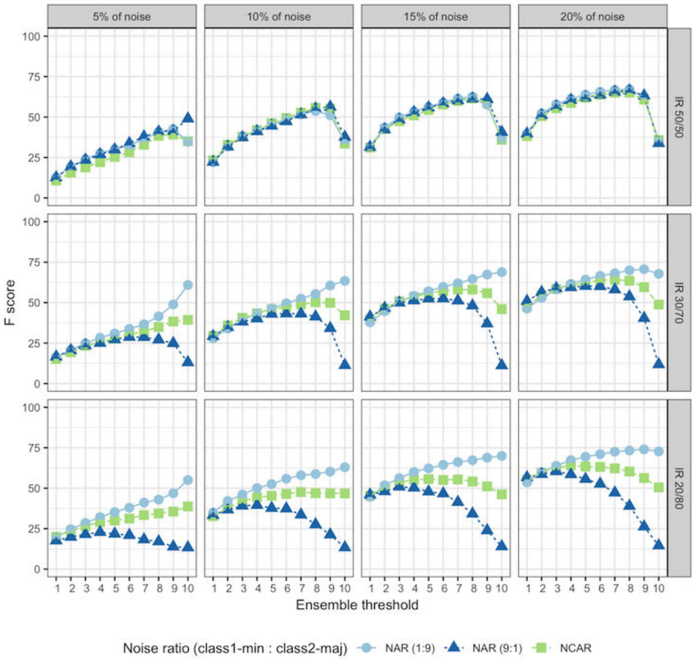

Figure 77 – *F-score* on *diabetes* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

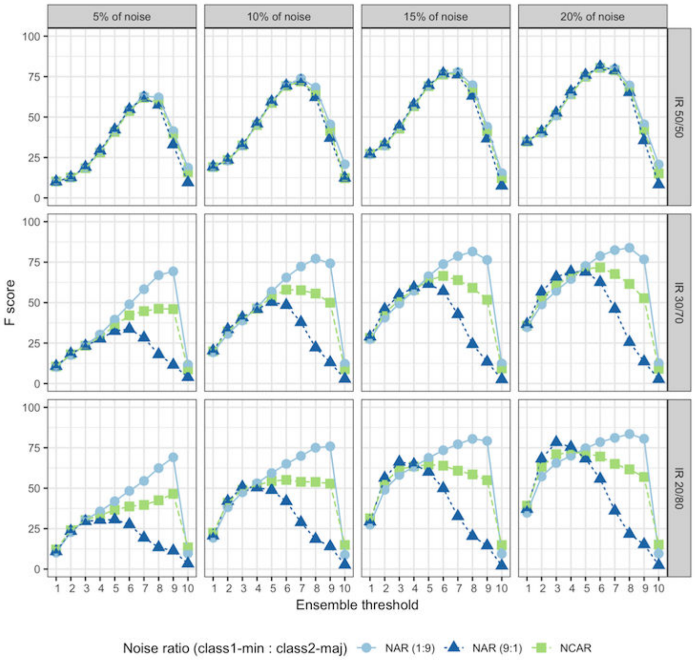

Figure 78 – *F-score* on *eeg-eye-state* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

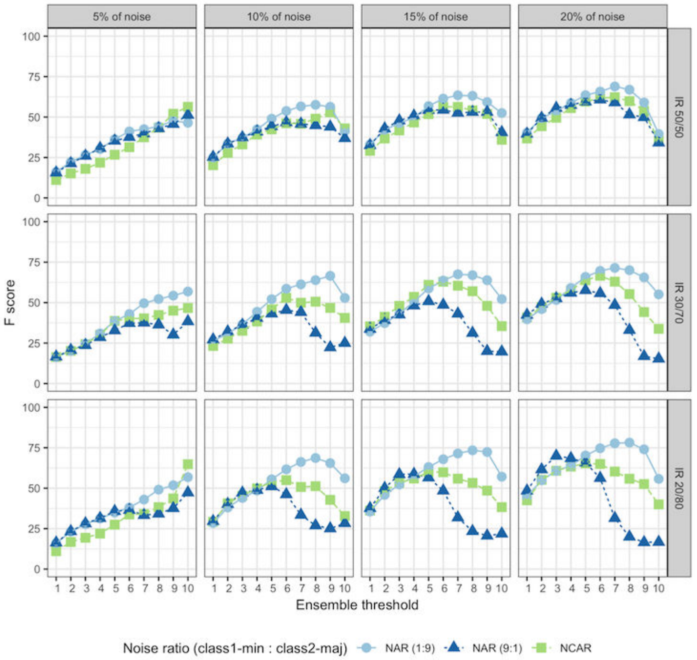

Figure 79 – *F-score* on *glass0* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

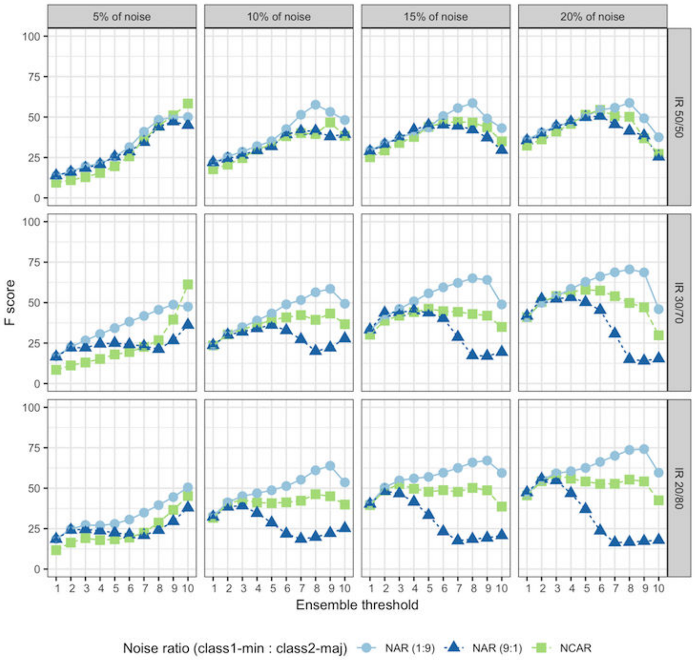

Figure 80 – *F-score* on *glass1* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

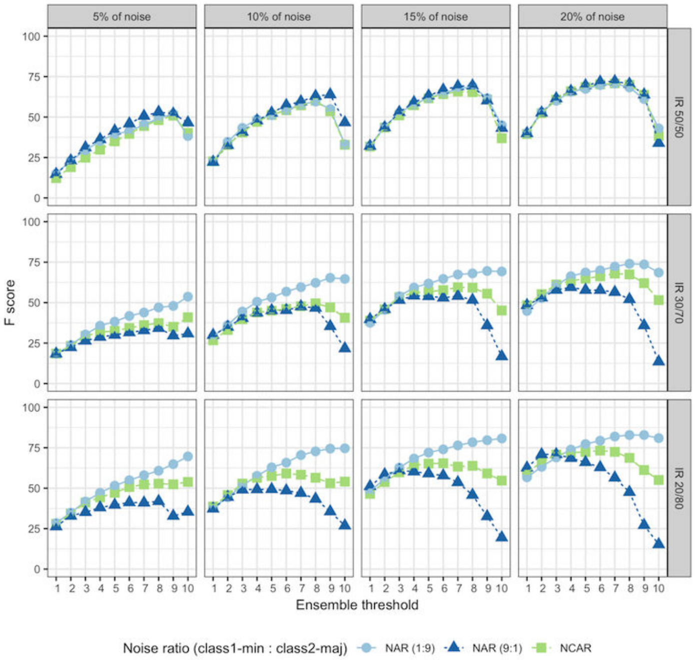

Figure 81 – *F-score* on *heart-c* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

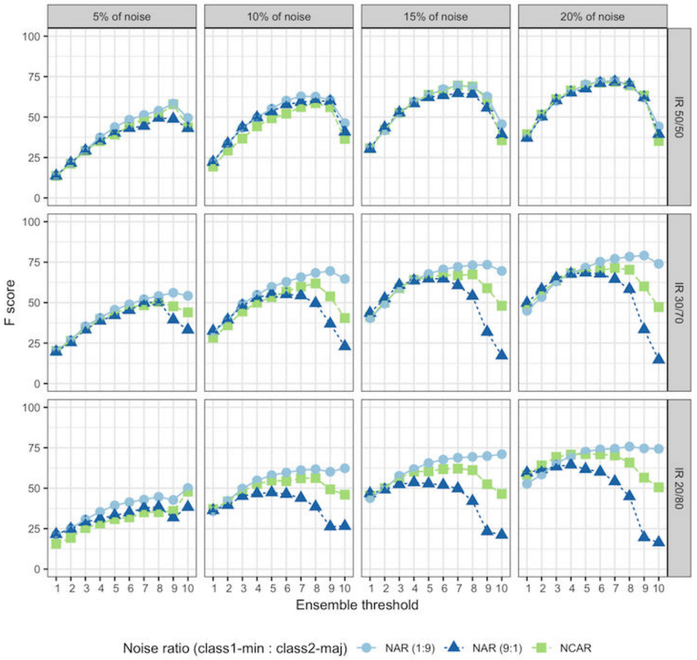

Figure 82 – *F-score* on *heart-statlog* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

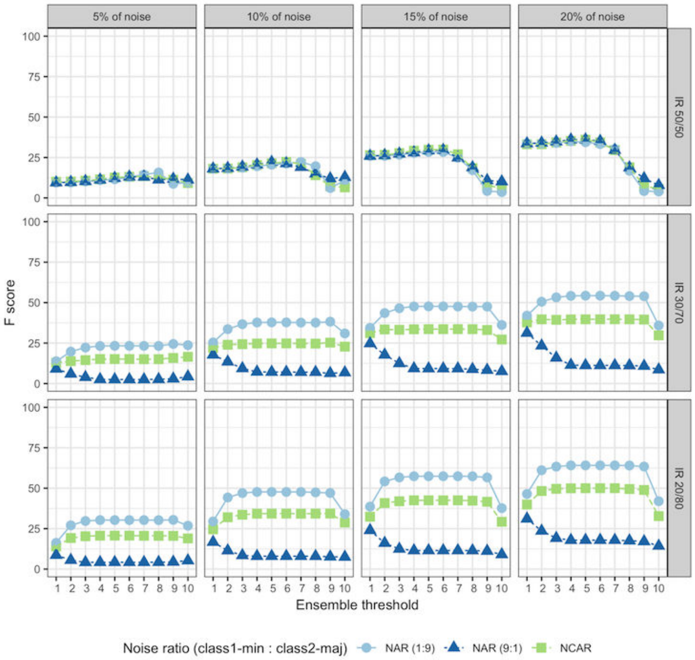

Figure 83 – *F-score* on *hill-valley* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

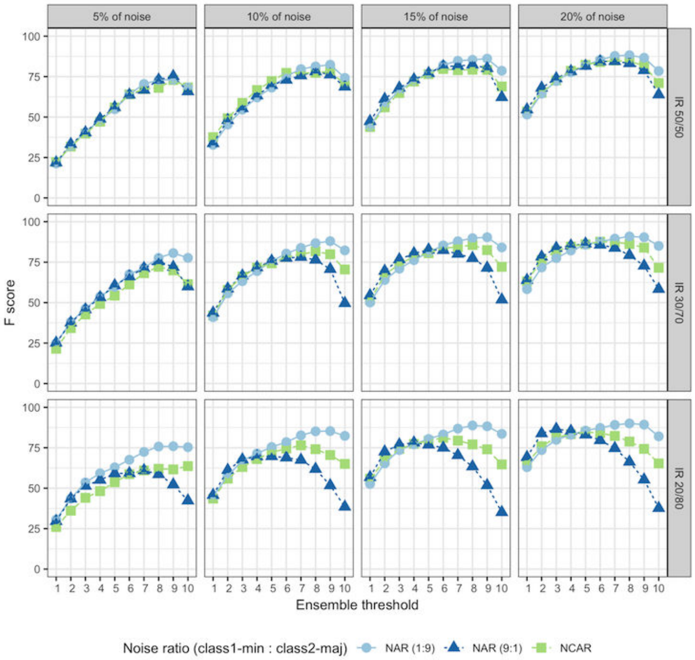

Figure 84 – *F-score* on *ionosphere* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

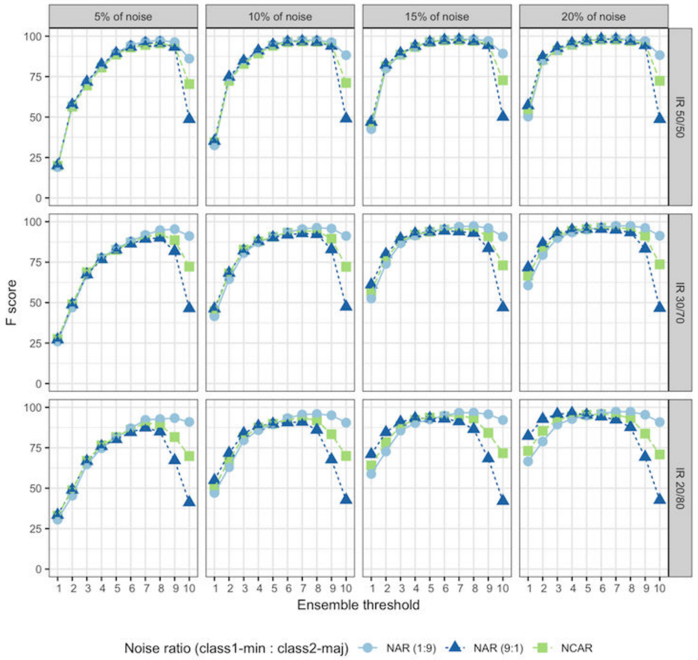

Figure 85 – *F-score* on *kr-vs-kp* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

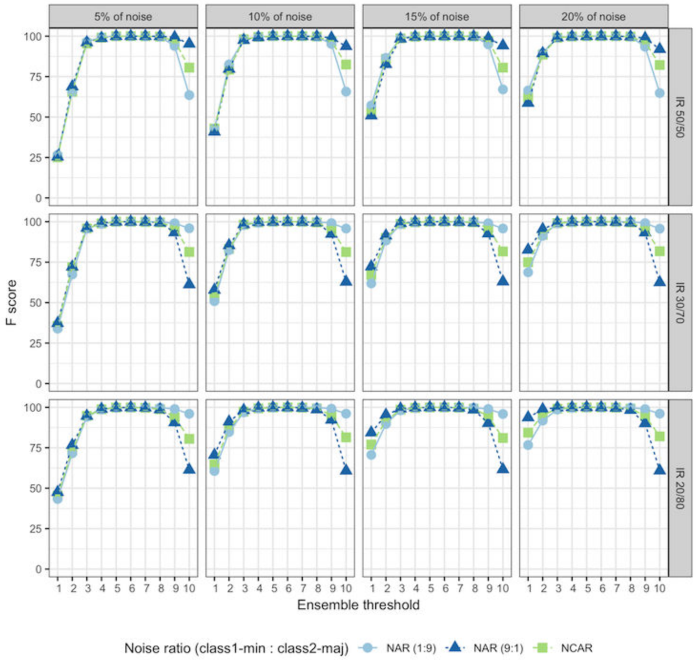

Figure 86 – *F-score* on *mushroom* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

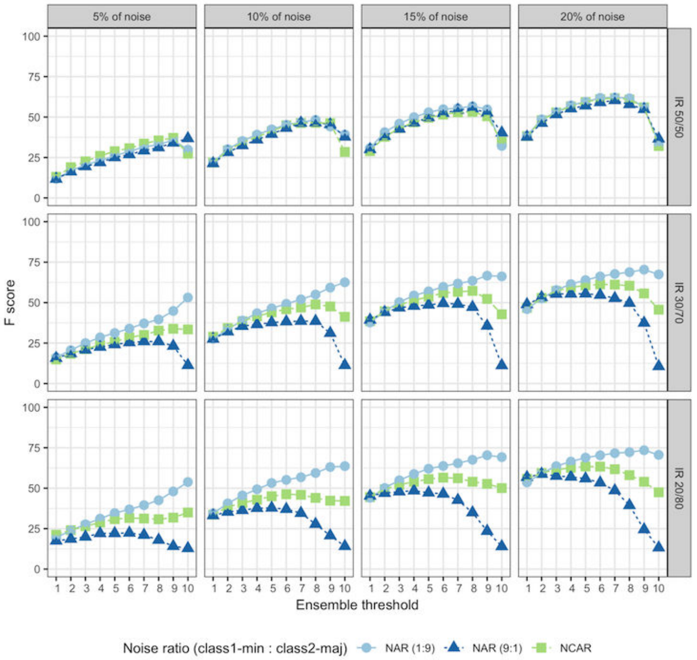

Figure 87 – *F-score* on *pima* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

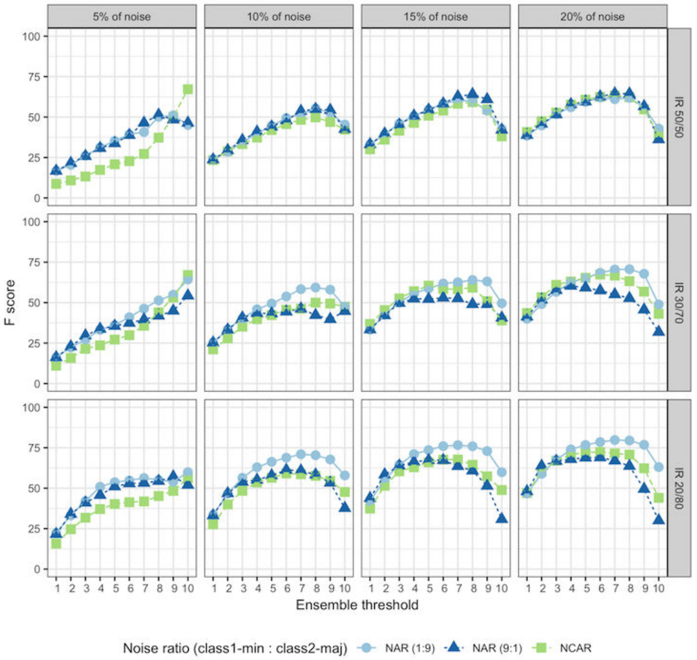

Figure 88 – *F-score* on *sonar* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

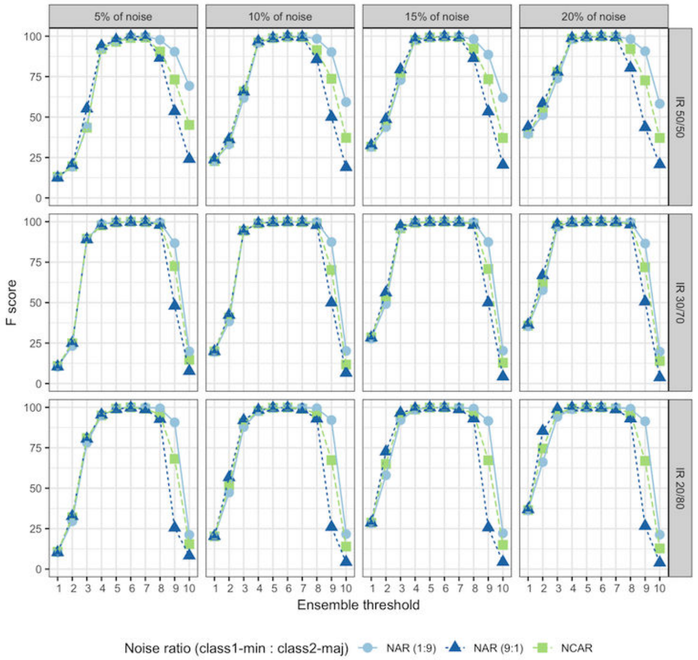

Figure 89 – *F-score* on *steel-plates-fault* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

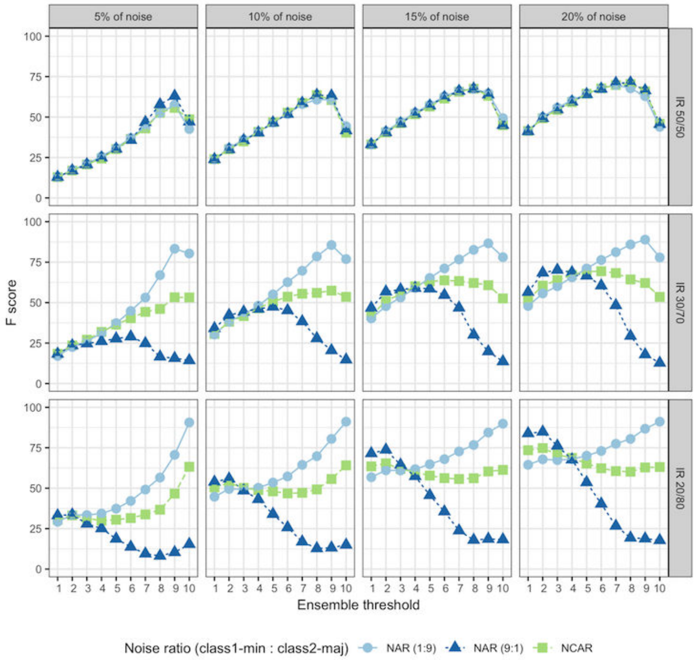

Figure 90 – *F-score* on *tic-tac-toe* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

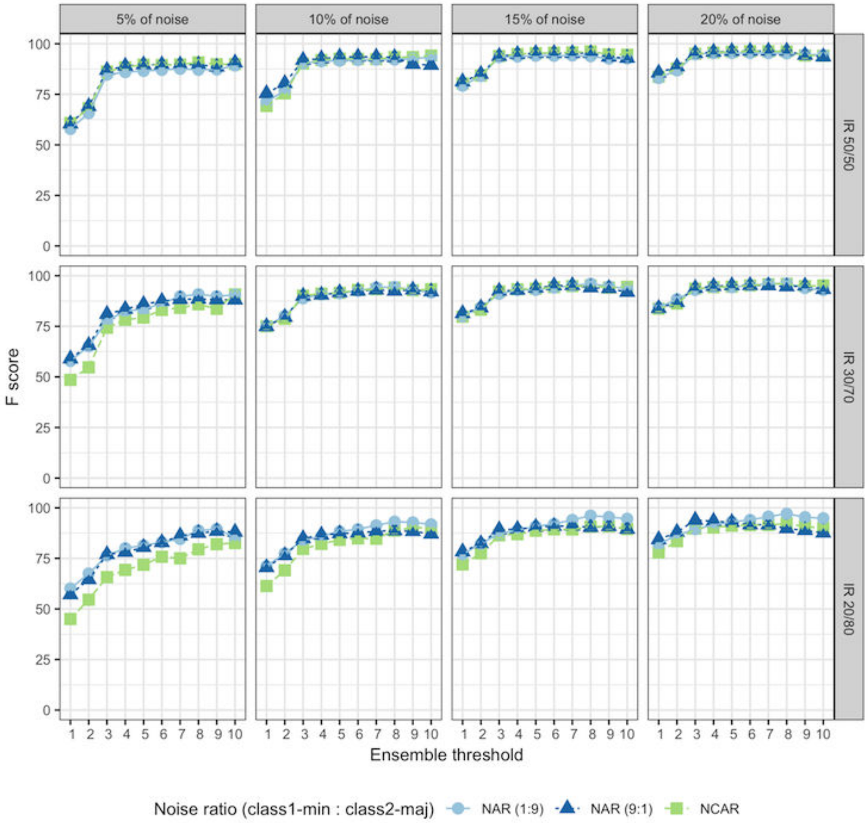

Figure 91 – *F-score* on *voting* problem under different ensemble vote *thresholds* (where 1 = 10%, 2 = 20%,..,10 = 100%).

B.3 STATISTICAL TESTS

Table 21 – Friedman test results of each problem. Non-significant differences ($\alpha > 0.05$) are marked with *. Part I.

Datasets	**IR**	**Friendman test result** **p-value**			
		5%	10%	15%	20%
arcene	50:50	0.0220	0.8700*	0.0920*	0.5840*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
breast-cancer-wisconsin	50:50	0.5900*	0.4090*	0.1110*	0.5470*
	30:70	0.6340*	0.2720*	0.9910*	0.7730*
	20:80	0.0390	0.0020	0.0040	0.0030
column2C	50:50	0.0000	0.5500*	0.5760*	1.0000*
	30:70	0.0100	0.0120	0.0000	0.0000
	20:80	0.0020	0.0000	0.0000	0.0000
credit	50:50	0.3580*	0.7290*	0.4580*	0.9670*
	30:70	0.0520*	0.0000	0.0000	0.0000
	20:80	0.0020	0.0000	0.0010	0.0000
cylinder-bands	50:50	0.0010	0.5250*	0.0140	0.1720*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
diabetes	50:50	0.1540*	0.5150*	0.5310*	0.9430*
	30:70	0.0020	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
eeg-eye-state	50:50	0.0410	0.5690*	0.0480	0.2330*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
glass0	50:50	0.0020	0.0660*	0.0390	0.0340
	30:70	0.0710*	0.0010	0.0000	0.0000
	20:80	0.4340*	0.0000	0.0000	0.0000
glass1	50:50	0.2400*	0.2080*	0.0640*	0.1990*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000

Table 23 – Friedman test results of each problem. Non-significantly difference ($\alpha > 0.05$) are marked with *. Part II.

Datasets	**IR**	**Friendman test result p-value**			
		5%	10%	15%	20%
heart-c	50:50	0.1030*	0.7690*	0.6530*	0.3810*
	30:70	0.0060	0.0000	0.0000	0.0000
	20:80	0.0160	0.0000	0.0000	0.0000
heart-statlog	50:50	0.7770*	0.0260	0.2180*	0.6900*
	30:70	0.1890*	0.0250	0.1010*	0.0250
	20:80	0.0360	0.0000	0.0000	0.0000
hill-valley	50:50	0.9430*	0.8750*	0.6480*	0.5870*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
ionosphere	50:50	0.6390*	0.0960*	0.3740*	0.6550*
	30:70	0.3880*	0.5640*	0.2420*	0.2540*
	20:80	0.0080	0.0010	0.0030	0.0030
kr-vs-kp	50:50	0.1230*	0.5560*	0.3330*	0.3190*
	30:70	0.8970*	0.1500*	0.2880*	0.1010*
	20:80	0.1830*	0.0370	0.0470	0.0000
pima	50:50	0.4000*	0.9920*	0.0090	0.5390*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
sonar	50:50	0.0000	0.6010*	0.0390	0.9750*
	30:70	0.0010	0.0030	0.0520*	0.0020
	20:80	0.0000	0.0550*	0.0040	0.0030
steel-plates-fault	50:50	0.6250*	0.6400*	0.7590*	0.1380*
	30:70	0.8190*	0.3680*	0.0740*	0.4170*
	20:80	0.1740*	0.0860*	0.0470	0.0000
tic-tac-toe	50:50	0.7920*	0.5060*	0.5990*	0.9270*
	30:70	0.0000	0.0000	0.0000	0.0000
	20:80	0.0000	0.0000	0.0000	0.0000
voting	50:50	0.2770*	0.5590*	0.0370	0.6150*
	30:70	0.1300*	1.0000*	0.3020*	0.8730*
	20:80	0.1070*	0.1080*	0.2180*	0.1740*

Table 25 – Wilcoxon test on real problems when there is 5% of noise in data. W \T \L = wins\ties\losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	6/0/13	10/0/9	16/0/3	11/0/8	16/0/3	17/0/2	13/0/6	15/0/4
		p-value	**0.019**	0.433	**0.002**	0.153	**0.001**	**0.002**	**0.038**	**0.004**
	NCAR	**W/T/L**		14/1/4	13/0/6	10/0/9	15/0/4	14/0/5	9/0/10	13/0/6
		p-value		**0.012**	**0.013**	0.615	**0.004**	**0.023**	0.763	**0.021**
	NAR(1:9)	**W/T/L**			7/0/12	4/0/15	7/0/12	5/0/14	4/0/15	7/0/12
		p-value			0.305	**0.009**	0.268	0.061	**0.009**	0.205
30:70	**NAR(9:1)**	**W/T/L**				13/0/6	16/1/2	13/0/6	13/0/6	17/0/2
		p-value				0.433	**0.000**	0.103	0.121	**0.001**
	NCAR	**W/T/L**					17/0/2	8/0/11	10/0/9	15/0/4
		p-value					**0.000**	0.457	0.952	**0.002**
	NAR(1:9)	**W/T/L**						2/0/17	3/1/15	7/0/12
		p-value						**0.001**	**0.002**	0.286
20:80	**NAR(9:1)**	**W/T/L**							12/0/7	19/0/0
		p-value							0.220	**0.000**
	NCAR	**W/T/L**								19/0/0
		p-value								**0.000**

Table 26 – Wilcoxon test on real problems when there is 10% of noise in data. W \T \L = wins\ties\losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	7/0/12	12/0/7	17/0/2	16/0/3	15/0/4	18/0/1	16/0/3	16/0/3
		p-value	0.615	0.165	**0.001**	**0.001**	**0.001**	**0.001**	**0.002**	**0.001**
	NCAR	**W/T/L**		11/0/8	11/0/8	10/0/9	11/0/8	8/0/11	9/0/10	11/0/8
		p-value		0.433	0.268	0.324	0.073	0.856	0.763	0.268
	NAR(1:9)	**W/T/L**			5/0/14	4/0/15	3/0/16	3/0/16	3/0/16	5/0/14
		p-value			**0.007**	**0.007**	**0.009**	**0.003**	**0.003**	**0.006**
30:70	**NAR(9:1)**	**W/T/L**				16/1/2	18/0/1	15/0/4	16/0/3	17/0/2
		p-value				**0.001**	**0.000**	**0.015**	**0.002**	**0.000**
	NCAR	**W/T/L**					18/0/1	4/0/15	9/0/10	16/0/3
		p-value					**0.000**	**0.004**	0.560	**0.004**
	NAR(1:9)	**W/T/L**						1/0/18	1/0/18	7/0/12
		p-value						**0.000**	**0.000**	0.056
20:80	**NAR(9:1)**	**W/T/L**							17/0/2	19/0/0
		p-value							**0.001**	**0.000**
	NCAR	**W/T/L**								19/0/0
		p-value								**0.000**

Table 27 – Wilcoxon test on real problems when there is 20% of noise in data. W \T \L = wins\ties\losses. p-value < 0.05 are highlighted.

IR	**Noise Model**		**50:50**		**30:70**			**20:80**		
			NCAR	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**	**NAR (9:1)**	**NCAR**	**NAR (1:9)**
50:50	**NAR(9:1)**	**W/T/L**	6/0/13	11/0/8	17/0/2	17/0/2	16/0/3	18/0/1	17/0/2	17/0/2
		p-value	0.587	0.533	**0.000**	**0.000**	**0.001**	**0.000**	**0.001**	**0.001**
	NCAR	**W/T/L**		11/0/8	9/0/10	9/0/10	10/0/9	11/0/8	11/0/8	13/0/6
		p-value		0.305	0.702	0.888	0.794	0.920	0.658	0.387
	NAR(1:9)	**W/T/L**			6/0/13	6/0/13	4/0/15	3/0/16	3/0/16	3/0/16
		p-value			**0.009**	**0.005**	**0.005**	**0.002**	**0.001**	**0.002**
30:70	**NAR(9:1)**	**W/T/L**				16/0/3	17/0/2	14/0/5	17/0/2	17/0/2
		p-value				**0.000**	**0.000**	**0.028**	**0.000**	**0.000**
	NCAR	**W/T/L**					18/0/1	3/0/16	10/0/9	16/0/3
		p-value					**0.000**	**0.001**	0.984	**0.003**
	NAR(1:9)	**W/T/L**						1/0/18	3/0/16	6/0/13
		p-value						**0.000**	**0.001**	**0.021**
20:80	**NAR(9:1)**	**W/T/L**							18/0/1	19/0/0
		p-value							**0.000**	**0.000**
	NCAR	**W/T/L**								19/0/0
		p-value								**0.000**