ตัวแบบจำแนกประเภทเพื่อนบ้านใกล้สุดแบบพลวัตโดยใช้ค่าปัจจัยผิดปกติแมส-เรโช-แวเรียนซ์ สำหรับปัญหาคลาสไม่ดุล

นางสาวพัชรสิริ เฟื่องฟู

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2566

DYNAMIC NEAREST NEIGHBOR CLASSIFIER USING MASS-RATIO-VARIANCE OUTLIER FACTORS FOR CLASS IMBALANCE PROBLEM

Ms. Patcharasiri Fuangfoo

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Applied Mathematics and
Computational Science
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2023

Thesis Title	DYNAMIC NEAREST NEIGHBOR CLASSIFIER USING MASS-RATIO-VARIANCE OUTLIER FACTORS FOR CLASS IMBALANCE PROBLEM
By	Ms. Patcharasiri Fuangfoo
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Associate Professor Krung Sinapiromsaran, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

. . . . Dean of the Faculty of Science
(Professor Pranut Potiyaraj, Ph.D.)

THESIS COMMITTEE

. . . . เพชรอาภา บุญเสริม . . . . Chairman
(Associate Professor Petarpa Boonserm, Ph.D.)

. . . . Thesis Advisor
(Associate Professor Krung Sinapiromsaran, Ph.D.)

. . . . กิติพร พลายมาศ . . . . Examiner
(Associate Professor Kitiporn Plaimas, Ph.D.)

. . . . External Examiner
(Associate Professor Chumphol Bunkhumpornpat, Ph.D.)

พัชรสิริ เฟื่องฟู : ตัวแบบจำแนกประเภทเพื่อนบ้านใกล้สุดแบบพลวัตโดยใช้ค่าปัจจัยผิดปกติแมส-เรโช-แวเรียนซ์ สำหรับปัญหาคลาสไม่ดุล. (DYNAMIC NEAREST NEIGHBOR CLASSIFIER USING MASS-RATIO-VARIANCE OUTLIER FACTORS FOR CLASS IMBALANCE PROBLEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร. กรุง สินอภิรมย์สราญ, 170 หน้า.

วัตถุประสงค์ของการจำแนกประเภทคือการกำหนดคลาสให้กับข้อมูลให้แม่นยำผ่านตัวจำแนกประเภท หนึ่งในตัวจำแนกประเภทที่เป็นที่รู้จักคือ ตัวแบบเพื่อนบ้านใกล้สุดเค ซึ่งคลาสของข้อมูลจะถูกกำหนดโดยการพิจารณาคลาสส่วนใหญ่จากกลุ่มตัวอย่าง เค ที่อยู่ใกล้ที่สุด อย่างไรก็ตาม ประสิทธิภาพของตัวแบบเพื่อนบ้านใกล้สุดเคจะลดลงเมื่อชุดข้อมูลที่คลาสไม่ได้ดุล

เพื่อแก้ไขปัญหานี้ ตัวแบบควรใช้ค่าเคที่ต่างกันสำหรับข้อมูลแต่ละตัว ตามตำแหน่งของตัวอย่างที่อยู่ในคลัสเตอร์หรือแยกตัวออกมา โดยใช้คะแนนที่คำนวณจากความหนาแน่นเรียก แมส-เรโช-แวเรียนซ์ เอาท์ไลเออร์ แฟคเตอร์ (เอ็มโอเอฟ) ที่ไร้พารามิเตอร์ เข้ากับกระบวนการเพื่อนบ้านใกล้สุดเค ช่วยในการกำหนดเพื่อนบ้านที่เหมาะสม

งานวิจัยของเราเน้นการพัฒนาตัวจำแนกประเภทเพื่อนบ้านที่ไดนามิกที่สุดเพื่อแก้ไขปัญหาความไม่สมดุลของคลาส ผลการทดลองกับข้อมูลจริง 10 ชุดแสดงให้เห็นว่าตัวจำแนกประเภทที่นำเสนอสามารถทำนายผลได้อย่างแม่นยำ ซึ่งมีความใกล้เคียงกับตัวแบบเพื่อนบ้านใกล้สุดเคโดยใช้พารามิเตอร์ เค ที่ดีที่สุด

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต พัชรสิริ เฟื่องฟู

ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

สาขาวิชา คณิตศาสตร์ประยุกต์และวิทยาการคณนา ลายมือชื่อ อ.ที่ปรึกษาร่วม ....

ปีการศึกษา 2566

6470121923 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : K-NEAREST NEIGHBOR, DYNAMIC NEAREST NEIGHBOR, MASS-RATIO-VARIANCE, AND CLASS IMBALANCE PROBLEM

PATCHARASIRI FUANGFOO : DYNAMIC NEAREST NEIGHBOR CLASSIFIER USING MASS-RATIO-VARIANCE OUTLIER FACTORS FOR CLASS IMBALANCE PROBLEM. ADVISOR : ASSOC. PROF. KRUNG SINAPIROMSARAN, Ph.D., 170 pp.

The objective of classification is to assign a class to a given data instance. One well-recognized classifier is the k-NN model, where the class of an instance is determined by considering the majority class among its k nearest neighbors. However, k-NN's performance weakens in imbalanced datasets.

To address this, adjusting k for each instance based on factors like its position relative to clusters or isolation, and integrating density-based scores from a parameter-free Mass-ratio-variance Outlier Factor (MOF) into the k-NN process, helps determine suitable nearest neighbors.

Our research focuses on the development of a dynamic nearest neighbor classifier tailored specifically to address class imbalance problems. Experimental results on ten real-world datasets show our classifier accurately forecasts outcomes, aligning closely with traditional k-NN with the best parameter k.

Department : Mathematics and Computer Science
Student's Signature พัชรสิริ เฟื่องฟู
Advisor's Signature
Field of Study : Applied Mathematics and Computational Science
Co-advisor's Signature
Academic Year : 2023

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my research advisor, Associate Professor Krung Sinapiromsaran, Ph.D., for his invaluable guidance, encouragement, and support throughout the course of my studies. His unwavering dedication to my research has been instrumental in the successful completion of my thesis. I am also deeply grateful to my thesis committee members, Associate Professor Petarpa Boonserm, Ph.D., Associate Professor Kitiporn Plaimas,Ph.D. and Associate Professor Chumphol Bunkhumpornpat, Ph.D., for their insightful feedback and suggestions that helped me to refine my research.

Furthermore, I wish to acknowledge the generous financial support provided by the graduate school of Chulalongkorn University, which commemorated the 72^{nd} anniversary of his Majesty King Bhumibol Aduladej, and the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, which provided me with funding to present my research at international conferences and supported my graduate studies.

I am also grateful to my family for their unwavering support and encouragement throughout my studies. Finally, I extend my sincere thanks to my friends and colleagues for their invaluable support, encouragement, and helpful suggestions during my time as a graduate student.

CONTENTS

CHAPTER I

INTRODUCTION

This chapter delves into the significance of machine learning, the concept of classification, and class imbalance problems. It will also cover the nearest neighbor classifier and fundamental principles of density-based learning. Additionally, an overview of the thesis will be provided.

1.1 Machine learning, Classification, Class imbalance

In the contemporary world, where the generation and utilization of vast amounts of data, often referred to as “big data”, is prevalent, the essential task is to employ algorithms and computational procedures for effective data management. An algorithm can be defined as a sequence of instructions that must be executed to convert input into output. The primary objective is to enable computers or machines to perform various tasks, encompassing learning, problem-solving, prediction, pattern recognition, and the facilitation of robotics.

Machine learning, as described in [1], encompasses the process of instructing computers to enhance their performance by using either example data or prior experiences. Typically, this process entails the definition of a model with specific parameters, followed by the execution of a computer program to optimize those parameters using training data. Machine learning leverages statistical theory to construct mathematical models that aid in drawing inferences from a given dataset. Within this context, computer science assumes a dual role. Firstly, it is responsible for developing efficient algorithms to address optimization problems and for managing and storing large data volumes during the training phase. Secondly, it is concerned with designing computationally efficient representations and algorithmic solutions to extract insights from the learned model. In certain scenarios, the efficiency of the learning or inference algorithm, as gauged by its space and time complexity, can be as critical as its accuracy in predicting outcomes.

Machine learning is categorized into three primary domains, as outlined in [17]: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, the machine learning approach is applied to problems where the provided data includes labeled instances. Conversely, unsupervised learning is a machine learning paradigm employed to discern patterns from unlabeled data, with the aim of grouping similar instances or reducing the dimensionality of the input data. Lastly, reinforcement learning constitutes a distinct realm within machine learning, focused on the strategies intelligent agents should adopt in their environments to optimize the cumulative reward they accrue.

Classification entails the task of assigning a class label to an instance within a provided dataset through a classifier. This classifier is constructed from training data using a classification algorithm. The primary objective of a classifier is to deduce a model from the data, allowing it to assign a class label based on the characteristics of instances. Various known classifiers are nearest neighbor, decision tree, and support vector machine.

Presently, classification problems often grapple with the challenge of class imbalance, as highlighted in [5], [9], [10], , and this issue has garnered considerable attention due to its prevalence in real-world scenarios, such as fraud detection, anomaly detection, and medical diagnosis. In such situations, the minority class, which is one of the two classes, contains significantly fewer instances compared to the majority class. As depicted in Figure 1.1, the data illustrates an instance of data imbalance, featuring two distinct groups. The larger of the two groups is referred to as the “majority,” characterized by its blue color, while the smaller group is denoted as the “minority,” represented by the color red.

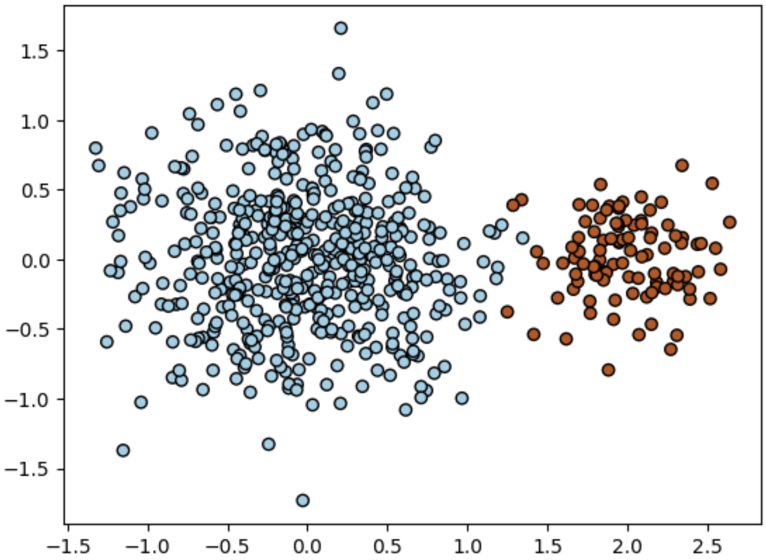


Figure 1.1: Example of imbalance data

Approaches to address class imbalance generally fall into one of three categories [11]: data sampling, algorithmic modification, and cost-sensitive methods. Sampling techniques aim to mitigate class imbalance by either oversampling the minority class instances or undersampling the majority class instances. Notably, a well-known oversampling technique is SMOTE [3], which enhances the representation of the minority class by generating synthetic examples along line segments connecting the minority class samples with their k nearest neighbors from the same class. In Figure 1.2, the sampling technique is illustrated. Figure 1.2(a) provides an instance of augmenting the sample size in smaller groups to approximate the quantity found in the larger group. Meanwhile, Figure 1.2(b) demonstrates a scenario where the sample size in a larger group is diminished to align with that of the smaller group.

(a) Oversampling method

(b) Undersampling method

Figure 1.2: Concept visualization of the oversampling and the undersampling method

Conversely, algorithmic modification methods adapt existing classification algorithms to bolster their effectiveness in handling imbalanced data. For example, the kENN approach [10] introduced an algorithm that identifies pivotal positive instances and utilizes them to accurately delineate the positive class boundary. The authors conducted a comparative analysis of their method against various k-NN-based classification techniques on diverse imbalanced datasets, demonstrating the superior performance of their approach in terms of classification accuracy, F-measure, and G-mean. Furthermore, the authors explored the impact of different parameters, such as the number of exemplars and the value of k on classification performance.

Zahra Hajizadeh, Mohammad Taheri, and Mansoor Zolghadri Jahromi [5] presented a method for classifying imbalanced data using nearest neighbor classification with locally weighted distance. This method addresses the challenge by assigning greater weight to the nearest neighbors belonging to the minority class. Comparative assessments against several other classification methods on diverse imbalanced datasets revealed that the locally weighted distance method excels in terms of classification accuracy and F-measure.

Cost-sensitive methods focus on assigning more weight to errors made on the minority class and may be applied either to the data or integrated into the classification algorithms themselves.

1.2 Nearest neighbor Literature

The fundamental principle of the nearest neighbor approach, as summarized by the statement “similar things are likely to be similar” [4], underlies its enduring popularity in classification. The enduring popularity of k-nearest neighbors (K-NN) can be attributed to four key factors, as outlined below.

Firstly, the flexibility of defining similarity in nearest neighbor methods involves choosing a feature space for data representation and an associated distance metric, such as Euclidean space and Euclidean distance.

Secondly, their efficiency in approximating nearest neighbor searches makes them well-suited for handling large, high-dimensional datasets, ensuring scalability.

Thirdly, these methods embrace a nonparametric approach, relying on data-driven predictions rather than rigid model assumptions.

Lastly, nearest neighbor methods support their decisions by revealing the nearest neighbors found in the dataset.

The Nearest Neighbor classifier (NN) [5] is particularly renowned in the field of data mining for its simplicity and impressive performance. In situations with a substantial number of training instances, the classification error rate of the nearest neighbor classifier typically remains only about twice that of the Bayes classifier.

The k-NN algorithm operates by identifying the k closest instances in the training set to a given query instance, based on a distance metric like Euclidean distance. For classification tasks, the predicted label for the query instance is determined by a majority vote among the labels of the k nearest neighbors. The choice of the value of k is a critical parameter for the k-NN algorithm, as it can significantly impact its performance. A smaller value of k increases sensitivity to local data variations, while a larger value of k enhances robustness to noise and outliers but may reduce sensitivity to local data structures.

The nearest neighbor algorithm, also referred to as the 1-nearest neighbor (1-NN), is a specific case of the k-nearest neighbor (k-NN) algorithm, where k is set to 1. In other words, the nearest neighbor algorithm identifies the single closest instance in the training set to a given query instance.

1.3 Review of density-based scoring

Hawkins [12] has put forth a common definition for outliers, characterizing them as observations that significantly deviate from the norm within a dataset. Outliers represent data points that are divergent, incongruous, inconsequential, or even potentially malevolent when compared to the majority of the data in a dataset. Detecting outliers [15] holds particular importance in various applications, including network intrusion detection, identifying fraudulent transactions, and aiding in medical diagnostics.

Over recent decades, numerous approaches for detecting outliers have emerged. These methods can be categorized into distribution-based, distance-based, clustering-based, and density-based techniques.

Within distribution-based approaches, an object is identified as an outlier when it exhibits a substantial deviation from a predefined standard distribution, such as the normal distribution or Poisson distribution.

In distance-based methods, the distance between an instance and its neighbors is calculated, and objects far removed from their group are identified as outliers. This is the most widely used technique for outlier detection, with various methods developed to compute distances between instances.

In cluster-based methods identify outliers during the process of forming clusters, where instances not assigned to any cluster are considered outliers.

In density-based method, the density of a specific object is assessed in relation to neighboring instances. The density of an instance is computed and compared with that of neighboring instances, yielding an outlier score in the density-based method. In this approach, normal instances and their neighbors exhibit similar densities, while outliers deviate from this density pattern. By evaluating the density of an instance in relation to that of its neighbors, this approach provides an effective means of detecting outliers in large datasets. As an example, consider the Local Outlier Factor (LOF), which provides a measure of how much an object deviates from its locally reachable neighborhood. An outlier is determined by selecting the instance with the highest local outlier factor, while objects with low LOF values are considered normal.

1.4 Thesis Overview

The remaining sections of this thesis book are structured as follows. The subsequent chapter will provide an overview of background knowledge and related work. Chapter III will delve into the conglomerate nearest neighbor classifier and its associated algorithm. Chapter IV will thoroughly explore the MOF guided classifier and its corresponding algorithm. The final chapter will serve as a conclusion to this research and will outline potential issues for future work.

CHAPTER II

BACKGROUND KNOWLEDGE AND RELATED WORKS

This chapter will delve into the essential principles of the k-nearest neighbor (k-NN) classifier, elucidate the process of evaluating its classification performance, and provide an overview of related research.

2.1 k-Nearest Neighbor classifier

The k-Nearest Neighbors (k-NN) algorithm [8], [16] is a popular instance-based learning approach used in machine learning. The k-NN classifier operates by determining the class label of a query instance based on its proximity to labeled instances in a training set. Specifically, the algorithm identifies the k nearest instances to the query instance using a distance metric, such as Euclidean. The class label of the query instance is then determined by identifying the single most frequent class label among the k nearest neighbors.

Consider a dataset denoted as D split as D_{train}, representing the training dataset, and D_{test}, representing the testing dataset. Further, let n_{train} be the number of instances in the training dataset, n_{test} be the number of instances in the testing dataset, and d be the dimension, referring to the number of attributes or features. To gain a comprehensive grasp of the algorithm's functioning, the algorithm breaks down the steps as follows:

- Given the training dataset:

$$(x_1^1, x_1^2, ..., x_1^d, y_1), (x_2^1, x_2^2, ..., x_2^d, y_2), ..., (x_{n_{train}}^1, x_{n_{train}}^2, ..., x_{n_{train}}^d, y_{n_{train}}) \in D_{train}$$

1. Collect labeled training data.
2. Store the training data in a data structure that allows for efficient distance

calculations.

3. Determine the value of k, the number of nearest neighbors to consider.

- Given the testing dataset:

$$(x_1^1, x_1^2, ..., x_1^d, y_1), (x_2^1, x_2^2, ..., x_2^d, y_2), ..., (x_{n_{test}}^1, x_{n_{test}}^2, ..., x_{n_{test}}^d, y_{n_{test}}) \in D_{test}.$$

For each test instance

1. Calculate Euclidean distance with all training instances.
2. Find the k nearest neighbors.
3. Determine the majority class among the k nearest neighbors.
4. Assign the majority class as the predicted output for the test instance.

This algorithm offers several advantages [13],[14], including ease of implementation, efficient performance with small training sets, independence from prior knowledge about the data's structure within the training set, no need for retraining when adding new training patterns, and an easily interpretable output.

However, the k-NN classification algorithm has limitations. It involves computing distances between the training data and a test instance, requiring additional computation to identify the k nearest neighbors. This negatively impacts the algorithm's scalability. Depending on the composition of a test instance's neighborhood, the appropriate k value for classification may vary significantly. A high k value can yield good accuracy, but it results in increased computational cost for nearest neighbor searching. Conversely, using a small k value reduces computational cost but may adversely affect accuracy.

The k-nearest neighbor (k-NN) algorithm is recognized to be afflicted by two primary issues [18],[19]:

1. It entails significant computational time and storage space, particularly when handling substantial datasets.

2. The effectiveness of classification relies on a single parameter, namely, k.

To tackle the initial challenge, researchers delve into potential solutions by considering techniques that reduce dimensionality or select pertinent features. This may involve the application of self-organizing feature maps (SOM) [7] or the incorporation of forward and backward sequential selection.

This research emphasizes on addressing the second challenge, wherein each test instance should not be restricted to consistently employing a fixed number of neighbors.

Earlier k-NN classification approaches typically determine the value of k either by assigning a constant fixed value for all test data or by employing cross-validation to estimate the k value for each test data point. This approach often results in a suboptimal prediction rate in practical classification scenarios because it does not account for the underlying data distribution. An example to illustrate this point is given below.

Figure 2.1 illustrates a k-NN classifier scenario with two distinct groups. The blue group is denoted as group 1, and the red group is referred to as group 2. The yellow instance represents the test instance, and there are two test instances, namely t1 and t2. Let's focus on t1. When using k=3, the prediction correctly identifies it as group 1. However, considering the example of t2, with k=3, the prediction mistakenly labels it as group 1. In contrast, when t2 is evaluated with k=5, the prediction is accurate, categorizing it as group 2. This highlights the importance of adapting the number of nearest neighbors for different test instances.

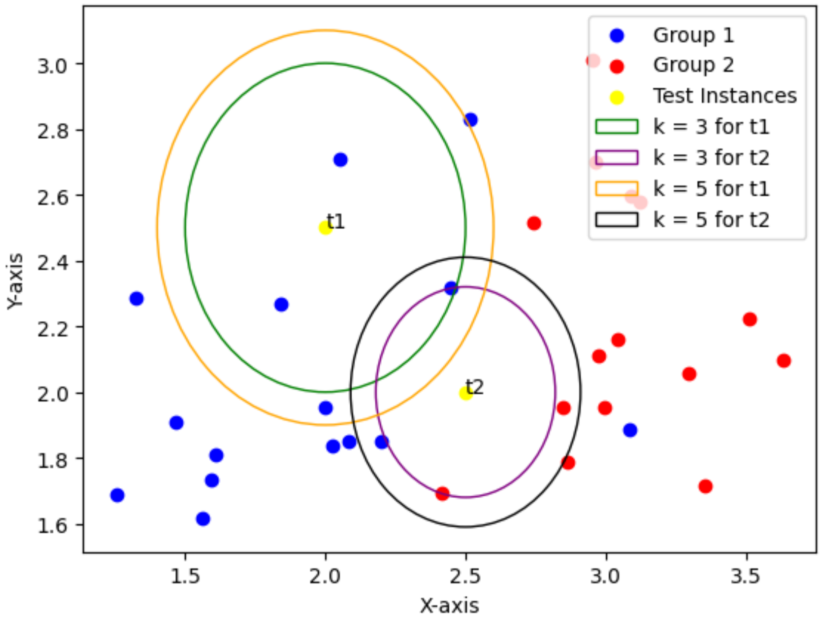
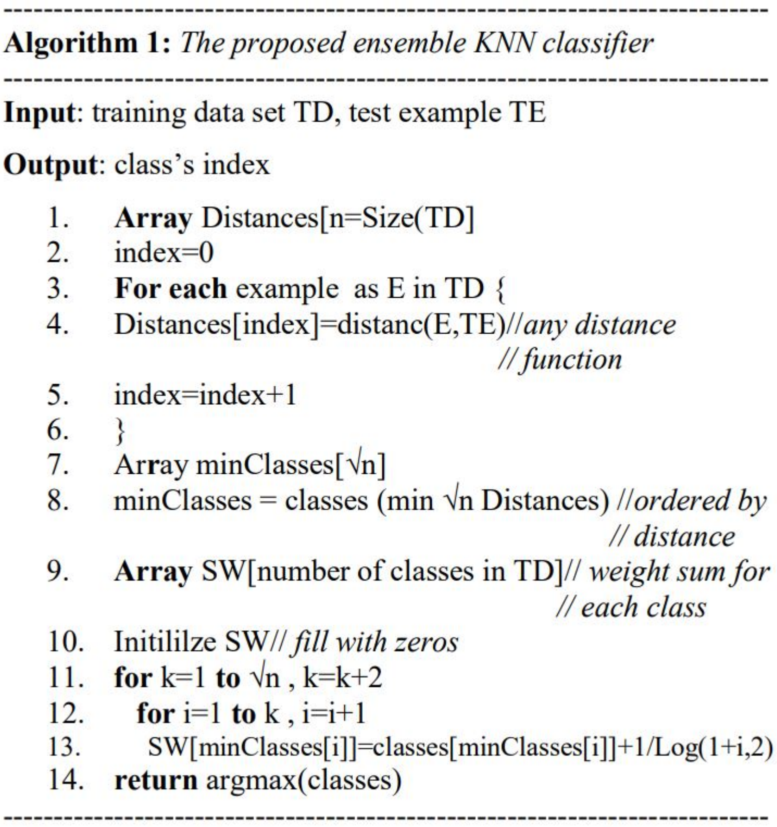


Figure 2.1: Example of k nearest neighbor for k=3 and k=5

2.1.1 The ensemble K-NN classifier

In 2014, A. B. Hassanat et al [6] introduced a novel approach to address the issue of classification performance dependency on the selection of an optimal number of neighbors, eliminating the necessity for specifying a fixed 'k' value in the classifier. Their proposal involves employing ensemble learning based on the nearest neighbor rule. Essentially, they utilize the traditional k-Nearest Neighbors (k-NN) classifier multiple times, each time with a different 'k' value. The range of 'k' values considered typically starts from 1, 3, 5, and extends up to the square root of the training set size. Each classifier in this ensemble contributes its vote for a particular class, and the final classification decision is made using a majority rule, wherein the class with the highest number of votes is selected. Figure 2.2 shows the algorithm of this classifier.



Algorithm 1: *The proposed ensemble KNN classifier*

Input: training data set TD, test example TE

Output: class's index

```
Array Distances[n=Size(TD]
index=0
For each example  as E in TD {
Distances[index]=distanc(E,TE)//any distance
                              // function
index=index+1
}
Array minClasses[√n]
minClasses = classes (min √n Distances) //ordered by
                                        // distance
Array SW[number of classes in TD]// weight sum for
                                 // each class
Initililze SW// fill with zeros
for k=1 to √n , k=k+2
  for i=1 to k , i=i+1
    SW[minClasses[i]]=classes[minClasses[i]]+1/Log(1+i,2)
return argmax(classes)
```

Figure 2.2: ensemble knn Algorithm

2.2 Mass-ratio-variance Outlier Factors

In 2021, a parameter-free outlier score called mass-ratio-variance outlier factor (MOF) [2] was introduced. This method is density-based approach, calculates the variance of the mass-ratio distribution of a given data instance, where the density of the instance is first computed and then compared to the densities of its neighboring instances. The resulting outlier score indicates the extent to which the instance deviates from the norm in terms of its density. Normal instances and their neighbors have similar densities, whereas outliers have densities that differ significantly from those of their neighbors. By evaluating the density of an instance in relation to that of its neighbors, this approach provides an effective means of detecting outliers in a large dataset. The mass-ratio of other instances is defined as the ratio of the number of instances within the sphere of the distance from the computed instance to that of other instances. The main mathematical notations are defined as described in [2]:

Definition 1. (Distance between $\boldsymbol{p}$ and $\boldsymbol{q}$) Given a dataset $D \subseteq \mathbb{R}^d$, the Euclidean distance of data point $\boldsymbol{p} = (p_1, ..., p_d) \in D$ to data point $\boldsymbol{q} = (q_1, ..., q_d) \in D$ denoted as $d(\boldsymbol{p}, \boldsymbol{q})$ is defined as

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_{i=1}^{d} (\boldsymbol{p}_i - \boldsymbol{q}_i)^2}.$$

Definition 2. (Neighborhoods of data point $\boldsymbol{q}$ with respect to data point $\boldsymbol{p}$) Given a dataset $D \subseteq \mathbb{R}^d$, the set of all data points within the neighborhood of data point $\boldsymbol{q} \in D$ with respect to data point $\boldsymbol{p} \in D$ is define as the set of points that lies within the ball centered at data point $\boldsymbol{q}$ with the radius $d(\boldsymbol{q}, \boldsymbol{p})$:

$$N_{\boldsymbol{p}}(\boldsymbol{q}) = \{\boldsymbol{o} \in D \mid d(\boldsymbol{q}, \boldsymbol{o}) \leqslant d(\boldsymbol{q}, \boldsymbol{p})\}.$$

Definition 3. (The mass-ratio of data point $\boldsymbol{q}$ with respect to data point $\boldsymbol{p}$) Given a dataset $D \subseteq \mathbb{R}^d$ and data point $\boldsymbol{p} \in D$ for any data point $\boldsymbol{q} \in D - \{\boldsymbol{p}\}$, the mass-ratio of data point $\boldsymbol{q}$ with respect to data point $\boldsymbol{p}$ is defined as

$$massR_{\boldsymbol{p}}(\boldsymbol{q}) = \frac{\mid N_{\boldsymbol{p}}(\boldsymbol{q}) \mid}{\mid N_{\boldsymbol{q}}(\boldsymbol{p}) \mid}.$$

Definition 4. (MOF of data point $\boldsymbol{p}$) Given a dataset $D \subseteq \mathbb{R}^d$ and the number of data points in D is n for data point $\boldsymbol{p} \in D$, μ_p is defined as mean of mass-ratio distribution of data point $\boldsymbol{p}$ and MOF of data point $\boldsymbol{p}$ is defined as the variance of the mass-ratio distribution of data point $\boldsymbol{p}$:

$$\mu_p = \frac{\sum_{i=1,q_i \neq p}^{n} massR_{\boldsymbol{p}}(\boldsymbol{q}_i)}{n-1}.$$

$$MOF(\boldsymbol{p}) = \frac{\sum_{i=1,q_i \neq p}^{n} (massR_{\boldsymbol{p}}(\boldsymbol{q}_i) - \mu_p)^2}{n-1}.$$

Figure 2.3 shows the algorithm of MOF. Commence by computing the distances between all instances in the dataset, proceed to determine the neighborhoods of data point $\boldsymbol{q}$ with respect to data point $\boldsymbol{p}$ ($N_{\boldsymbol{p}}(\boldsymbol{q})$) and the neighborhoods of data point $\boldsymbol{p}$ with respect to data point $\boldsymbol{q}$ ($N_{\boldsymbol{q}}(\boldsymbol{p})$) , then calculate the mass-ratio of data point $\boldsymbol{q}$ with respect to data point $\boldsymbol{p}$, and ultimately compute the MOF using the provided formula.

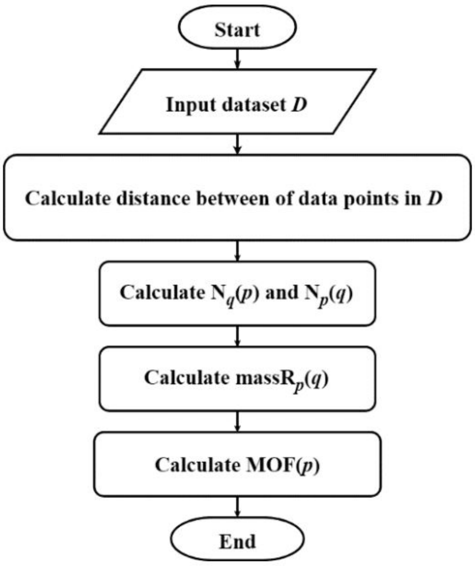

Start
Input dataset *D*
Calculate distance between of data points in *D*
Calculate $N_q(p)$ and $N_p(q)$
Calculate $massR_p(q)$
Calculate MOF(*p*)
End

Figure 2.3: MOF Algorithm

2.3 Evaluation

The metrics used to evaluate the quality of classification are derived from a confusion matrix [5],[17] as depicted in Table 2.2, which tabulates the number of correctly and incorrectly classified instances for each class. The rows in this matrix represent the true or actual class assignments of instances, while the columns display the predicted class assignments made by the algorithm. The sum of all the entries within the confusion matrix equals the total population of the test set. In other words, every sample in the test set must be categorized within one of the matrix's entries.

	Positive prediction	Negative prediction
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

Table 2.1: Confusion matrix for a two-class problem.

2.3.1 Precision

The precision [17] is defined as the number of true positives divided by the number of all samples that were classified as positive

$$Precision = \frac{TP}{TP+FP}.$$

2.3.2 Recall

The recall [17], also called true positive rate (TPR) or sensitivity, is the number of true positives divided by all samples that are actually positives

$$Recall = \frac{TP}{TP+FN}.$$

2.3.3 F1-score

To penalize false positives (also known as type I errors) and false negatives (also known as type II errors), a new metric that takes into account these values merges the precision and the recall (sensitivity) together. The Fβ-*score* creates a trade-off between these two metrics [17]. it is defined as

$$F\beta\text{-}score = (1+\beta^2)\frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}.$$

which in terms of TP, FN, and FP, it goes as

$$F\beta\text{-}score = \frac{(1+\beta^2) \times TP}{(1+\beta^2) \times TP + \beta^2 \times FN + FP}.$$

A particular case of the $F\beta$-*score* is the $F1$-*score*; this happens when $\beta = 1$, that is

$$F1\text{-}score = \frac{2 \times TP}{2 \times TP + FN + FP}.$$

2.3.4 Accuracy

The accuracy [17] of a machine learning algorithm is one of the most simple metrics; it is calculated by taking all the samples that were correctly predicted by the algorithm and dividing them by the total number of samples. Using the notation of the confusion matrix, the accuracy is

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

2.3.5 Example

Here's an example of a confusion matrix for a binary classification problem, such as determining whether an email is spam or not.

	Positive prediction	Negative prediction
Positive class	120	20
Negative class	10	900

Table 2.2: example of a confusion matrix for a binary classification problem

In this example:

- The top-left cell (120) represents the number of emails that were correctly classified as “Spam” (True Positives).
- The bottom-left cell (10) represents the number of emails that were actually “No Spam” but were incorrectly classified as “Spam” (False Positives).
- The top-right cell (20) represents the number of emails that were actually “Spam” but were incorrectly classified as “No Spam” (False Negatives).
- The bottom-right cell (900) represents the number of emails that were correctly classified as “No Spam” (True Negatives).

To calculate precision, recall, F1-score, and accuracy using the values from the provided confusion matrix, apply the following formulas.

$$Precision = \frac{TP}{TP + FP} = \frac{120}{120 + 10} = 0.9231.$$

$$Recall = \frac{TP}{TP + FN} = \frac{120}{120 + 20} = 0.8571.$$

$$F1\text{-}score = \frac{2 \times TP}{2 \times TP + FN + FP} = \frac{2 \times 120}{2 \times 120 + 20 + 10} = 0.8889.$$

$$Accuracy = \frac{120 + 900}{120 + 900 + 10 + 20} = 0.9643.$$

In this example, while the accuracy is relatively high at approximately 96.43%, it's important to understand why accuracy alone may not be the best metric to evaluate the performance of a classification model. The reason lies in the class imbalance present in the data. In the provided confusion matrix:

- There are 900 true negatives (No Spam correctly classified as No Spam).
- There are 120 true positives (Spam correctly classified as Spam).
- There are 20 false negatives (Spam incorrectly classified as No Spam).
- There are 10 false positives (No Spam incorrectly classified as Spam).

The class distribution is skewed because there are many more "No Spam" instances (920) compared to "Spam" instances (130).

Accuracy is the ratio of correctly classified instances to the total number of instances. In this case, it gives a high accuracy score because the model correctly classifies a majority of the "No Spam" instances.

The problem with using accuracy in this scenario is that it doesn't take into account the consequences of misclassifying the minority class (in this case, "Spam"). A high accuracy can be misleading when dealing with imbalanced datasets because the model may appear to perform well due to the dominant class, while it might perform poorly in correctly identifying the minority class.

This is where precision, recall, and the F1-score come into play. Precision and recall provide insights into the model's performance on the positive class ("Spam" in this case). Precision tells you how many of the predicted positive instances were actually positive, and recall tells you how many of the actual positive instances were correctly predicted.

In this example, the precision and recall values help assess the model's ability to correctly identify "Spam" emails without being overly biased by the large number of "No Spam" emails.

In summary, accuracy can be misleading when dealing with imbalanced datasets, so it's important to consider multiple metrics like precision, recall, and F1-score to have a more comprehensive evaluation of a model's performance, especially in situations where class distribution is uneven.

2.4 Data used

This section provides an overview of the data utilized in the experiment, encompassing both synthesized data and information sourced from the UCI dataset.

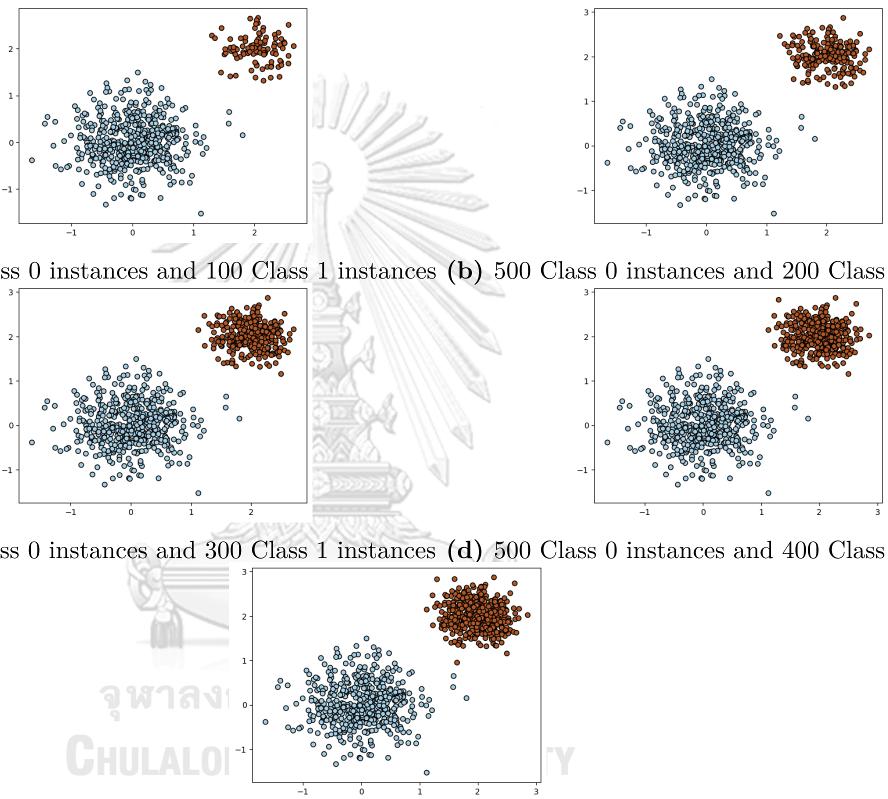
2.4.1 Synthesized datasets

The synthesized data is divided into three categories based on the level of sample overlap between the two groups, as delineated follow: 1. No overlap 2. Slight overlap 3. Large overlap. Within each category, data is created using three different patterns: Gaussian, moon shaped, and circle. Each pattern is produced with five unique instances for each class, facilitating a comparison between balanced and imbalanced datasets. Specifically, Class 0 consistently comprises 500 instances, while Class 1 varies from 100 to 500 instances, representing different degrees of class imbalance. In this context, the color blue is used to signify Class 0, while the color red is employed to denote Class 1.

2.4.1.1 No overlap

1. Gaussian format

Figure 2.4 presents a Gaussian plot of the synthesized data, demonstrating a complete absence of overlap between the two classes. In Figure 2.4(a), Class 0 comprises 500 instances, and Class 1 consists of 100 instances. Figure 2.4(b) displays Class 0 with 500 instances and Class 1 with 200 instances. Figure 2.4(c) illustrates Class 0 with 500 instances and Class 1 with 300 instances. In Figure 2.4(d), Class 0 contains 500 instances, and Class 1 encompasses 400 instances. Finally, Figure 2.4(e) portrays Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

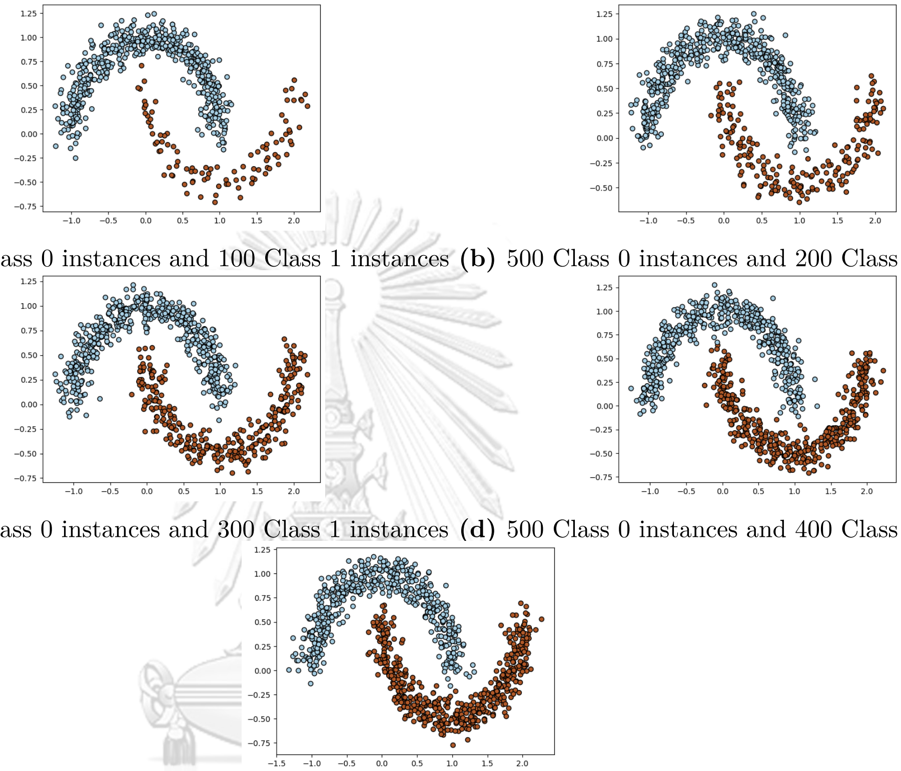
(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.4: Visualization of synthesized Gaussian data with no overlap

2. Moon shaped format

Figure 2.5 showcases a moon-shaped plot of the synthesized data, revealing a complete absence of overlap between the two classes. In Figure 2.5(a), Class 0 consists of 500 instances, while Class 1 comprises 100 instances. Figure

2.5(b) exhibits Class 0 with 500 instances and Class 1 with 200 instances. Figure 2.5(c) depicts Class 0 with 500 instances and Class 1 with 300 instances. In Figure 2.5(d), Class 0 encompasses 500 instances, and Class 1 includes 400 instances. Finally, Figure 2.5(e) illustrates Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

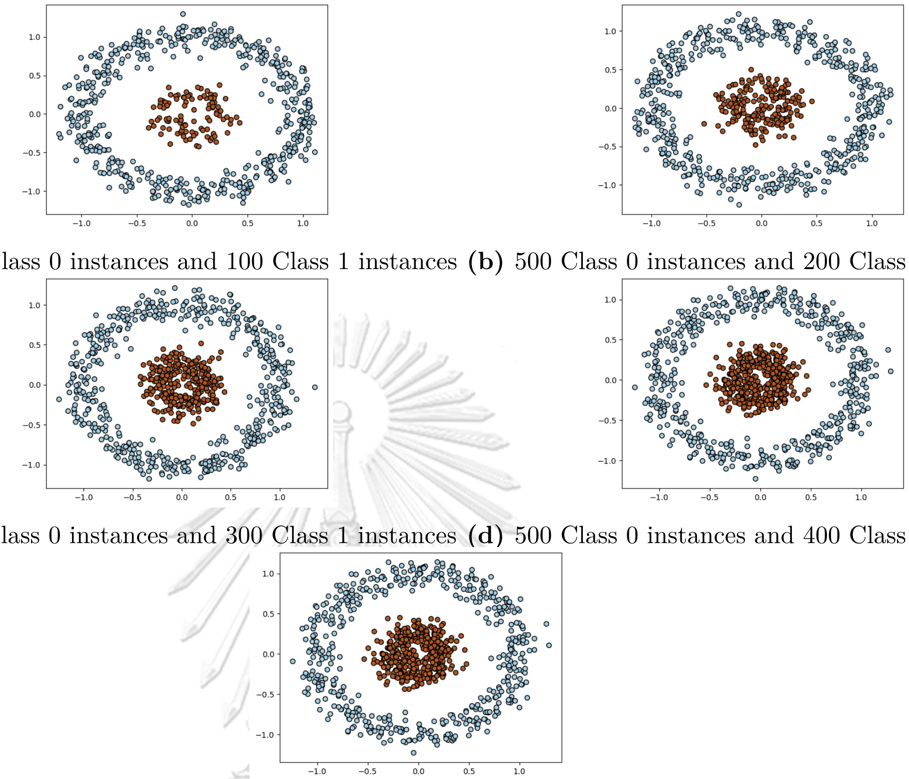
(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.5: Visualization of synthesized moon shaped data with no overlap

3. Circle format

Figure 2.6 displays a circular plot of the synthesized data, clearly indicating the absence of overlap between the two classes. In Figure 2.6(a), Class 0 is represented by 500 instances, while Class 1 comprises 100 instances. Figure 2.6(b) presents Class 0 with 500 instances and Class 1 with 200 instances. Figure 2.6(c) showcases Class 0 with 500 instances and Class 1 with 300 instances. In Figure 2.6(d), Class 0 encompasses 500 instances, while Class 1 includes 400 instances. Finally, Figure 2.6(e) illustrates Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

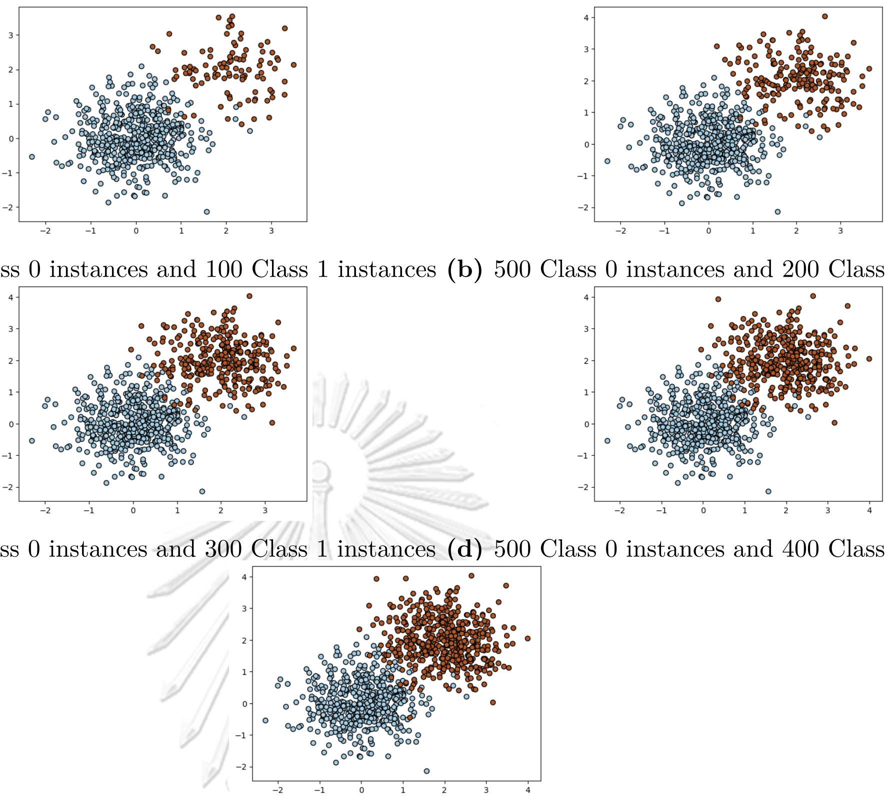
Figure 2.6: Visualization of synthesized circle data with no overlap

2.4.1.2 Slight overlap

1. Gaussian format

Figure 2.7 depicts a Gaussian plot of the synthesized data, showcasing a subtle overlap between the two classes. In Figure 2.7(a), Class 0 is represented by 500 instances, and Class 1 comprises 100 instances. Figure 2.7(b) presents Class 0 with 500 instances and Class 1 with 200 instances. Figure 2.7(c) illustrates Class 0 with 500 instances and Class 1 with 300 instances. In Figure 2.7(d), Class 0 consists of 500 instances, while Class 1 encompasses

400 instances. Finally, Figure 2.7(e) portrays Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

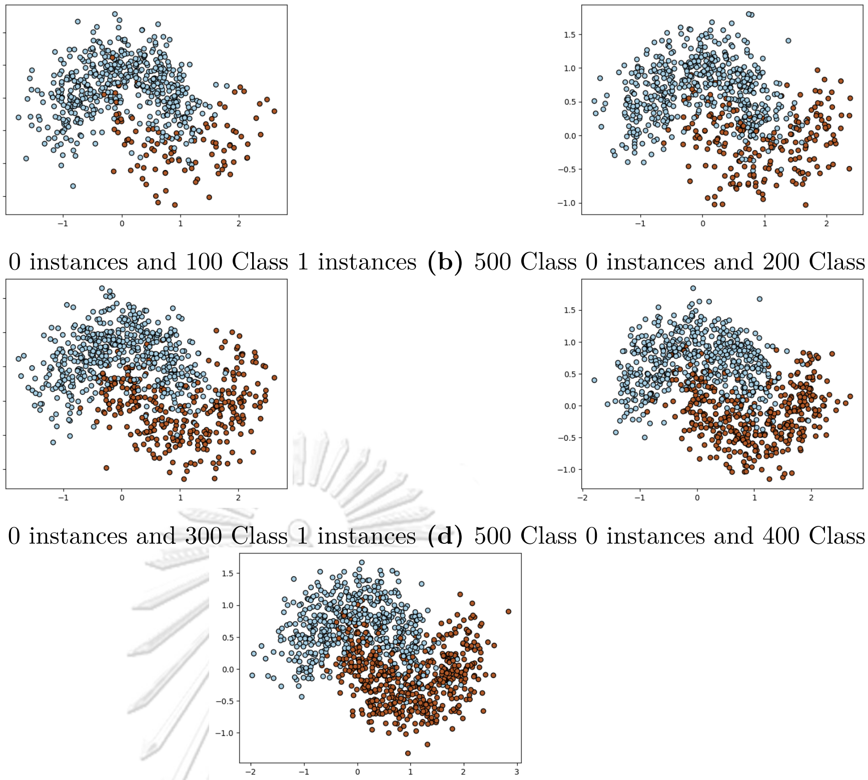
(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.7: Visualization of synthesized Gaussian data with slight overlap

2. Moon shaped format

Figure 2.8 illustrates a moon-shaped plot of the synthesized data, highlighting a subtle degree of overlap between the two classes. Within Figure 2.8(a), Class 0 is represented by 500 instances, while Class 1 consists of 100 instances. In Figure 2.8(b), Class 0 comprises 500 instances, and Class 1 includes 200 instances. Figure 2.8(c) demonstrates Class 0 with 500 instances and Class 1 with 300 instances. Meanwhile, Figure 2.8(d) displays Class 0 encompassing 500 instances, with Class 1 totaling 400 instances. Finally, in Figure 2.8(e), Class 0 is depicted with 500 instances, and Class 1 with 500 instances."



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

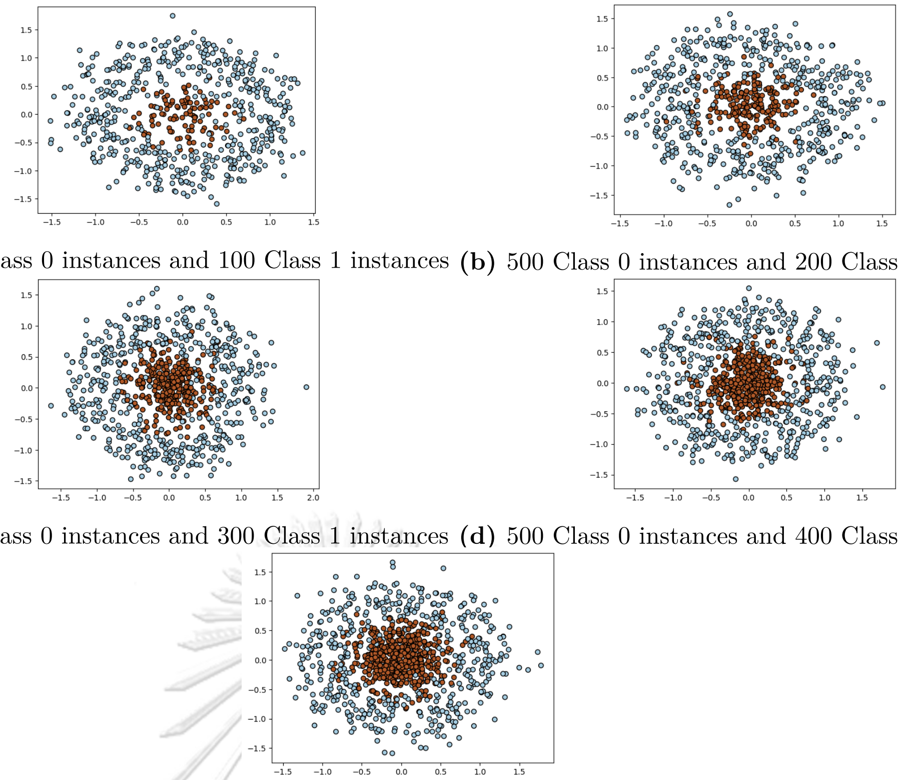
(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.8: Visualization of synthesized moon shaped data with slight overlap

3. Circle format

Figure 2.9 showcases a circular plot of the synthesized data, emphasizing a subtle level of overlap between the two classes. In Figure 2.9(a), Class 0 comprises 500 instances, while Class 1 is represented by 100 instances. Figure 2.9(b) presents Class 0 with 500 instances, and Class 1 includes 200 instances. Figure 2.9(c) illustrates Class 0 with 500 instances and Class 1 with 300 instances. Concurrently, Figure 2.9(d) reveals Class 0 encompassing 500 instances, while Class 1 totals 400 instances. Finally, in Figure 2.9(e), Class 0 is portrayed with 500 instances, and Class 1 consists of 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

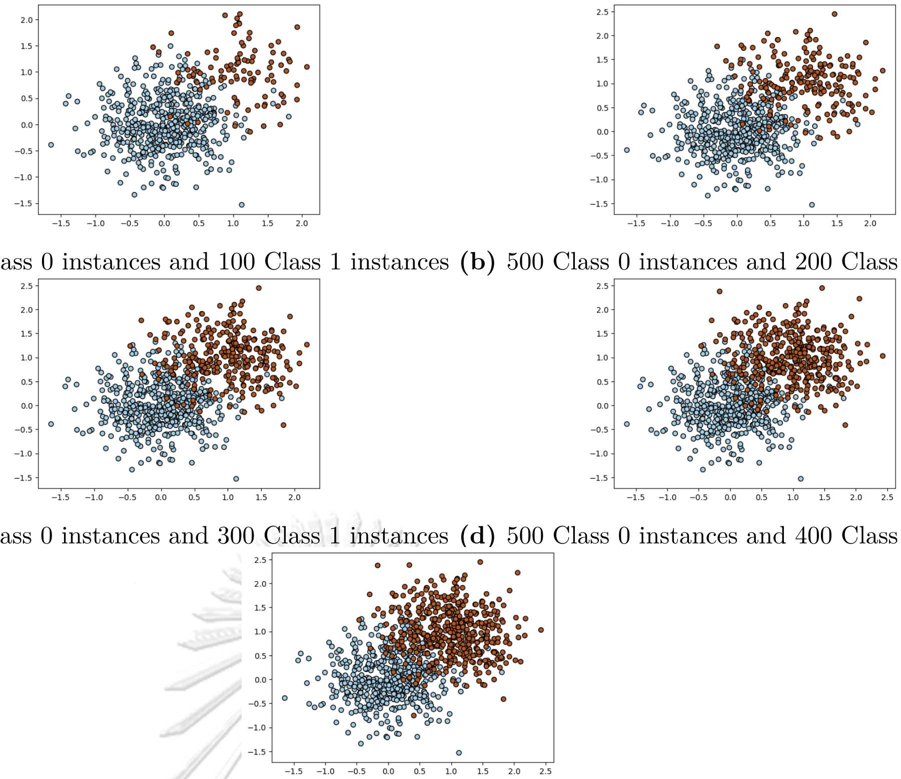
(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.9: Visualization of synthesized circle data with slight overlap

2.4.1.3 Large overlap

1. Gaussian format

   Figure 2.10 displays a Gaussian plot of the synthesized data, highlighting a significant overlap between the two classes. Within Figure 2.10(a), Class 0 is represented by 500 instances, while Class 1 comprises 100 instances. Figure 2.10(b) presents Class 0 with 500 instances, and Class 1 includes 200 instances. Figure 2.10(c) illustrates Class 0 with 500 instances and Class 1 with 300 instances. Meanwhile, in Figure 2.10(d), Class 0 consists of 500 instances, while Class 1 encompasses 400 instances. Finally, Figure 2.10(e) portrays Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

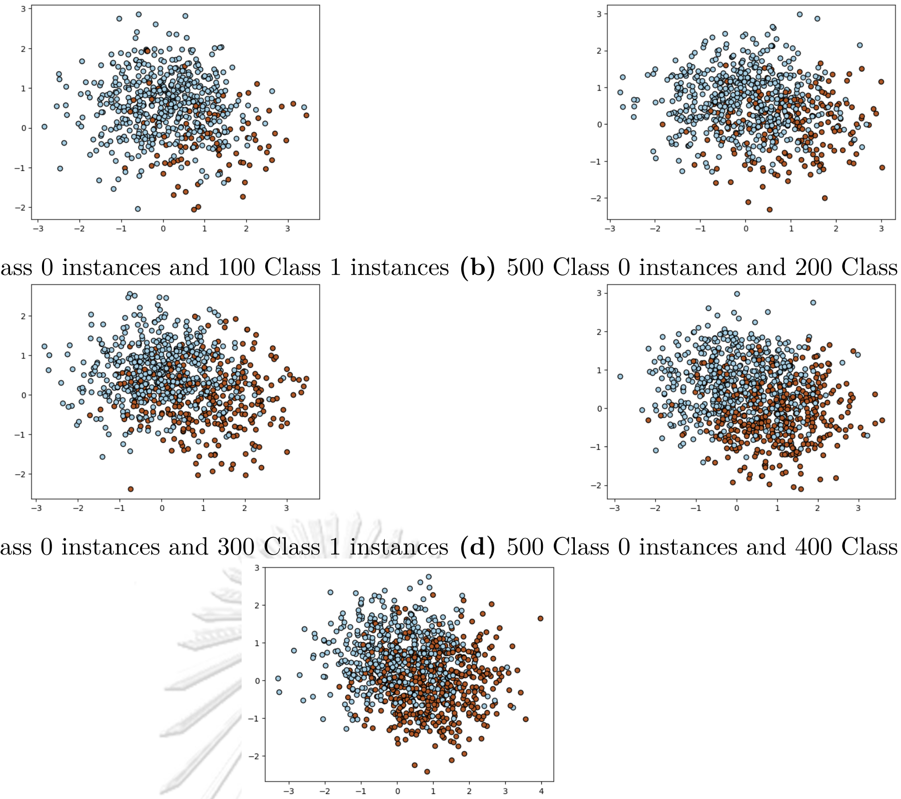
(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.10: Visualization of synthesized Gaussian data with large overlap

2. Moon shaped format

Figure 2.11 showcases a moon-shaped plot of the synthesized data, emphasizing a significant overlap between the two classes. Within Figure 2.11(a), Class 0 is represented by 500 instances, while Class 1 comprises 100 instances. Figure 2.11(b) presents Class 0 with 500 instances, and Class 1 includes 200 instances. Figure 2.11(c) illustrates Class 0 with 500 instances and Class 1 with 300 instances. Meanwhile, in Figure 2.11(d), Class 0 consists of 500 instances, while Class 1 encompasses 400 instances. Finally, Figure 2.11(e) portrays Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

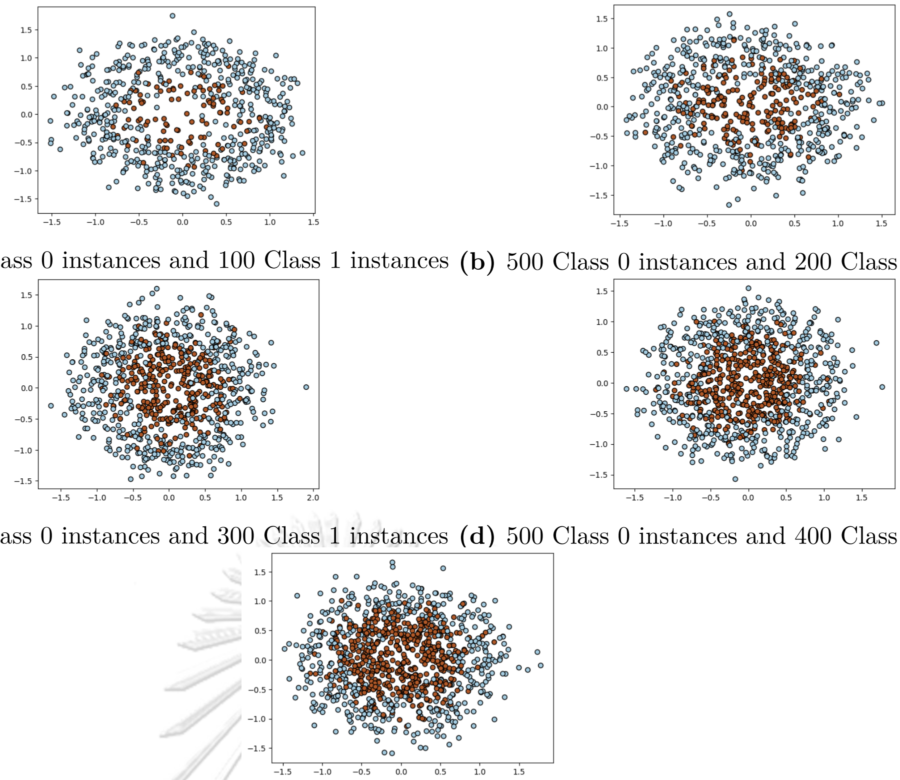
(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.11: Visualization of synthesized moon shaped data with large overlap

3. Circle format

Figure 2.12 presents a circular plot of the synthesized data, emphasizing a significant overlap between the two classes. Within Figure 2.12(a), Class 0 is represented by 500 instances, while Class 1 comprises 100 instances. Figure 2.12(b) displays Class 0 with 500 instances, and Class 1 includes 200 instances. Figure 2.12(c) illustrates Class 0 with 500 instances and Class 1 with 300 instances. Meanwhile, in Figure 2.12(d), Class 0 consists of 500 instances, while Class 1 encompasses 400 instances. Finally, Figure 2.12(e) portrays Class 0 with 500 instances and Class 1 with 500 instances.



(a) 500 Class 0 instances and 100 Class 1 instances **(b)** 500 Class 0 instances and 200 Class 1 instances

(c) 500 Class 0 instances and 300 Class 1 instances **(d)** 500 Class 0 instances and 400 Class 1 instances

(e) 500 Class 0 instances and 500 Class 1 instances

Figure 2.12: Visualization of synthesized circle data with large overlap

2.4.2 UCI datasets

In this part, we present the outcomes derived from ten real-world datasets sourced from the UCI repository. These datasets include the Wine dataset, the Sonar dataset, the Glass dataset, the Harberman dataset, the Liver dataset, the ionosphere dataset, the Wholesale dataset, the Cancer dataset, the German dataset, and the QSAR dataset, all of which consist of only two classes. The table labeled as 2.3 provides essential information for each dataset, including its name, the number of instances denoted in the “#Inst” column, the number of instances in majority in the “#maj”, the number of instances in minority in the “#min”, and the number of attributes specified in the “#Att” column.

No.	Name	#Inst	#Maj	#Min	#Att
1	wine	178	107	71	13
2	Sonar	208	111	97	50
3	Glass	214	197	17	9
4	Haberman	306	225	81	3
5	Liver	345	200	145	6
6	Ionosphere	351	225	126	34
7	Wholesale	440	316	124	6
8	Cancer	709	458	251	9
9	German	1000	700	300	24
10	QSAR	1055	699	356	41

Table 2.3: Description of the dataset used

This chapter explores the algorithm that will be compared with the method presented in the subsequent chapter and outlines the methodology for evaluating the classifier's performance. The performance metrics employed in this study include precision, recall, and F1-score, along with accuracy. Additionally, discussions encompass the datasets utilized in the experiments, comprising both synthesized datasets and real-world datasets sourced from the UCI dataset.

CHAPTER III

CONGLOMERATE NEAREST NEIGHBOR CLASSIFIER

This chapter elucidates the Conglomerate Nearest Neighbor Classifier, encompassing the algorithm itself, experimental findings derived from both synthesized data and the UCI dataset, and concludes with a comprehensive discussion.

3.1 Conglomerate Nearest Neighbor Classifier (CNNC) algorithm

The suggested composite nearest neighbor classifier involves two distinct learning phases:

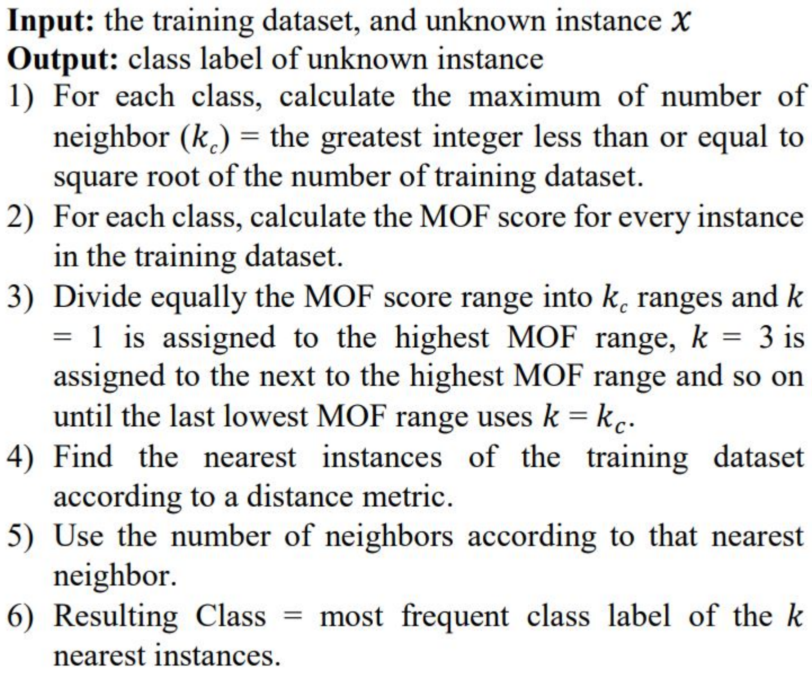
1. the training phase, which involves assigning the number of nearest neighbors to all instances, and
2. the testing phase, where the class of an instance is determined.

During the training phase, the conglomerate nearest neighbor algorithm establishes the maximum number of nearest neighbors for the dataset. This is achieved by determining an odd integer, denoted as K, which is less than or equal to the square root of the number of instances in each class. MOF is subsequently computed for each class and partition, considering odd integers ranging from 1 to K to determine the number of nearest neighbors assigned to each instance. It is important to note that the choice of the number of neighbors is influenced by the instance's location within the dataset. For instances situated within the interior, where they are surrounded by other instances, a higher number of neighbors is assigned to enhance predictive accuracy. Conversely, for instances positioned at the dataset's border, a smaller number of neighbors is utilized to mitigate misclassification. If an instance is far from any cluster, a small number of neighbors (i.e., 1) is considered effective.

To ascertain the number of neighbors based on MOF, the range of MOFs for each class, derived from the training dataset, is divided into segments, with each segment spanning from the smallest to the largest MOF values. These segments are evenly divided into the largest integer less than or equal to the square root of the number of instances in the respective class (k_c). For example, the highest MOF range is assigned a value of k=1, the next highest range is assigned a value of k=3, and this pattern continues until the lowest MOF range, which is assigned a value of $k = k_c$.

In the testing phase, when predicting the class of an unknown instance, denoted as x, the conglomerate nearest neighbor algorithm initially identifies the closest instance, x_c. It then extracts the MOF value associated with x_c and uses it to determine the number of neighbors (k_x) to be utilized. Finally, the class of x is determined by assessing the majority class among the k_x nearest neighbors of x.

The algorithm for the conglomerate nearest neighbor classifier is illustrated in Figure 3.1.



```
Input: the training dataset, and unknown instance $x$
Output: class label of unknown instance
1) For each class, calculate the maximum of number of neighbor ($k_c$) = the greatest integer less than or equal to square root of the number of training dataset.
2) For each class, calculate the MOF score for every instance in the training dataset.
3) Divide equally the MOF score range into $k_c$ ranges and $k$ = 1 is assigned to the highest MOF range, $k$ = 3 is assigned to the next to the highest MOF range and so on until the last lowest MOF range uses $k = k_c$.
4) Find the nearest instances of the training dataset according to a distance metric.
5) Use the number of neighbors according to that nearest neighbor.
6) Resulting Class = most frequent class label of the $k$ nearest instances.
```

Figure 3.1: The conglomerate nearest neighbor algorithm

3.2 Experimental results of the conglomerate nearest neighbor classifier

The performance of both the Best k-NN algorithm which is selecting the k that yields the highest F1-score among k values from 1, 3, ..., the square root of the training data and the ensemble NN algorithm will be compared using the precision, recall, F1-score, and accuracy obtained from both the synthesized datasets and UCI datasets, to determine which algorithm outperforms the other.

3.2.1 Synthesized Dataset

Numerical results, initially documented in tables within Appendix A, are now visually represented in this section through bar graphs. This graphical presentation aims to enhance the clarity of distinctions in numbers. The bar graphs illustrate variations in precision, recall, F1-score, and accuracy between the ensemble Nearest Neighbor (ensemble NN) and Best k-NN, as well as between Conglomerate Nearest Neighbor Classifier (CNNC) and Best k-NN. A negative value on the bar graph signifies that the ensemble NN or CNNC exhibits lower performance than Best k-NN, while a positive value indicates that the ensemble NN or CNNC surpasses Best k-NN.

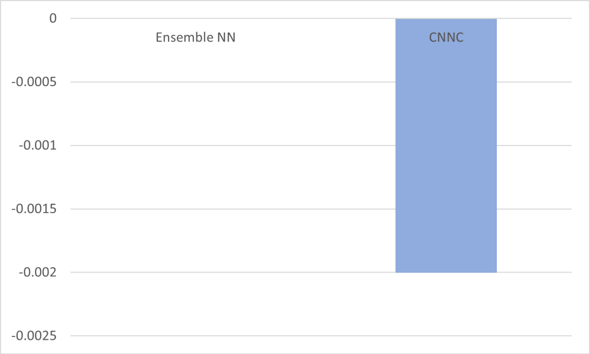
3.2.1.1 No overlap

- Gaussian format

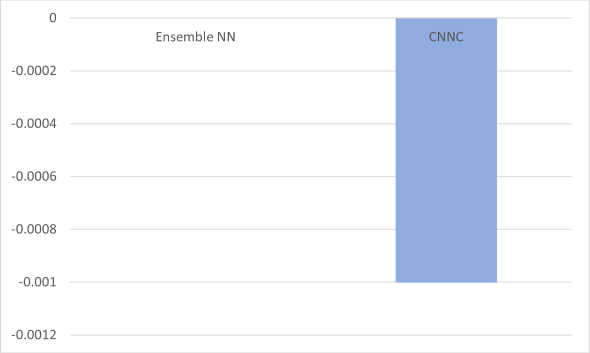
  In scenarios where data is significantly imbalanced, that is, when Class 0 comprises 500 instances and Class 1 has 100 and 200 instances, the precision, recall, F1-score, and accuracy metrics for Best k-NN, Ensemble NN, and CNNC all equal to 1.

  As the data distribution becomes more balanced, Best k-NN consistently maintains precision, recall, F1-score, and accuracy at 1. However, Ensemble NN and CNNC exhibit values that deviate from 1, yet remain close to or equal to 1.

  In Figure 3.2, precision, recall, F1-score, and accuracy metrics are illustrated for a dataset containing 300 class 1 instances. It is evident that Conglomerate Nearest Neighbor Classifier (CNNC) exhibits slightly lower values in precision, recall, F1-score, and accuracy compared to both Best k-NN and Ensemble NN.



(a) Differences in precision, and F1-score between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN



(b) Differences in recall and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN

Figure 3.2: Differences in results between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 300 class 1 instances.

When Class 1 comprises 400 instances, both Ensemble NN and CNNC exhibit only marginal decreases in precision, recall, F1-score, and accuracy compared to Best k-NN as shown in figure 3.3.

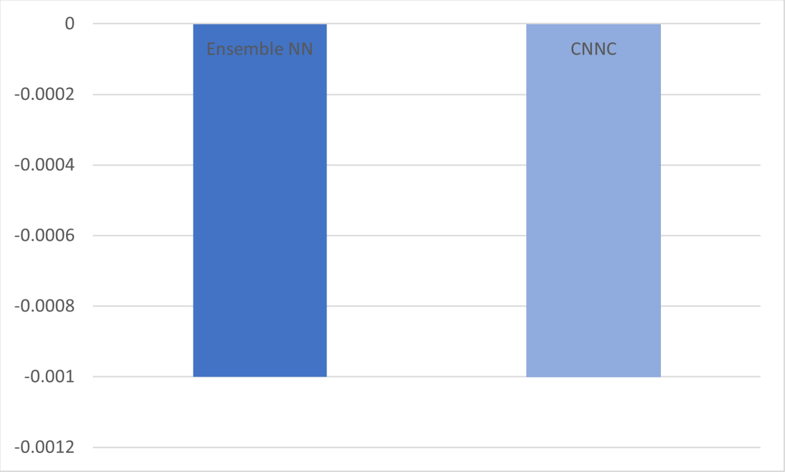


Figure 3.3: Differences in results between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

In Figure 3.4, precision, recall, F1-score, and accuracy metrics are illustrated for a dataset containing 500 class 1 instances. It is evident that both CNNC and Best *k*-NN outperform Ensemble NN in terms of precision, recall, F1-score, and accuracy.

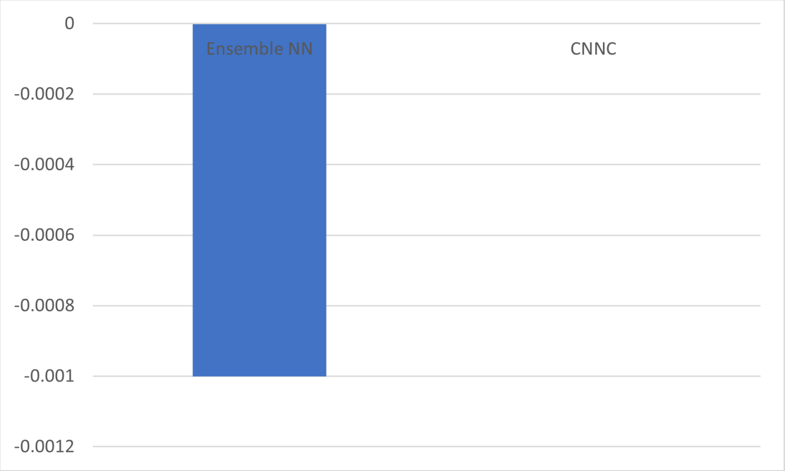
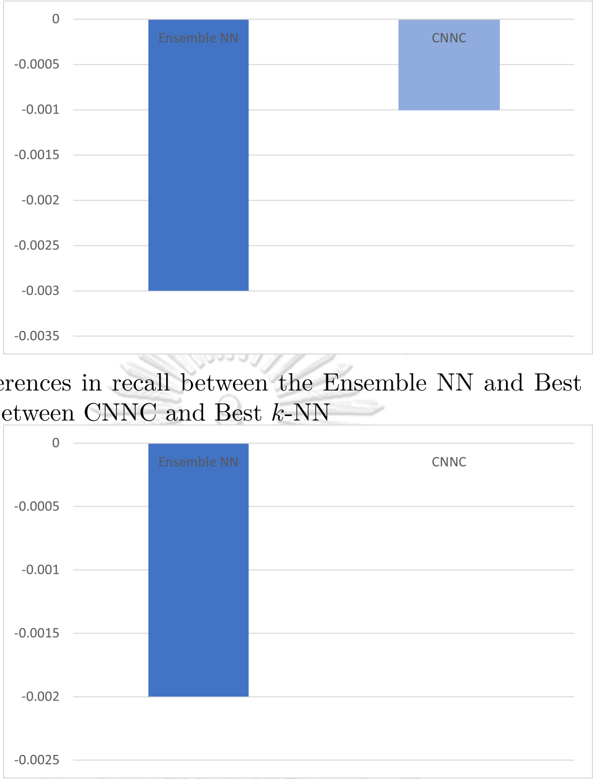


Figure 3.4: Differences in results between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

- Moon shaped format

In the scenario with 100 class 1 instances, the precision and accuracy for

Best *k*-NN, Ensemble NN, and CNNC are all perfect with a score of 1. Nevertheless, Figures 3.5 illustrate that the recall and F1-score of CNNC surpass those of Ensemble NN but are slightly less than those of Best *k*-NN.

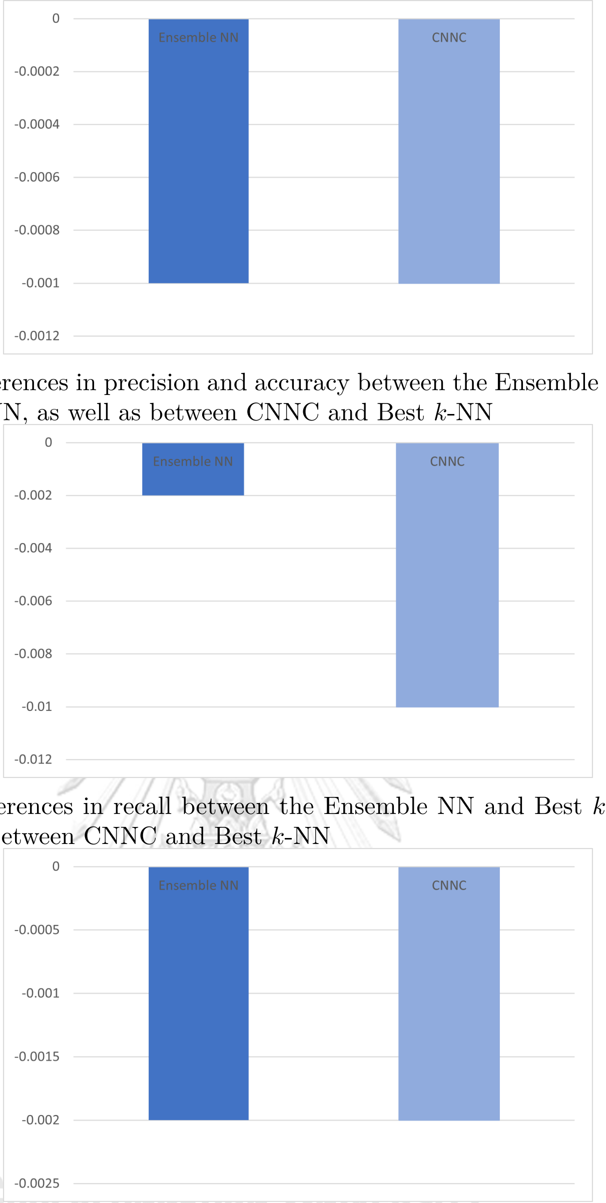


(a) Differences in recall between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN

(b) Differences in F1-score between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN

Figure 3.5: Differences in results between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

In the case of data featuring 200 class 1 instances, the precision and accuracy of both the ensemble NN and CNNC are marginally lower than those of Best *k*-NN, as illustrated in Figure 3.6(a). Meanwhile, Figure 3.6(b) highlights that the recall of CNNC is higher than Ensemble NN but lower than Best *k*-NN. Additionally, Figure 3.6(c) showcases the F1-score, revealing that the values for Ensemble NN and CNNC are slightly lower than those of Best *k*-NN.



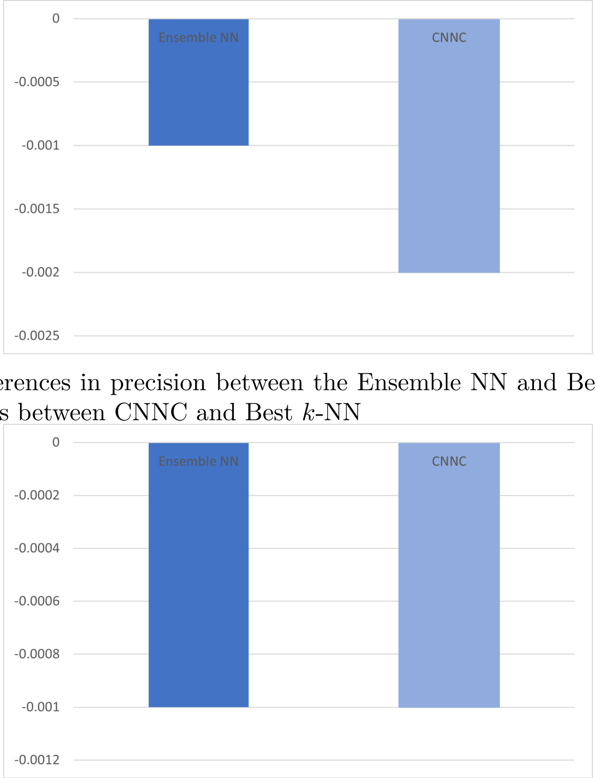
(a) Differences in precision and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN

(b) Differences in recall between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN

(c) Differences in F1-score between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN

Figure 3.6: Differences in results between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

In the context of data comprising 300 class 1 instances, Figure 3.7(a) illustrates that the precision of CNNC is slightly lower than both Ensemble NN and Best *k*-NN. However, in Figure 3.7(b), it is observed that the recall, F1-score, and accuracy of both Ensemble NN and CNNC are equivalent to each other but slightly lower than those of Best k-NN.



(a) Differences in precision between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN

(b) Differences in recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN

Figure 3.7: Differences in results between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 300 class 1 instances.

In the case of data featuring 400 class 1 instances, the precision, recall, F1-score, and accuracy of CNNC are identical to those of Best k-NN, all scoring 1. This performance is superior to that of Ensemble NN, as depicted in Figure 3.8.

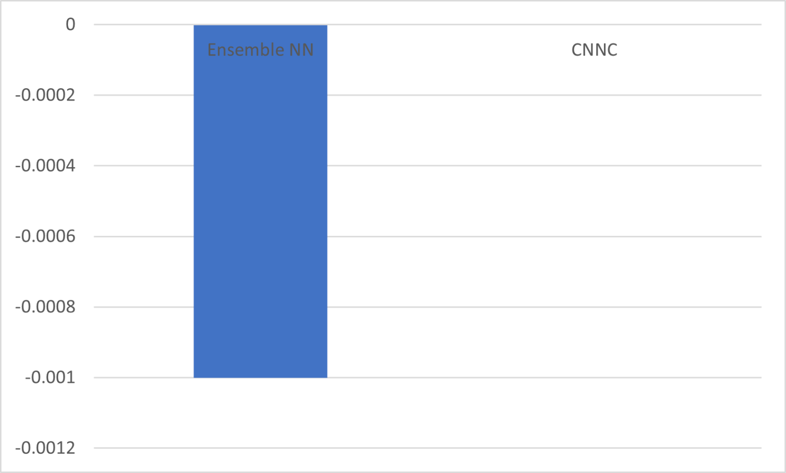
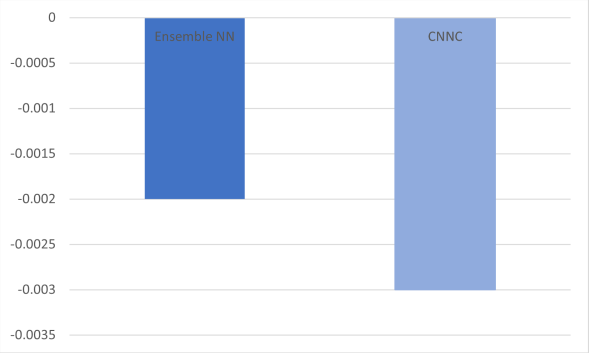
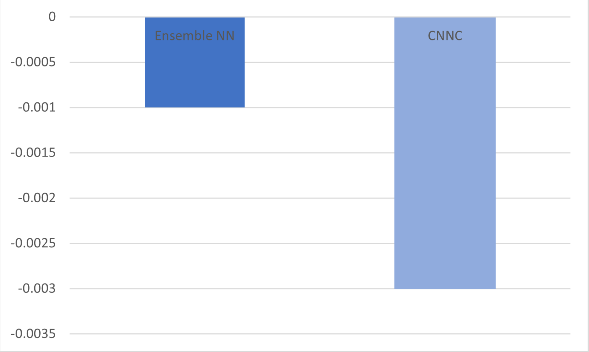


Figure 3.8: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

In the dataset with 500 class 1 instances, precision, recall, F1-score, and accuracy of CNNC were found to be lower than those of Ensemble NN and Best *k*-NN, as illustrated in Figures 13 and 14.



(a) Differences in precision, recall, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN



(b) Differences in F1-score, and between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN

Figure 3.9: Differences in results between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Circle format

  The precision, recall, F1-score, and accuracy of Best k-NN, Ensemble NN, and CNNC exhibited perfect scores of 1 when evaluated on circle format with no overlap.

3.2.1.2 Slight overlap

- Gaussian format In datasets with 100, 300, 400, and 500 class 1 instances, it is evident that the precision, recall, F1-score, and accuracy of CNNC are consistently lower than those of Ensemble NN, and both are inferior to Best k-NN. This comparison is depicted in Figures 3.10, 3.11, 3.12, and 3.13, respectively.

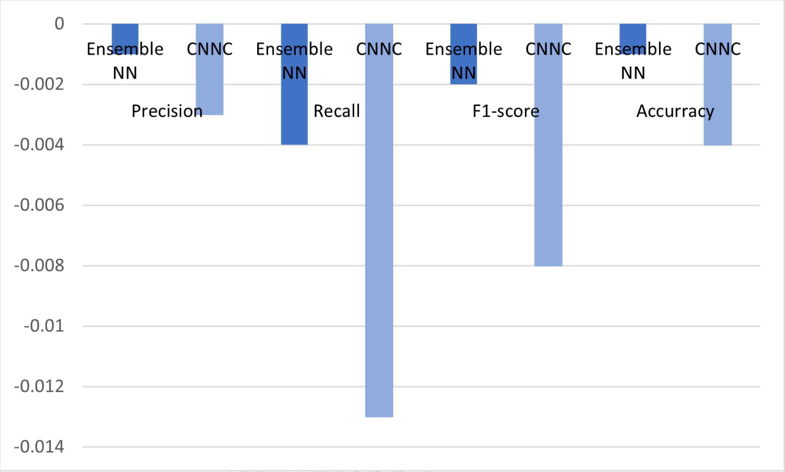


Figure 3.10: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

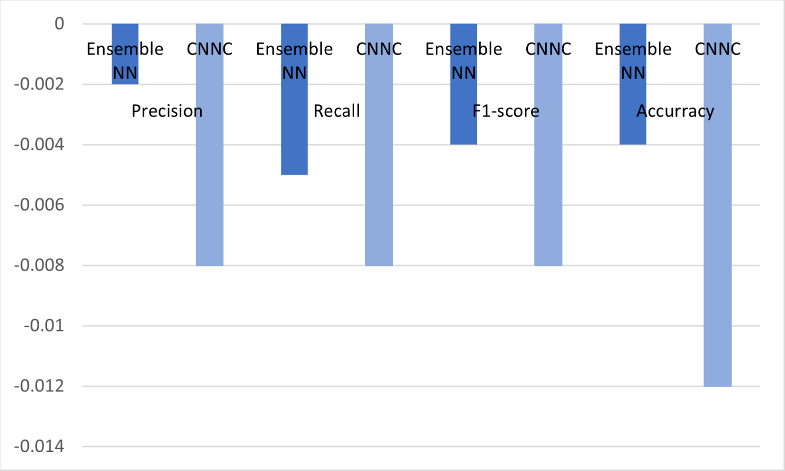


Figure 3.11: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

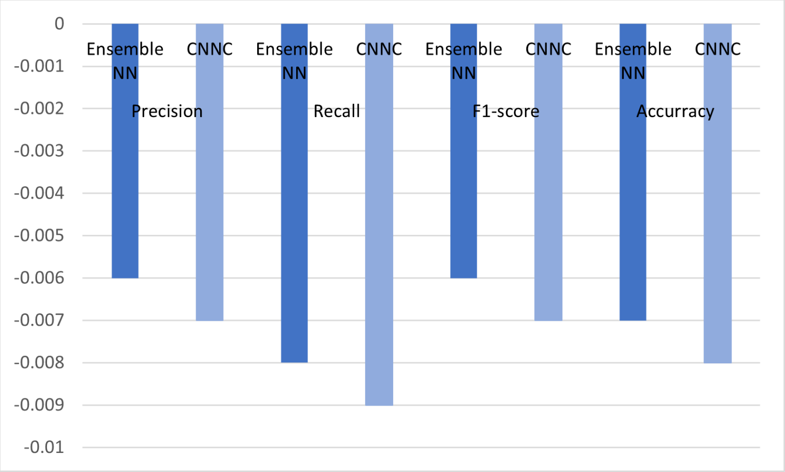


Figure 3.12: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

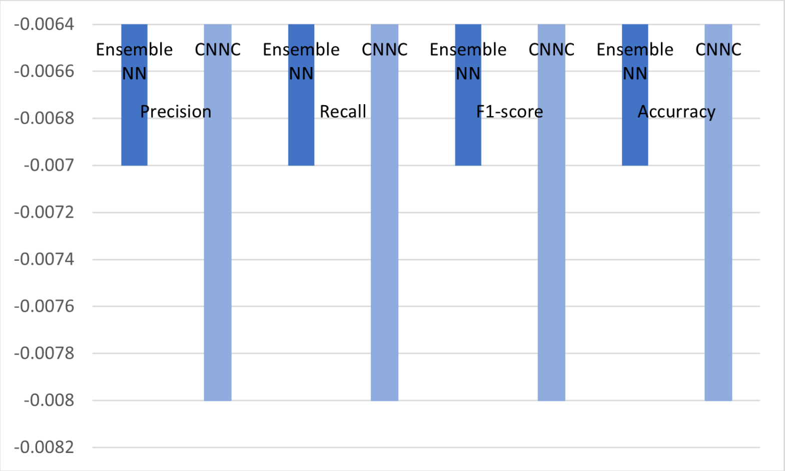


Figure 3.13: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

In the dataset featuring 200 class 1 instances, the precision, recall, and F1-score of CNNC are lower than those of Ensemble NN, and both are inferior to Best *k*-NN. Meanwhile, the accuracy of Ensemble NN surpasses that of Best *k*-NN, whereas the accuracy of CNNC is equal to that of Best *k*-NN, as depicted in Figure 3.14.

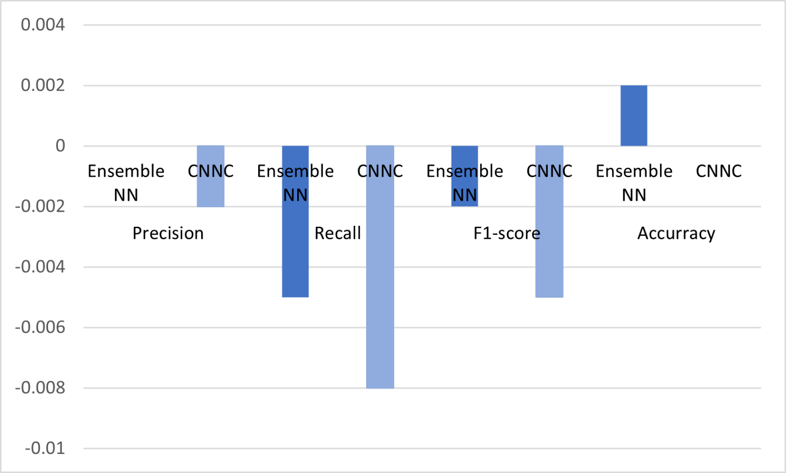


Figure 3.14: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 200 class 1 instances.

- Moon shaped format

  In datasets featuring 100, 300, and 500 class 1 instances, the precision, recall, F1-score, and accuracy of CNNC are higher than those of Ensemble NN but remain inferior to Best k-NN. This comparison is illustrated in Figure 3.15, 3.16, and 3.20, respectively.

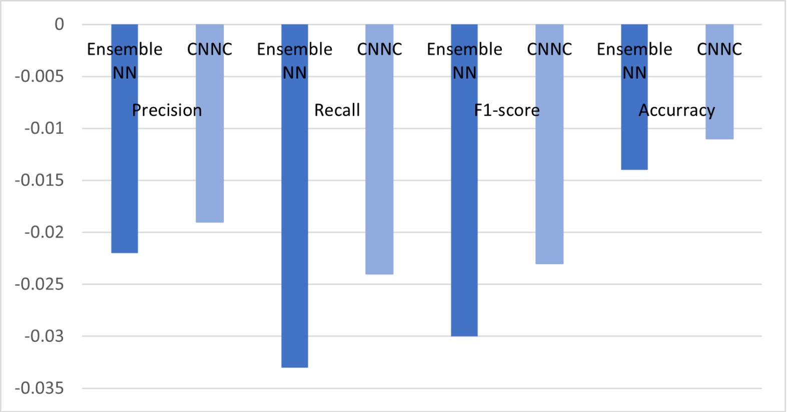


Figure 3.15: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 100 class 1 instances.

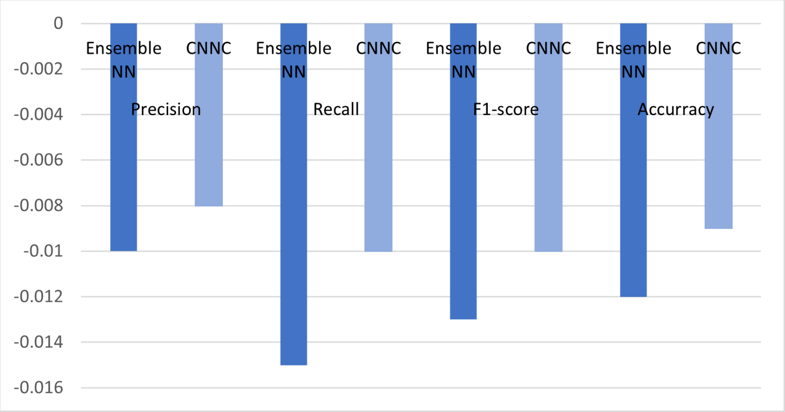


Figure 3.16: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 300 class 1 instances.

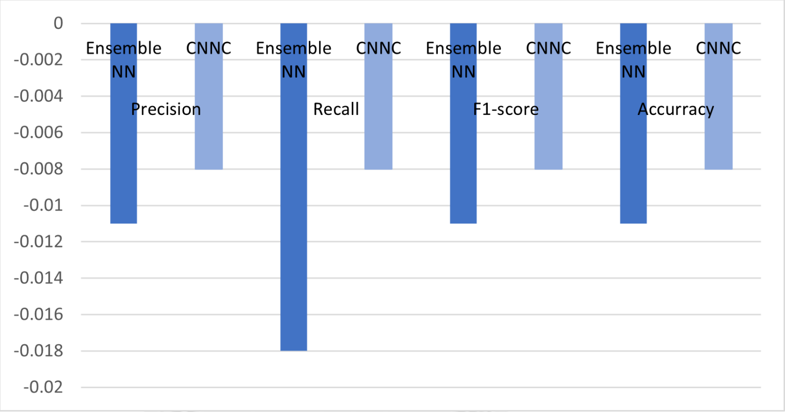


Figure 3.17: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 500 class 1 instances.

In the dataset with 200 class 1 instances, the precision, recall, F1-score, and accuracy of CNNC are found to be lower than those of Ensemble NN, and both are inferior to Best k-NN, as indicated in Figure 3.18.

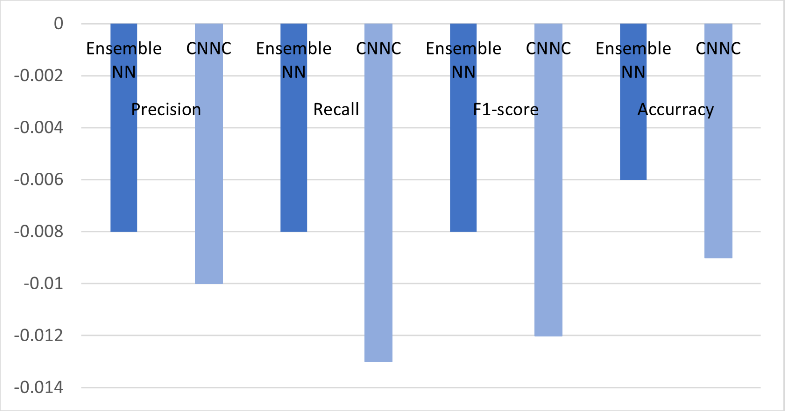


Figure 3.18: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

In the dataset featuring 400 class 1 instances, CNNC's precision, recall, F1-score, and accuracy are observed to be lower than Ensemble NN. While CNNC's recall is equal to the Ensemble NN. However, precision, recall, F1-score, and accuracy of CNNC and Ensemble NN are both lower than Best *k*-NN, as depicted in Figure 3.19.

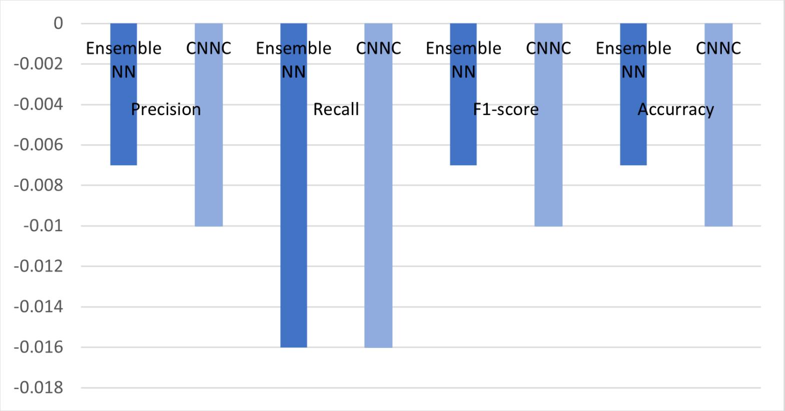


Figure 3.19: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

- Circle format

In datasets with class 1 instances numbering 100, 200, and 500, it is evident that the precision, recall, F1-score, and accuracy of CNNC are consistently lower than those of Ensemble NN. Additionally, both CNNC and Ensemble NN exhibit values that are inferior to Best k-NN, as depicted in Figures 3.20, 3.21, and 3.25.

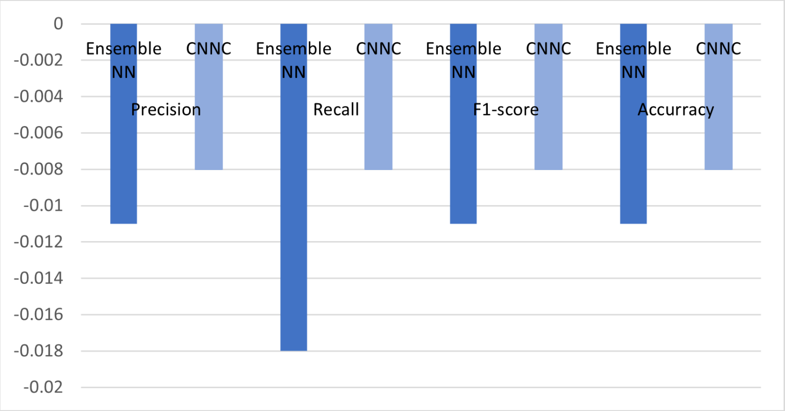


Figure 3.20: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 100 class 1 instances.

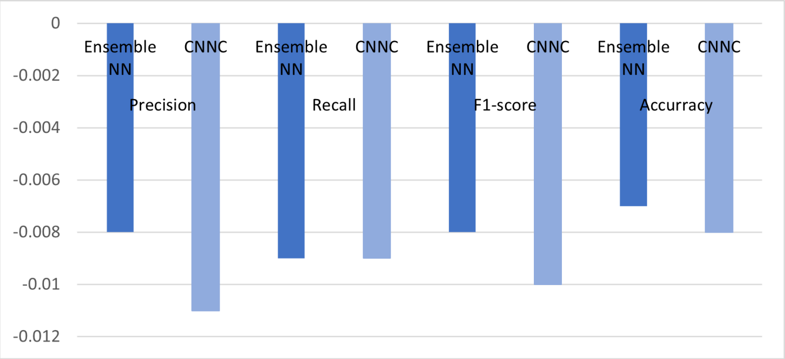


Figure 3.21: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 200 class 1 instances.

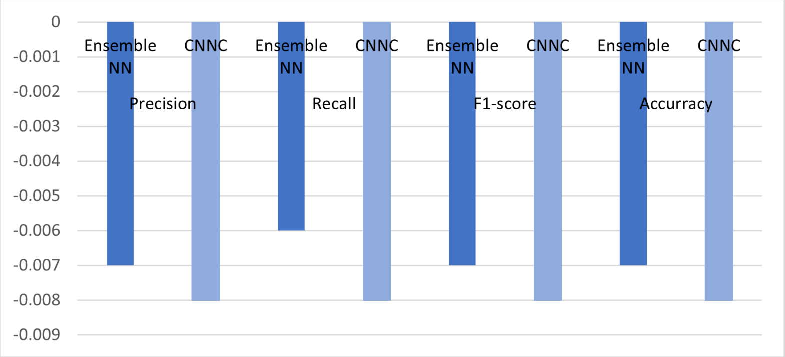


Figure 3.22: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 500 class 1 instances.

In the dataset with 300 instances of class 1, it is apparent that the precision, recall, F1-score, and accuracy of CNNC are equal to those of Ensemble NN. However, both CNNC and Ensemble NN display values lower than those of Best k-NN, as depicted in Figure 3.23.

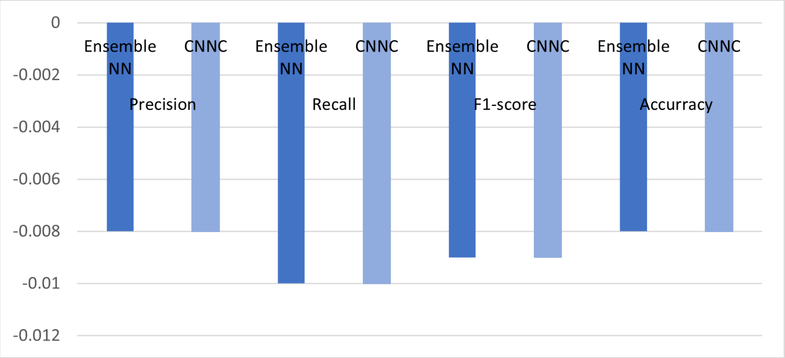


Figure 3.23: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 300 class 1 instances.

In the dataset with 400 instances of class 1, the precision, F1-score, and accuracy of CNNC were observed to be lower than those of Ensemble NN. Nevertheless, the recall values of CNNC and Ensemble NN were identical. However, both CNNC and Ensemble NN exhibited values lower than those of Best k-NN, as demonstrated in Figure 3.24.

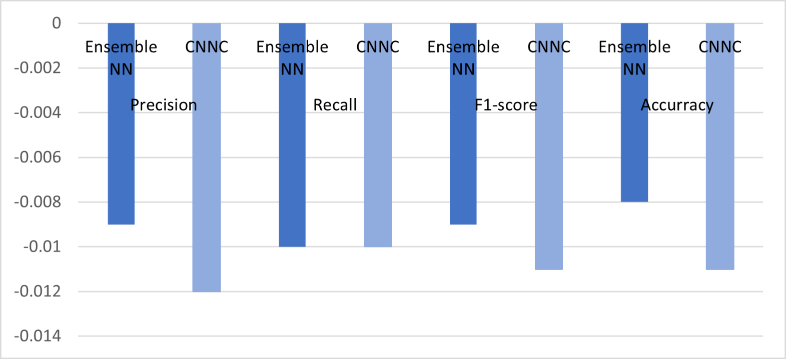


Figure 3.24: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 400 class 1 instances.

3.2.1.3 Large overlap

- Gaussian format

In the dataset with 100 instances of class 1, it is evident that the precision and accuracy of CNNC are lower than those of Ensemble NN. The recall of CNNC is higher than that of Ensemble NN, and the F1-score of CNNC is equal to Ensemble NN. However, both precision, recall, F1-score, and accuracy of both CNNC and Ensemble NN are lower than those of Best k-NN, as illustrated in Figure 3.25.

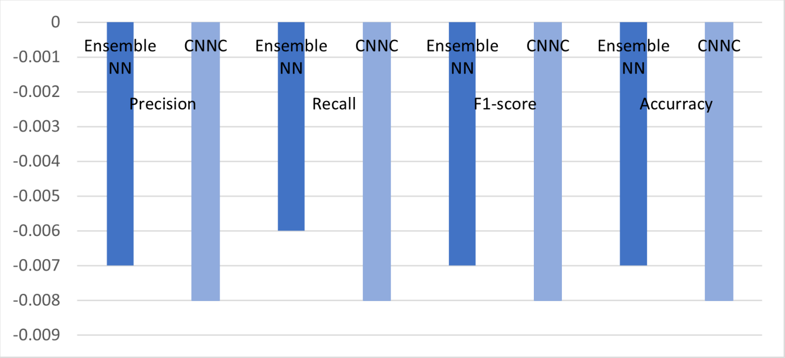


Figure 3.25: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 100 class 1 instances.

In the dataset featuring 200 instances of class 1, it is evident that the precision, F1-score, and accuracy of CNNC are lower than those of Ensemble NN. However, the recall of CNNC equals that of Ensemble NN. Yet, both precision, recall, F1-score, and accuracy of both CNNC and Ensemble NN are lower than those of Best k-NN, as illustrated in Figure 3.26.

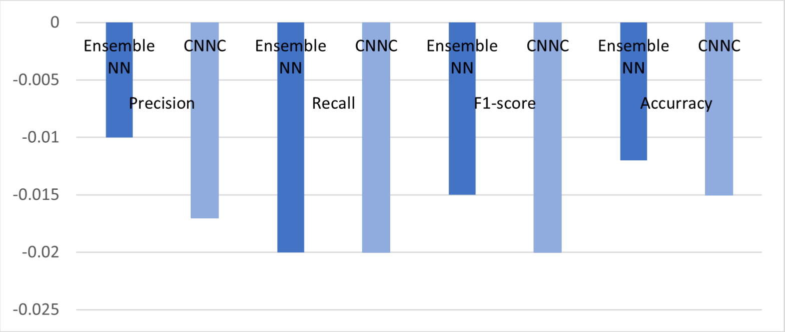


Figure 3.26: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 200 class 1 instances.

In datasets with 300, 400, and 500 instances of class 1, it is apparent that the precision, recall, F1-score, and accuracy of CNNC surpass those of Ensemble NN. However, both precision, recall, F1-score, and accuracy of both CNNC and Ensemble NN are lower than those of Best k-NN, as depicted in Figures

3.27, 3.28, and 3.29, respectively.

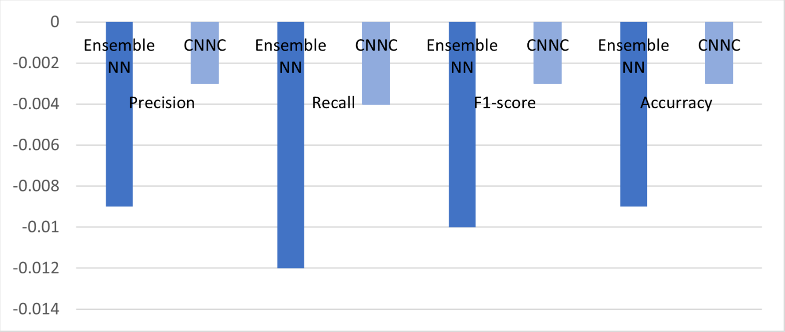


Figure 3.27: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

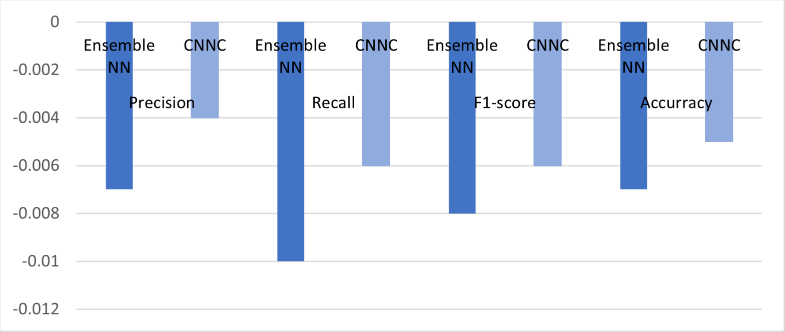


Figure 3.28: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

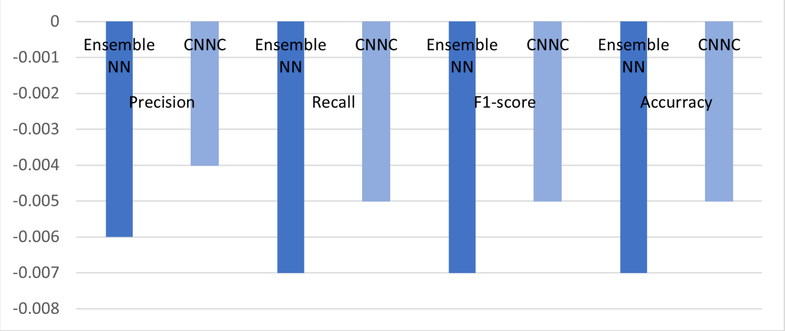


Figure 3.29: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best *k*-NN, as well as between CNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

- Moon shaped format

In the dataset with 100 instances of class 1, the precision and accuracy of CNNC are observed to be lower than those of Ensemble NN. However, the recall and F1-score of CNNC are higher than those of Ensemble NN. Nonetheless, all precision, recall, F1-score, and accuracy metrics for both CNNC and Ensemble NN are lower than those of Best k-NN, as demonstrated in Figure 3.30.

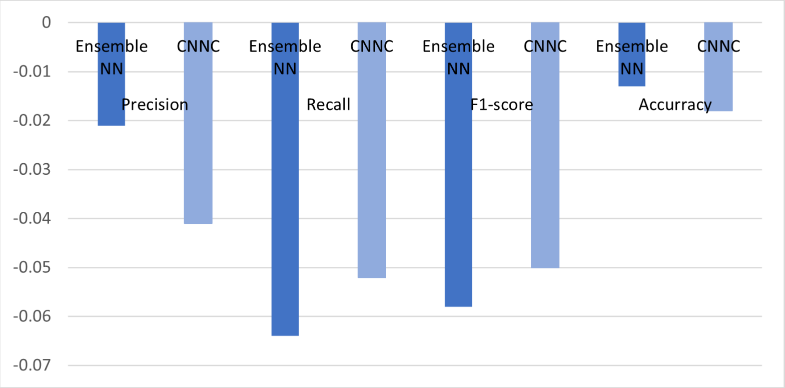


Figure 3.30: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 100 class 1 instances.

In the dataset with 200 instances of class 1, it is evident that the precision and F1-score of CNNC surpass both Best k-NN and Ensemble NN. Additionally, the accuracy of CNNC is equal to that of Best k-NN and higher than Ensemble NN. Meanwhile, the recall of CNNC is higher than Ensemble NN but lower than Best k-NN, as illustrated in Figure 3.31.

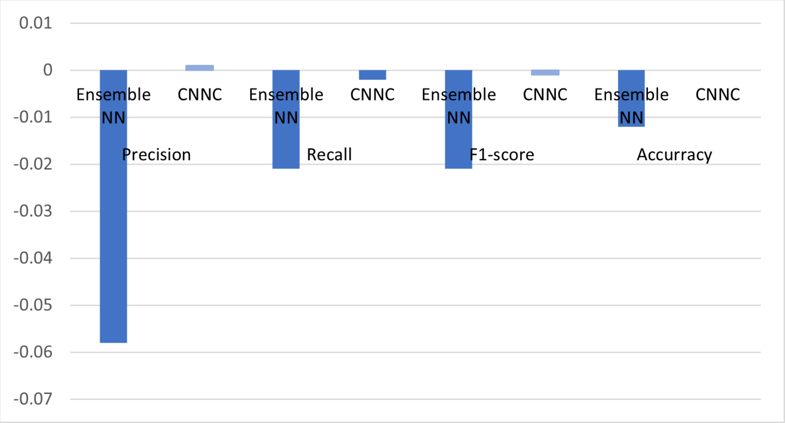


Figure 3.31: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 200 class 1 instances.

In the dataset with 300 instances of class 1, it is observed that the precision, recall, F1-score, and accuracy of CNNC are lower than those of Ensemble NN, and both are lower than those of Best k-NN, as depicted in Figure 3.32.

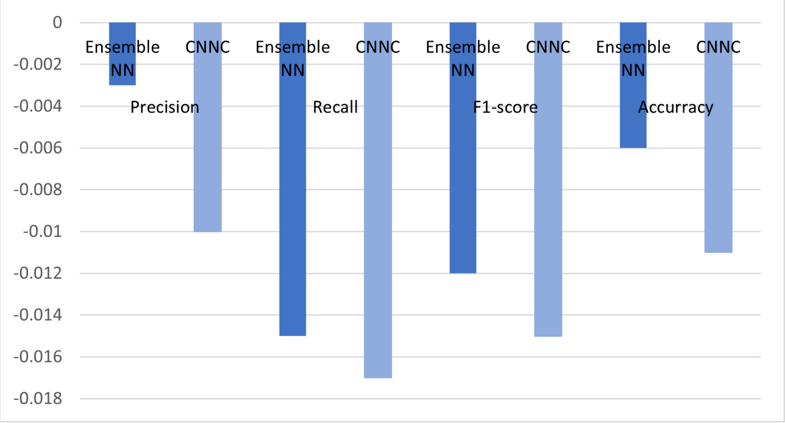


Figure 3.32: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 300 class 1 instances.

In datasets with 400 and 500 instances of class 1, it is observed that the precision, recall, F1-score, and accuracy of CNNC are higher than those of Ensemble NN. However, both are lower than those of Best k-NN, as depicted in Figures 3.33 and 3.34, respectively.

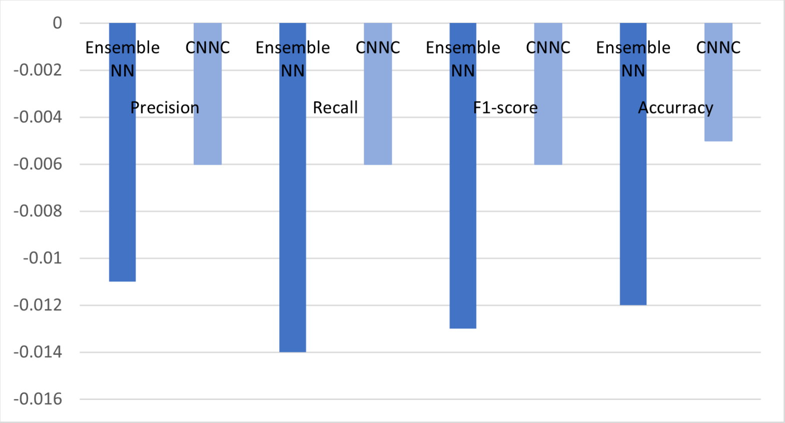


Figure 3.33: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 400 class 1 instances.

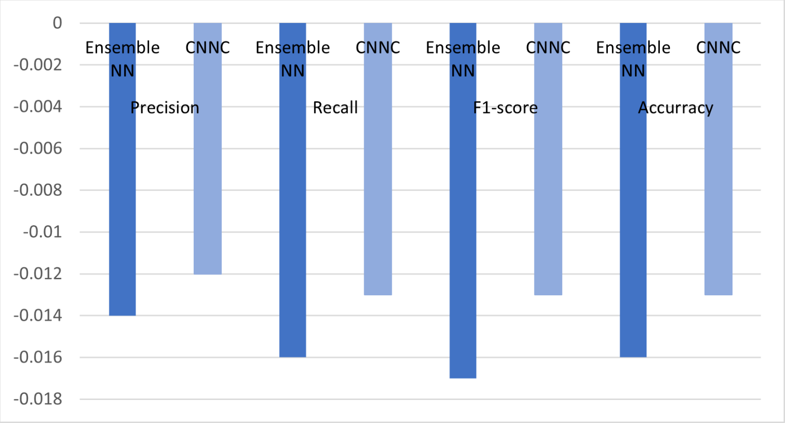


Figure 3.34: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Circle format

In the dataset with 100 instances of class 1, it is observed that the precision of CNNC is lower than that of Ensemble NN. The precision of Ensemble NN is higher than that of Best k-NN. Moreover, the recall, F1-score, and accuracy of CNNC are lower than those of Ensemble NN, and both are lower than those of Best k-NN, as depicted in Figure 3.35.

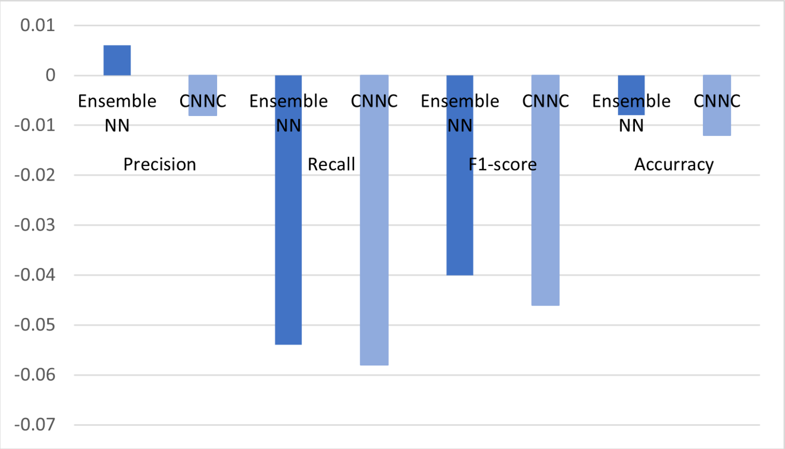


Figure 3.35: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 100 class 1 instances.

In the dataset with 200 and 500 instances of class 1, it is observed that the precision, recall, F1-score, and accuracy of CNNC are lower than those of Ensemble NN, and both are lower than those of Best k-NN, as depicted in Figures 3.36 and 3.37, respectively.

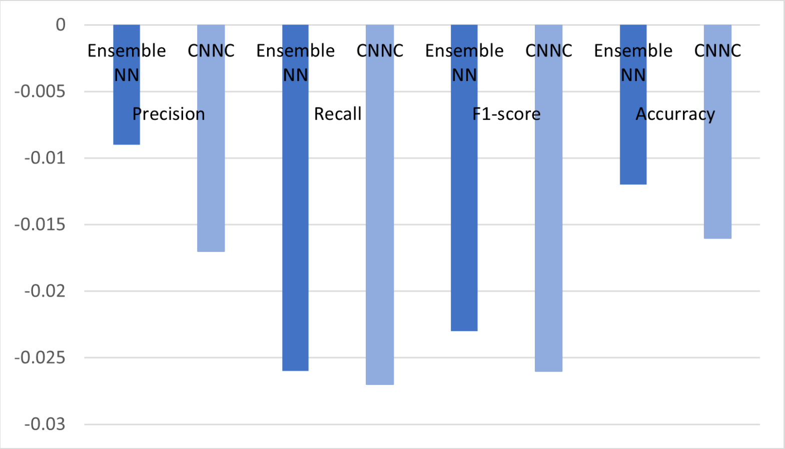


Figure 3.36: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 200 class 1 instances.

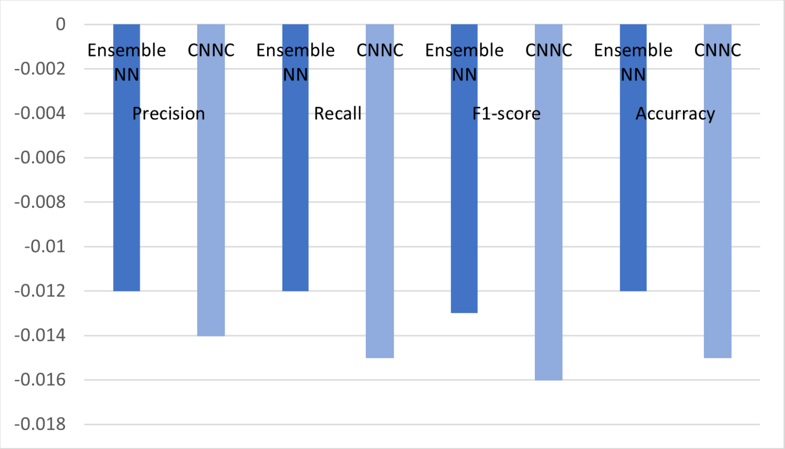


Figure 3.37: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 500 class 1 instances.

In the dataset with 300 instances of class 1, it is observed that the precision and accuracy of CNNC are lower than those of Ensemble NN. However, the F1-score of CNNC is equal to that of Ensemble NN, and the recall of CNNC is higher than that of Ensemble NN. Nevertheless, precision, recall, F1-score, and accuracy of both CNNC and Ensemble NN are lower than those of Best k-NN, as shown in Figure 3.38.

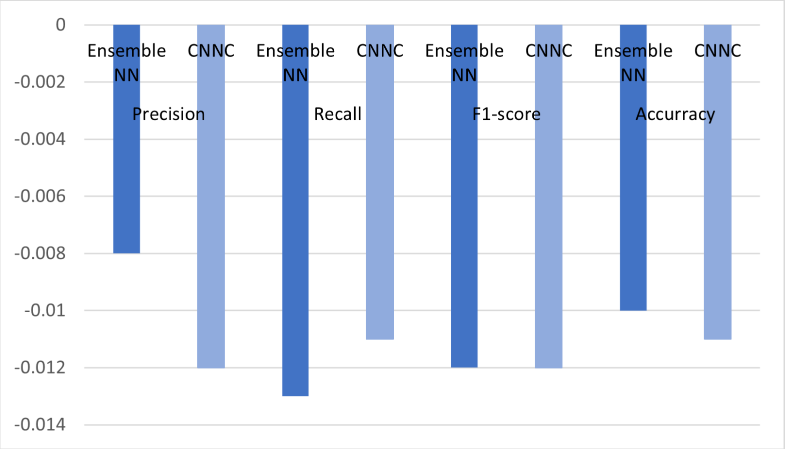


Figure 3.38: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 300 class 1 instances.

In datasets with 400 instances of class 1, it is observed that the precision, recall, F1-score, and accuracy of CNNC are higher than those of Ensemble NN. However, both are lower than those of Best k-NN, as depicted in Figure 3.39.

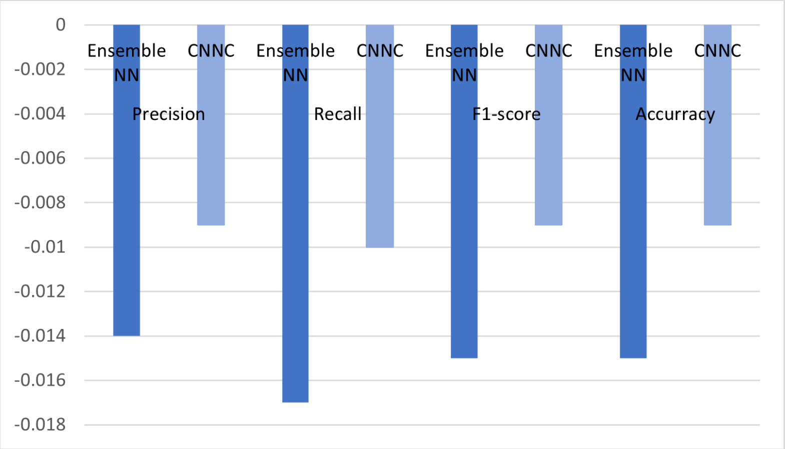


Figure 3.39: Differences in precision, recall, F1-score, and accuracy between the Ensemble NN and Best k-NN, as well as between CNNC and Best k-NN for a dataset containing 400 class 1 instances.

3.2.2 UCI Dataset

In this subsection, the outcomes of the experiments conducted on the UCI dataset by each classifier are outlined across four tables. Specifically, Table 3.1 showcases Precision, Table 3.2 features Recall, Table 3.3 presents the F1-score, and Table 3.4 displays accuracy. The values within each table represent the mean ± standard deviation, derived from 10 repetitions of the experiment.

Data	Best k-NN	The ensemble NN	CNNC
Wine	0.93±0.024	0.918±0.026	0.899±0.047
Sonar	0.828±0.042	0.768±0.044	0.685±0.043
Glass	0.853±0.085	0.827±0.084	0.776±0.16
Haberman	0.69±0.044	0.641±0.1	0.65±0.065
Liver	0.699±0.049	0.684±0.063	0.659±0.065
Ionosphere	0.899±0.024	0.873±0.023	0.879±0.026
Wholesale	0.894±0.034	0.876±0.037	0.877±0.04
Cancer	0.971±0.014	0.964±0.015	0.962±0.015
German	0.639±0.045	0.643±0.071	0.617±0.062
QSAR	0.804±0.019	0.778±0.026	0.747±0.031

Table 3.1: The Precision of the CNNC compared to other classifiers in UCI datasets.

Data	Best k-NN	The ensemble NN	CNNC
Wine	0.924±0.039	0.898±0.035	0.893±0.043
Sonar	0.823±0.038	0.738±0.04	0.669±0.045
Glass	0.831±0.065	0.748±0.042	0.717±0.041
Haberman	0.624±0.019	0.57±0.035	0.584±0.019
Liver	0.683±0.042	0.656±0.06	0.645±0.061
Ionosphere	0.834±0.035	0.773±0.027	0.785±0.032
Wholesale	0.899±0.029	0.877±0.027	0.877±0.03
Cancer	0.973±0.012	0.959±0.015	0.958±0.014
German	0.613±0.036	0.542±0.024	0.54±0.02
QSAR	0.816±0.02	0.789±0.025	0.765±0.031

Table 3.2: The recall of the CNNC compared to other classifiers in UCI datasets.

Data	Best *k*-NN	The ensemble NN	CNNC
Wine	0.924±0.031	0.904±0.025	0.893±0.04
Sonar	0.821±0.041	0.733±0.046	0.662±0.051
Glass	0.839±0.066	0.761±0.053	0.727±0.061
Haberman	0.632±0.022	0.565±0.059	0.584±0.027
Liver	0.682±0.041	0.654±0.066	0.643±0.063
Ionosphere	0.853±0.032	0.793±0.034	0.806±0.034
Wholesale	0.895±0.03	0.875±0.032	0.876±0.034
Cancer	0.972±0.012	0.961±0.015	0.96±0.014
German	0.615±0.038	0.511±0.036	0.514±0.031
QSAR	0.808±0.019	0.781±0.026	0.751±0.031

Table 3.3: The F1-score of the CNNC compared to other classifiers in UCI datasets.

Data	Best *k*-NN	The ensemble NN	CNNC
Wine	0.935±0.019	0.918±0.016	0.908±0.032
Sonar	0.824±0.037	0.745±0.044	0.674±0.051
Glass	0.921±0.032	0.92±0.024	0.921±0.022
Haberman	0.759±0.027	0.742±0.035	0.744±0.029
Liver	0.702±0.041	0.685±0.056	0.666±0.057
Ionosphere	0.876±0.026	0.834±0.026	0.842±0.028
Wholesale	0.909±0.026	0.892±0.028	0.893±0.031
Cancer	0.975±0.011	0.965±0.013	0.964±0.013
German	0.699±0.039	0.706±0.03	0.7±0.024
QSAR	0.825±0.019	0.8±0.024	0.768±0.029

Table 3.4: The accuracy of the CNNC compared to other classifiers in UCI datasets.

As evident from the data presented in Table 3.1-3.4, our method demonstrates superior accuracy compared to both the bestk-NN, and the ensemble NN in the Glass dataset. It also outperforms the ensemble NN in the Harberman dataset and the Ionosphere dataset. Our method exhibits higher precision and recall than the ensemble NN in the Harberman dataset, Ionosphere dataset, and Wholesale dataset. Moreover, our method achieves a higher F1-score than the ensemble NN across several datasets, including the Harberman dataset, Ionosphere dataset, Wholesale dataset, and German dataset. This could be attributed to the arrangement of these datasets in edge-like structures, enhancing the efficiency of MOFs in approximating the location of instance. Notably, there is not a significant distinction between the performance of the Best k-NN and the ensemble NN.

3.3 The discussion of the conglomerate nearest neighbor classifier

This chapter introduces the conglomerate nearest neighbor algorithm, which operates without requiring any specific parameters. The assignment of different nearest neighbors for each instance is based on Mass-ratio-variance Outlier Factors (MOF), which adapts to the density of instances in a dataset. In experiments conducted with synthesized datasets, this algorithm demonstrates similar performance to the traditional k-NN and the ensemble NN approaches.

For two-class synthesized datasets, including the k-NN algorithm, the ensemble NN algorithm, and the conglomerate nearest neighbor algorithm, all three exhibit comparable precision, recall, F1-score and accuracy when classifying unknown instances in the testing set. However, the conglomerate nearest neighbor consistently lags behind the performance of the k-NN, irrespective of whether the dataset exhibits issues related to overlapping or class imbalance.

When applied to real-world datasets, the conglomerate nearest neighbor algorithm effectively predicts the class of unknown instances, performing at a comparable level to the k-NN. It is worth noting that in specific datasets, such as the German dataset and the Glass dataset, the conglomerate nearest neighbor outperforms the k-NN in terms of accuracy. Additionally, the conglomerate nearest neighbor achieves a higher F1-score than the ensemble NN in various datasets, including the Harberman dataset, Ionosphere dataset, Wholesale dataset, and German dataset.

What sets the conglomerate nearest neighbor apart is its ability to deliver comparable performance to the k-NN without the need for fine-tuning specific parameters. In contrast, the k-NN may require parameter optimization for optimal results. Furthermore, the conglomerate nearest neighbor algorithm demonstrates similar performance to the ensemble NN.

CHAPTER IV

MOF-GUIDED CONGLOMERATE NEAREST NEIGHBOR CLASSIFIER

This chapter provides an overview of the classifier based on MOF (Mass-ratio-variance Outlier Factors), encompassing detailed algorithms and pseudocode.

4.1 MOF-guided conglomerate nearest neighbor classifier (MNNC) algorithm

MNNC's method bears resemblance to kNN, necessitating the storage of training instances for subsequent retrieval during the testing phase. During testing, MNNC dynamically determines the number of neighbors for each test instance based on MOF. Upon the arrival of a new test instance, MOF is computed, leading to three distinct cases:

In Case 1, when MOF is greater than or equal to 1, a single neighbor is employed. This choice is based on the understanding that a high MOF value indicates that the test instance is distantly positioned from other clusters. Opting for a small number of neighbors in this scenario helps mitigate the risk of inaccurate predictions.

In Case 2, if MOF falls within the range [a, 1), the number of neighbors is configured as $\frac{\sqrt{n}}{2}$, where 'a' takes values from the range 0.01 to 0.7, and 'n' signifies the number of instances in the training set. This adjustment is made considering the likelihood that a test instance resides on the periphery of one of the clusters.

By utilizing a number of neighbors greater than 1 but not excessively high, more effective predictions can be achieved.

In Case 3, when MOF is less than ‘a’, the number of neighbors is set to $\sqrt{n}$. This choice is motivated by the scenario where the test instance is positioned within cluster. Here, using a larger number of neighbors proves beneficial for enhancing the accuracy of predictions.

Each specific value of ‘a’ is denoted as follows: When a = 0.1, it is referred to as MNNC(1); when a = 0.3, it is denoted as MNNC(2); when a = 0.5, it is labeled as MNNC(3); when a = 0.7, it is designated MNNC(4); when a = 0.01, it is named MNNC(5); when a = 0.03, it is assigned the label MNNC(6); when a = 0.05, it is termed MNNC(7); and when a = 0.07, it is recognized as MNNC(8).

4.2 Experimental results of the MOF-guided conglomerate nearest neighbor classifier

The performance of both the *k*-NN algorithm and the ensemble *k*-NN algorithm will be compared using the precision, recall, F1-score, and accuracy obtained from both the synthesized datasets and UCI datasets, to determine which algorithm outperforms the other.

4.2.1 Synthesized Dataset

Similar to Chapter 3, numeric outcomes can be found in Appendices B, C, D, and E. Precision is depicted in Appendix B, recall in Appendix C, F1-score in Appendix D, and accuracy in Appendix E.

4.2.1.1 Precision

- No overlap

- Gaussian format

For datasets comprising 100, 200, 300, and 400 instances of class 1, the precision values for all iterations of ensemble NN and MNNC are consistent with the Best k-NN, both registering a precision of 1. However, when the dataset size is 500 instances of class 1, only ensemble NN and MNNC(6) demonstrate precision values equal to the Best k-NN. The remaining versions of MNNC exhibit lower precision values compared to Best k-NN and ensemble NN, as depicted in Figure 4.1.

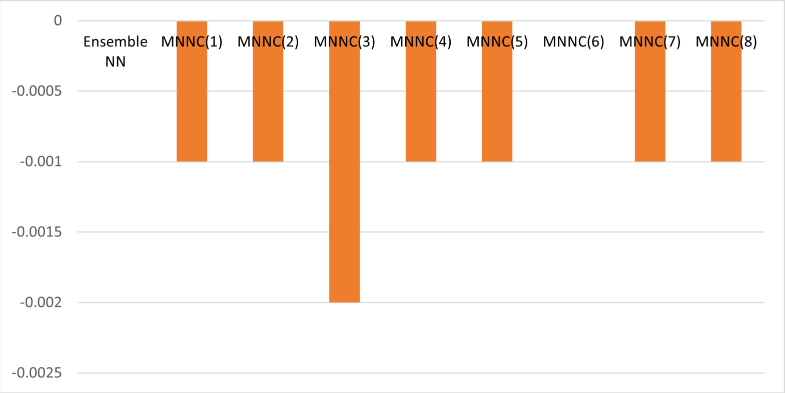


Figure 4.1: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Moon shaped format

For datasets with 100 and 500 instances of class 1, the precision of both ensemble NN and all versions of MNNC is lower than that of Best k-NN, as illustrated in Figures 4.2 and 4.3, respectively.

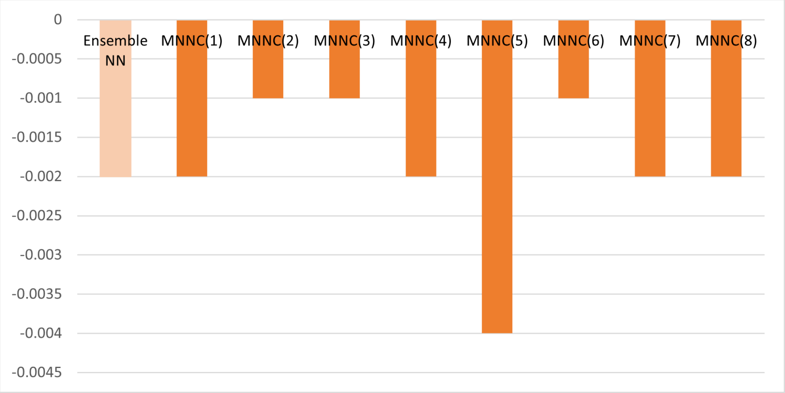


Figure 4.2: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

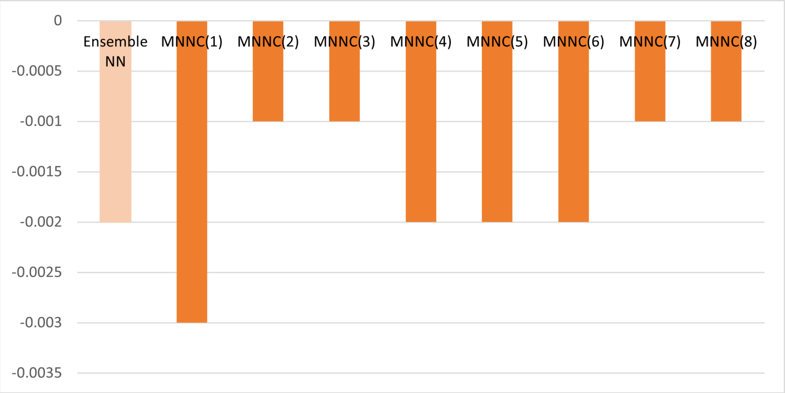


Figure 4.3: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

For datasets comprising 200 instances of class 1, the precision values for ensemble NN, MNNC(2), MNNC(3), MNNC(4), MNNC(7), and MNNC(8) are equivalent to that of Best k-NN, while the precision of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.4.

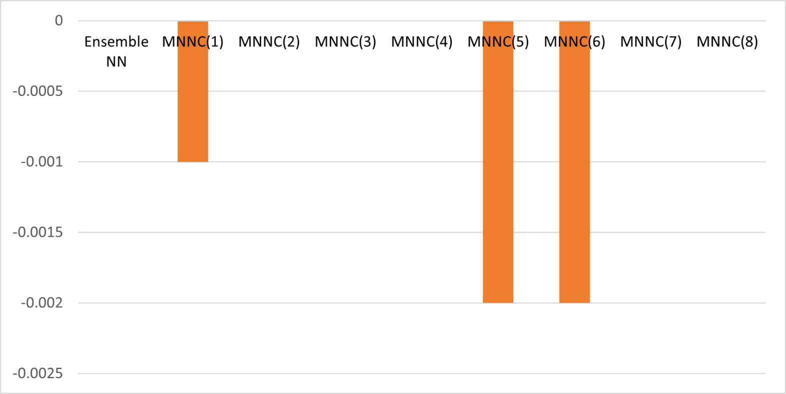


Figure 4.4: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

For datasets comprising 300 instances of class 1, the precision values for ensemble NN, MNNC(2)and MNNC(6) are equivalent to that of Best *k*-NN, while the precision of the remaining versions of MNNC is lower than Best *k*-NN, as illustrated in Figure 4.5.

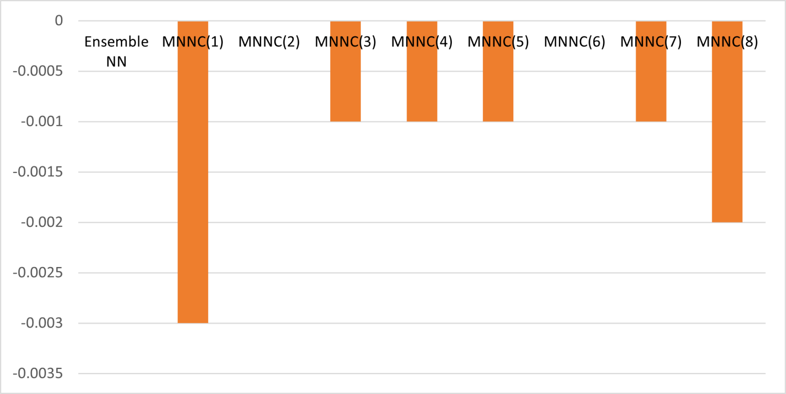


Figure 4.5: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

For datasets comprising 400 instances of class 1, the precision values for ensemble NN, MNNC(3), MNNC(4), and MNNC(8) are equivalent to that of Best *k*-NN, while the precision of the remaining versions of

MNNC is lower than Best k-NN, as illustrated in Figure 4.6.

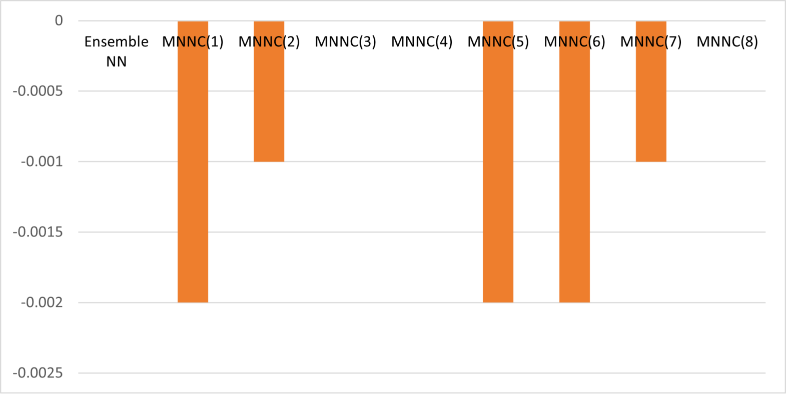


Figure 4.6: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

- Circle format

  The precision of Best k-NN, Ensemble NN, and all MNNC exhibited perfect scores of 1 when evaluated on circle format with no overlap.

- Slight overlap

  - Gaussian format

    The precision of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.7 to 4.11, respectively.

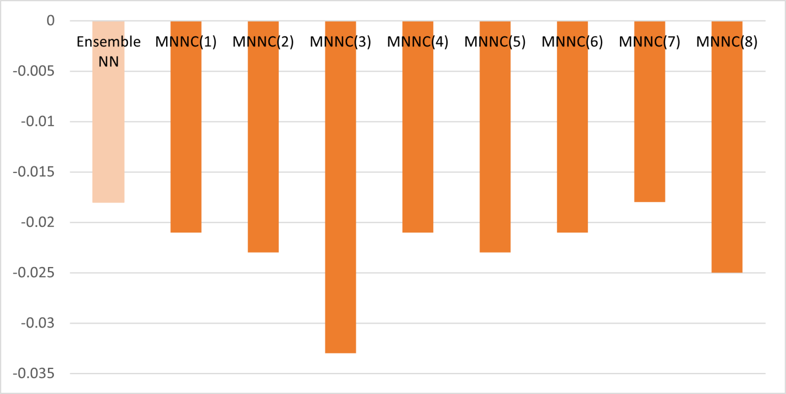


Figure 4.7: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

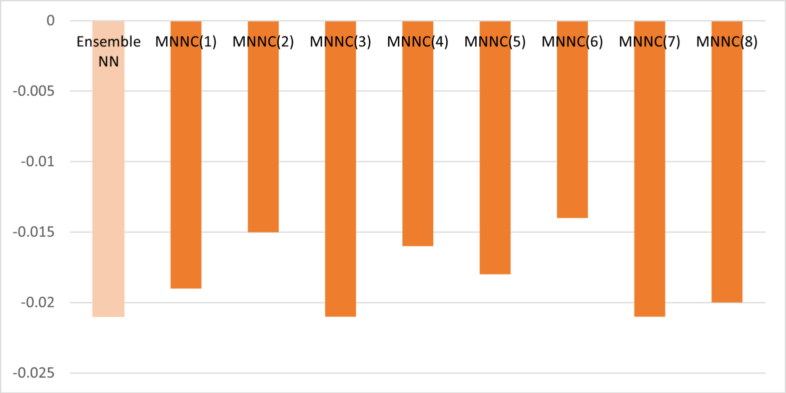


Figure 4.8: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

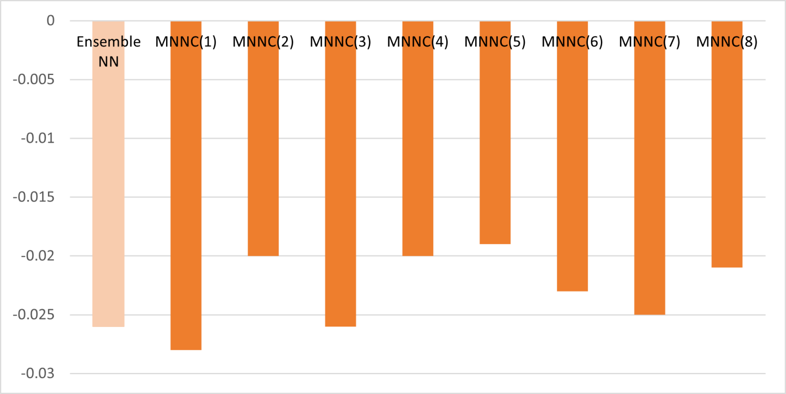


Figure 4.9: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

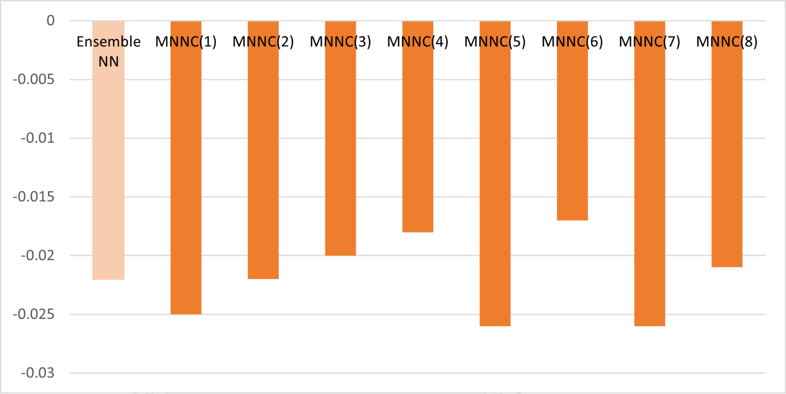


Figure 4.10: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

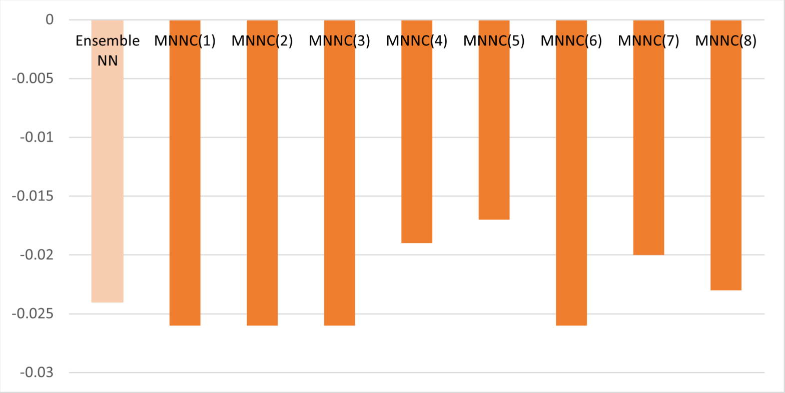


Figure 4.11: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Moon shaped format

  The precision of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.12 to 4.16, respectively.

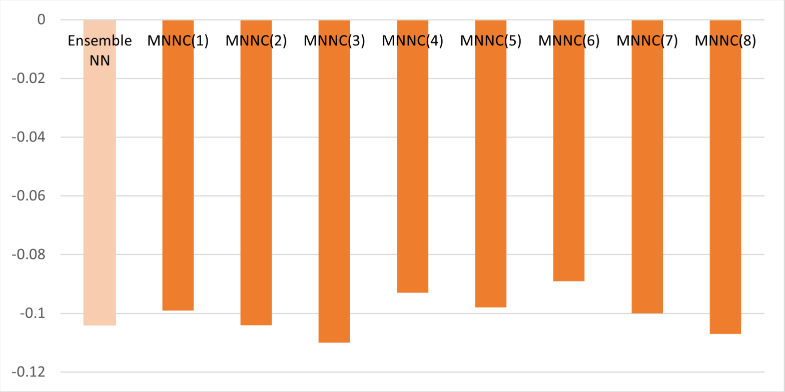


Figure 4.12: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

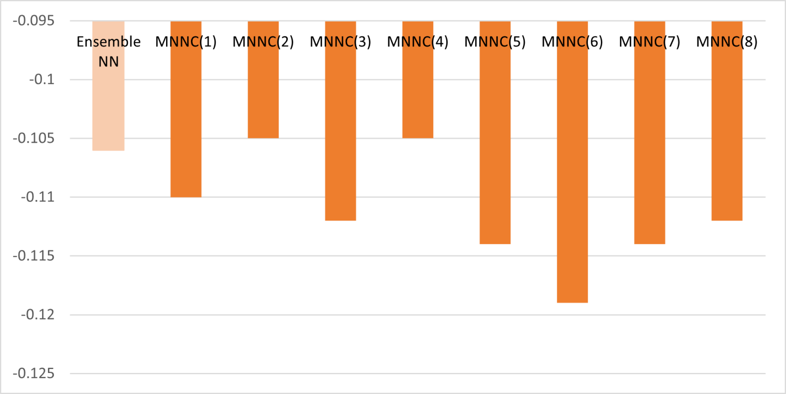


Figure 4.13: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

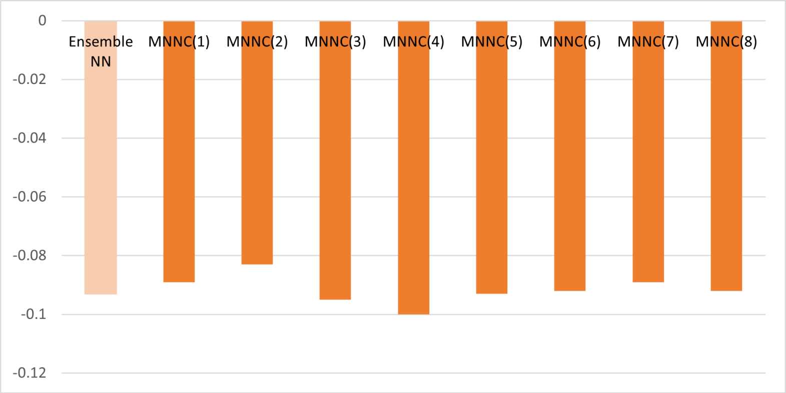


Figure 4.14: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

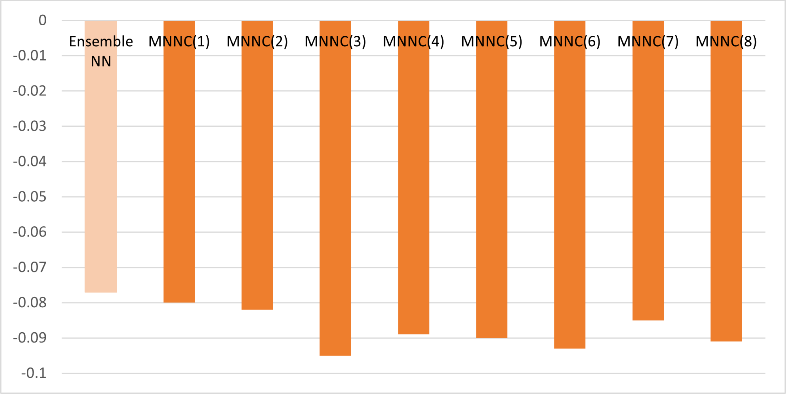


Figure 4.15: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

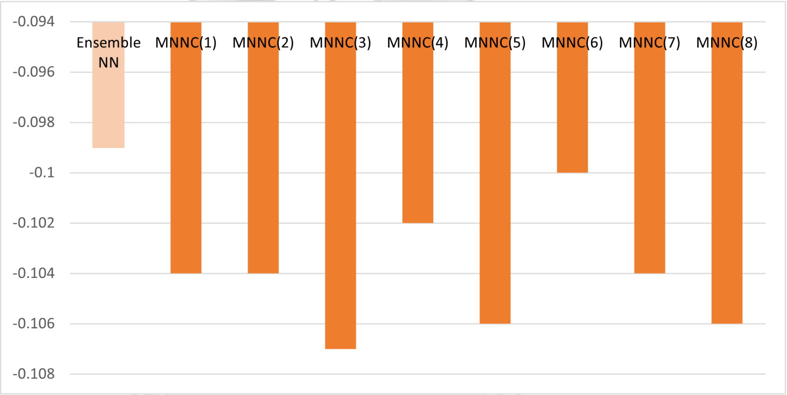


Figure 4.16: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Circle format

The precision of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.17 to 4.21, respectively.

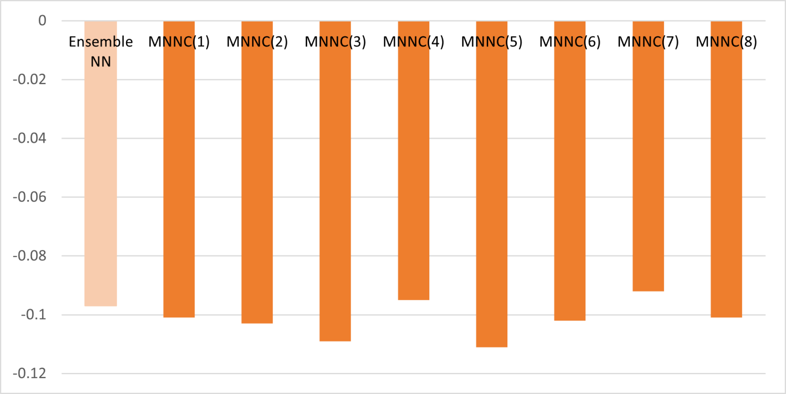


Figure 4.17: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

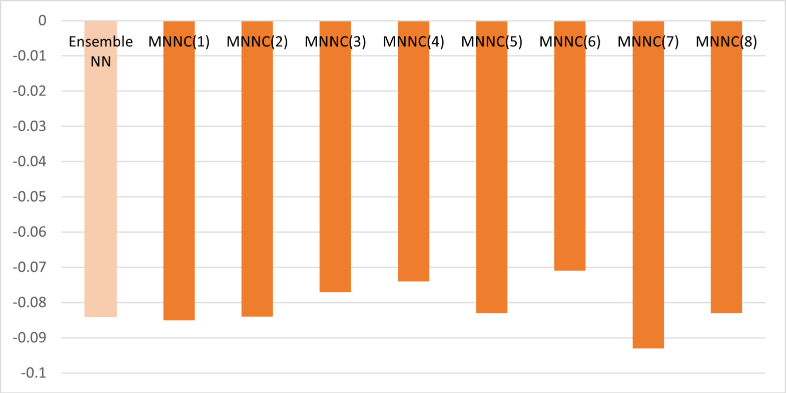


Figure 4.18: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

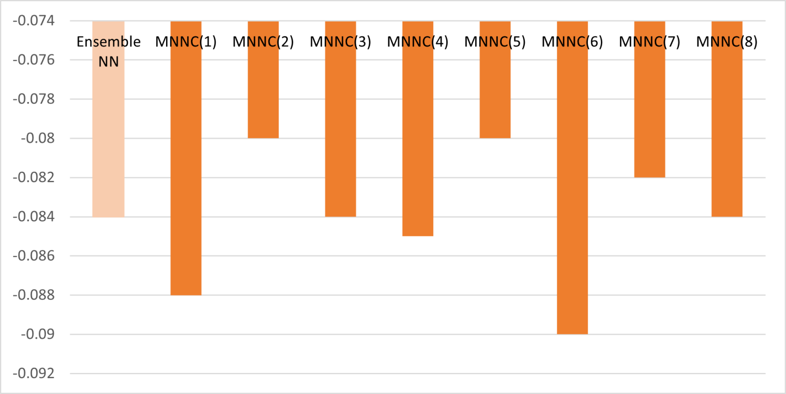


Figure 4.19: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

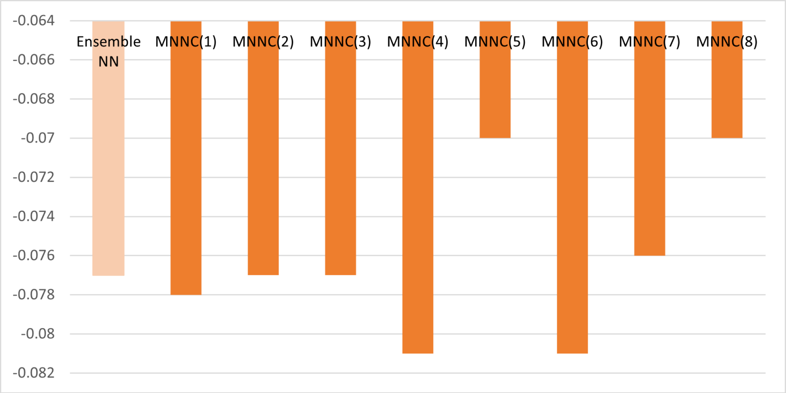


Figure 4.20: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

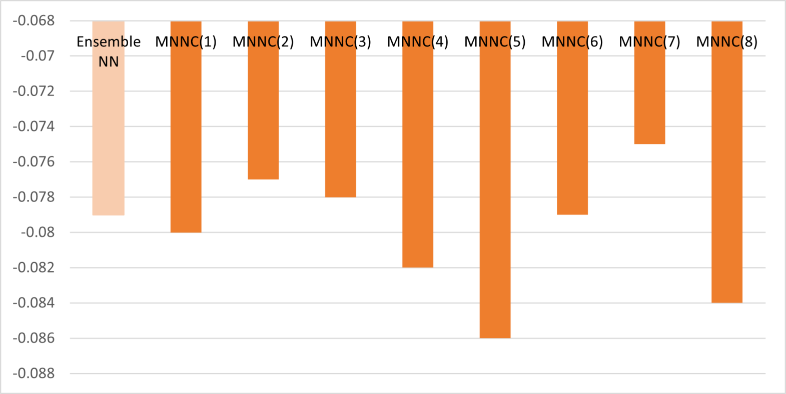


Figure 4.21: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

- Large overlap
  - Gaussian format

    The precision of both ensemble NN and all versions of MNNC is consistently lower than that of Best *k*-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.22 to 4.26, respectively.

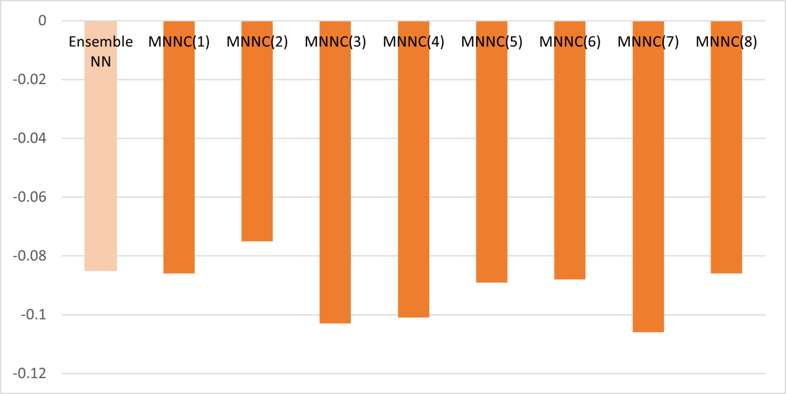


Figure 4.22: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

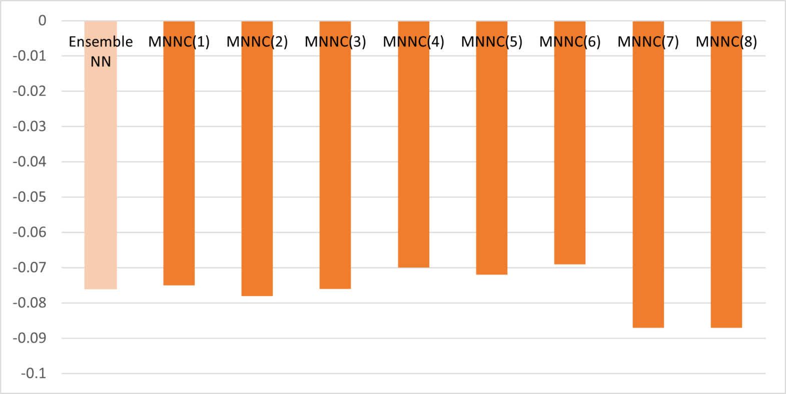


Figure 4.23: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

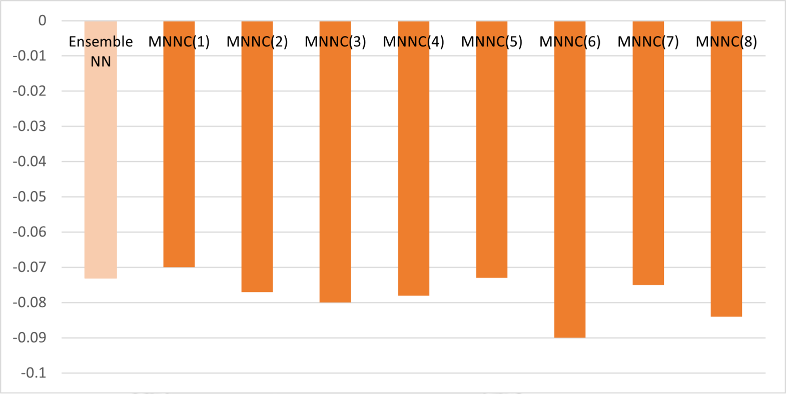


Figure 4.24: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

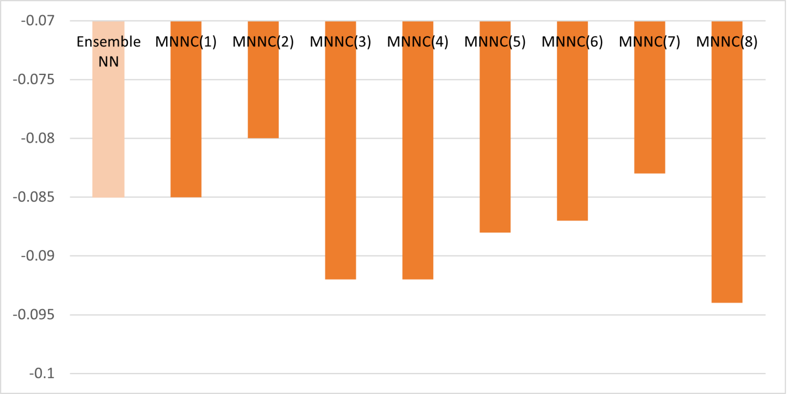


Figure 4.25: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

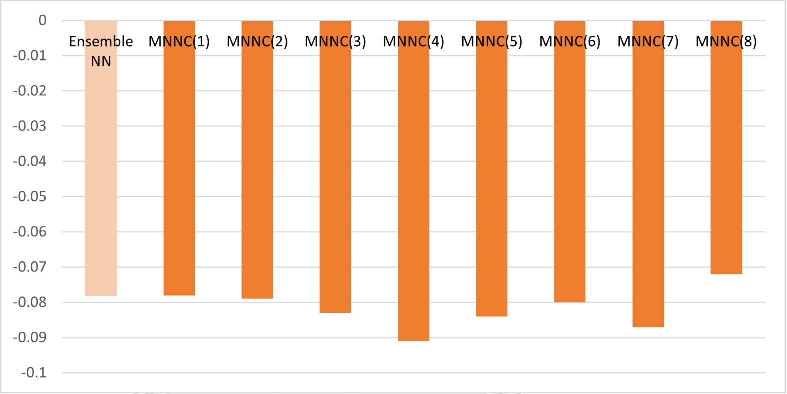


Figure 4.26: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

The precision of both ensemble NN and all versions of MNNC is consistently lower than that of Best *k*-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.27 to 4.31, respectively.

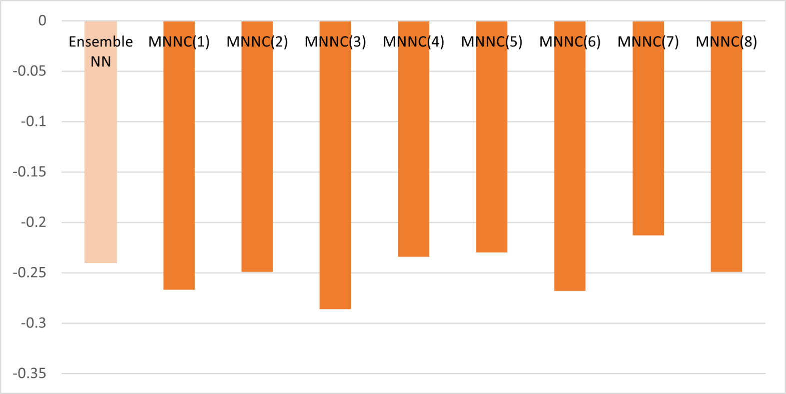


Figure 4.27: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

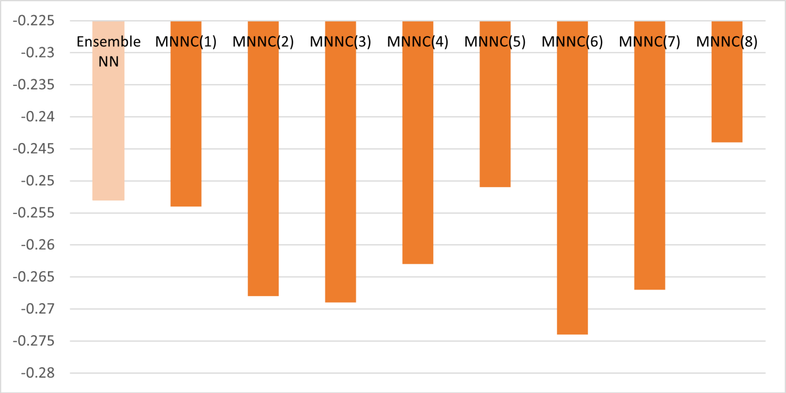


Figure 4.28: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

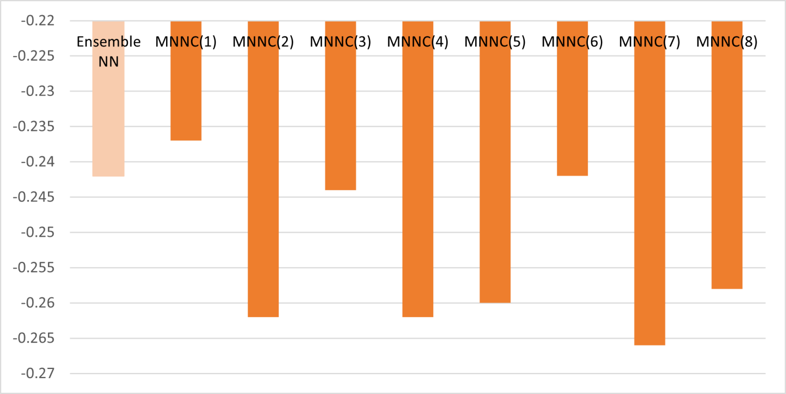


Figure 4.29: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

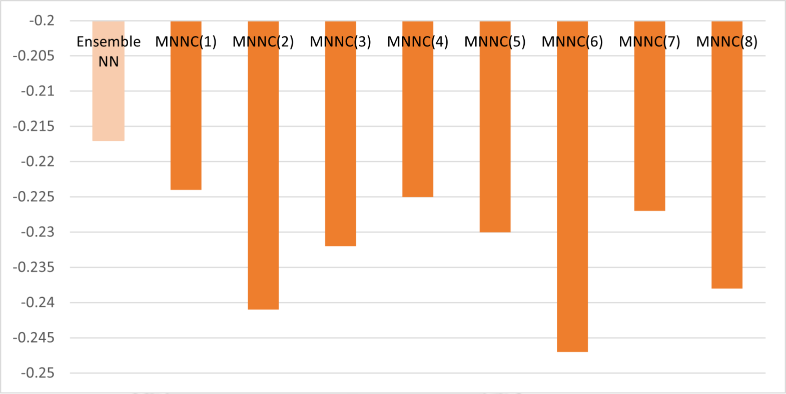


Figure 4.30: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

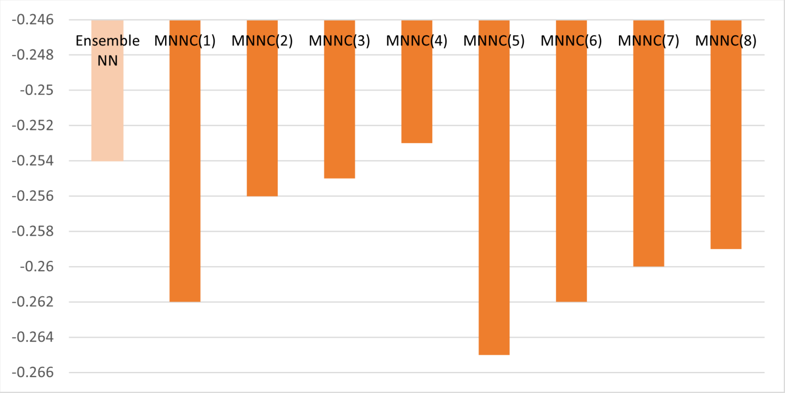


Figure 4.31: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Circle format

  The precision of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.32 to 4.36, respectively.

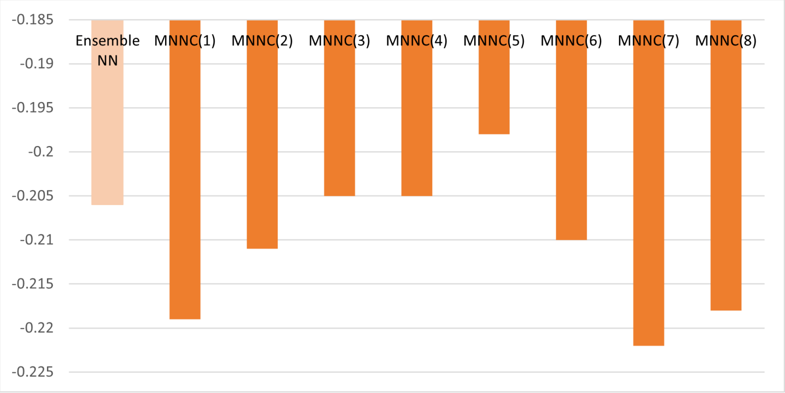


Figure 4.32: Differences in precision, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

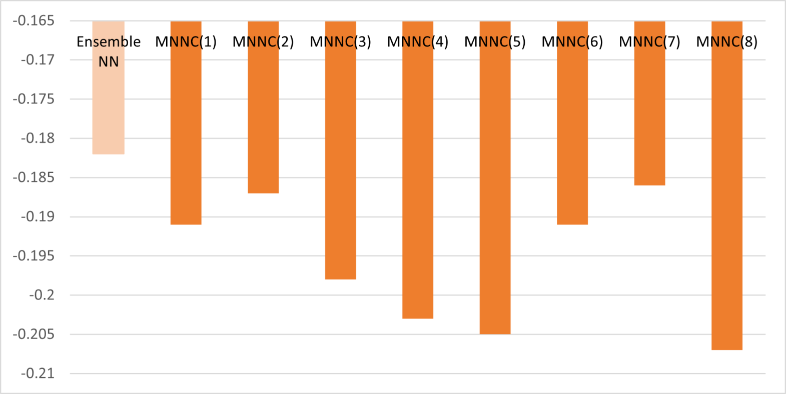


Figure 4.33: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

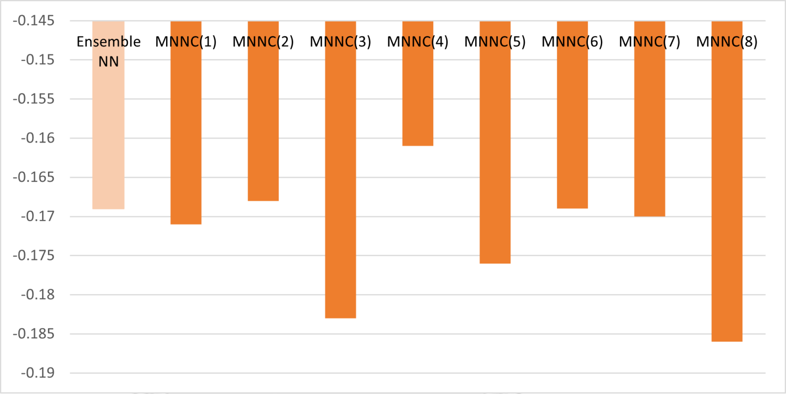


Figure 4.34: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

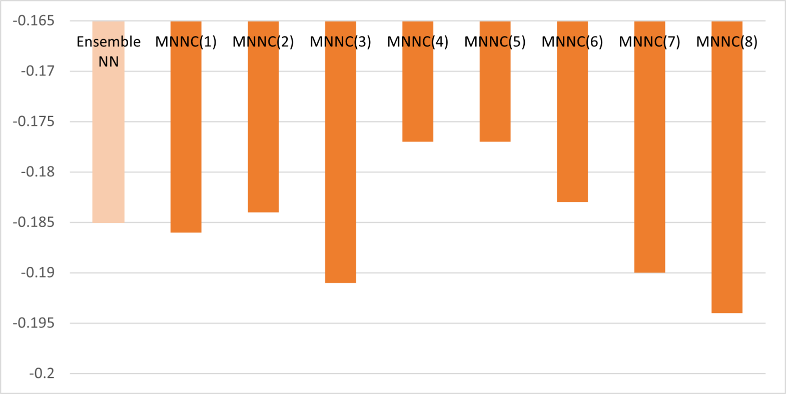


Figure 4.35: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

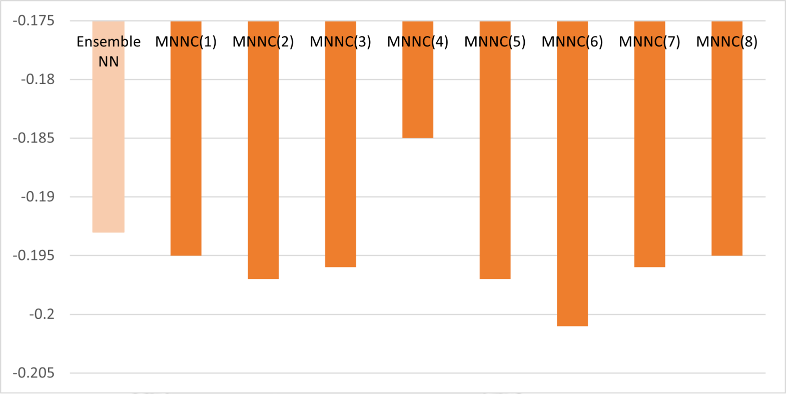


Figure 4.36: Differences in precision, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

4.2.1.2 Recall

- No overlap
  - Gaussian format

    For datasets comprising 100, 200, 300, and 400 instances of class 1, the recall values for all iterations of ensemble NN and MNNC are consistent with the Best *k*-NN, both registering a recall of 1. However, when the dataset size is 500 instances of class 1, only ensemble NN and MNNC(6) demonstrate recall values equal to the Best k-NN. The remaining versions of MNNC exhibit lower recall values compared to Best k-NN and ensemble NN, as depicted in Figure 4.37.

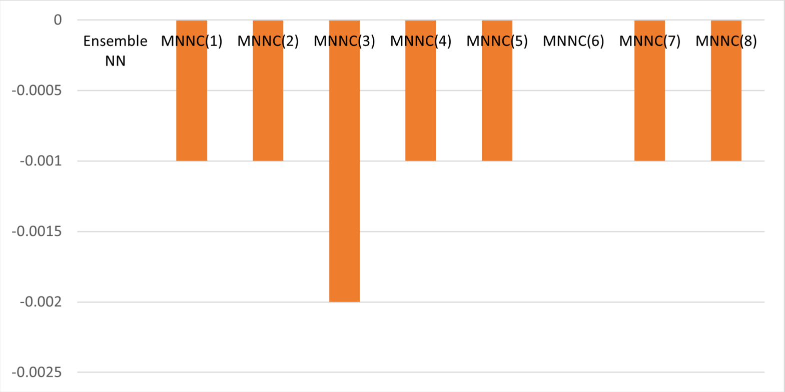


Figure 4.37: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

For datasets with 100 and 500 instances of class 1, the recall of both ensemble NN and all versions of MNNC is lower than that of Best k-NN, as illustrated in Figures 4.38 and 4.39, respectively.

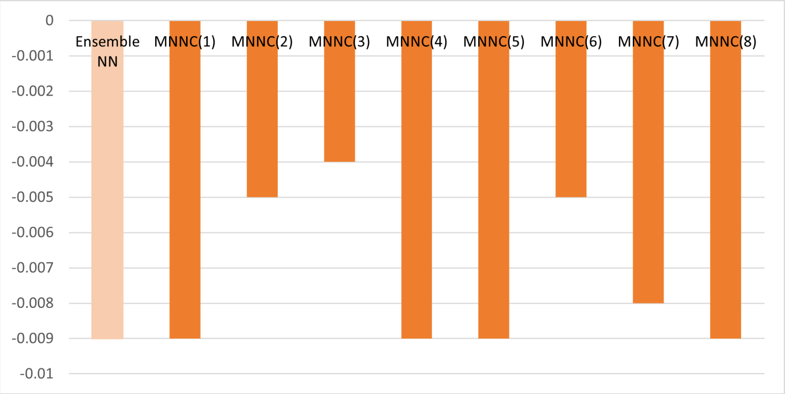


Figure 4.38: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

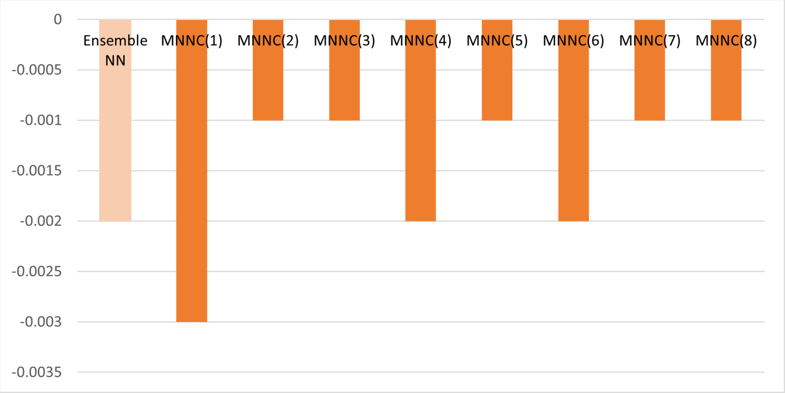


Figure 4.39: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

For datasets comprising 200 instances of class 1, the recall values for ensemble NN, MNNC(3), MNNC(4), and MNNC(7) are equivalent to that of Best k-NN, while the recall of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.40.

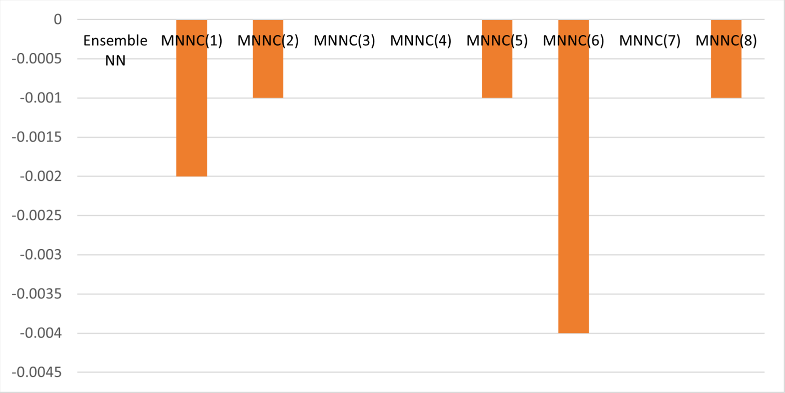


Figure 4.40: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

For datasets comprising 300 instances of class 1, the recall values for ensemble NN, MNNC(2), MNNC(3), MNNC(4), MNNC(6) and MNNC(7) are equivalent to that of Best k-NN, while the recall of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.41.

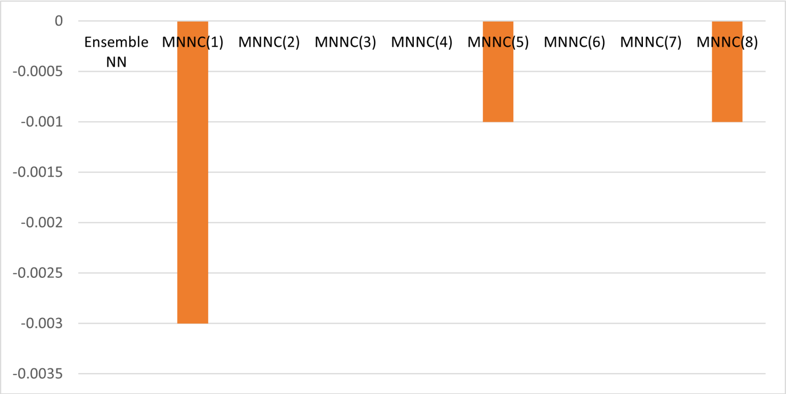


Figure 4.41: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

For datasets comprising 400 instances of class 1, the recall values for ensemble NN, MNNC(2), MNNC(3), MNNC(4), MNNC(5), and MNNC(8) are equivalent to that of Best k-NN, while the recall of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.42.

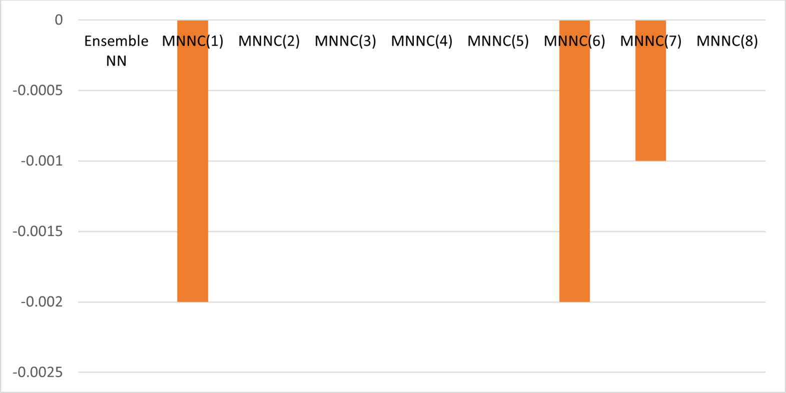


Figure 4.42: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

– Circle format

The recall of Best k-NN, Ensemble NN, and all MNNC exhibited perfect scores of 1 when evaluated on circle format with no overlap.

- Slight overlap

– Gaussian format

The recall of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.43 to 4.47, respectively.

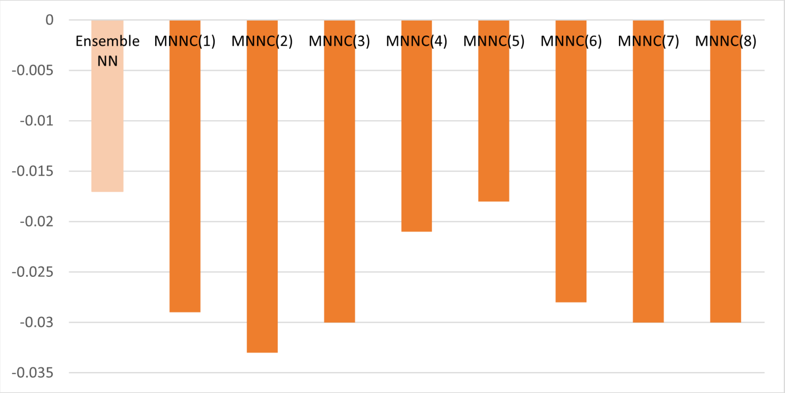


Figure 4.43: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

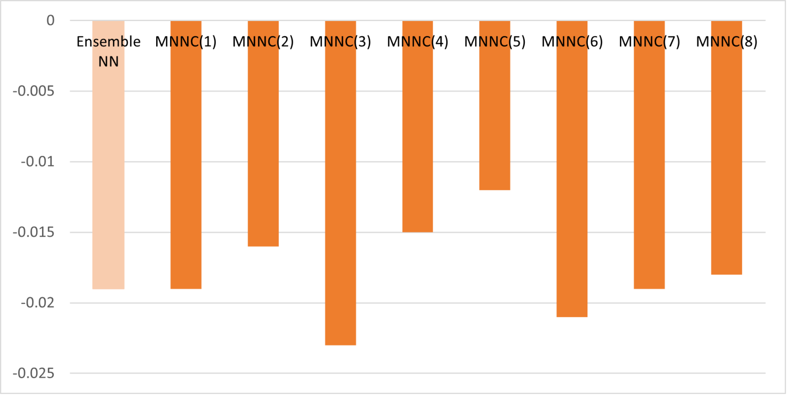


Figure 4.44: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

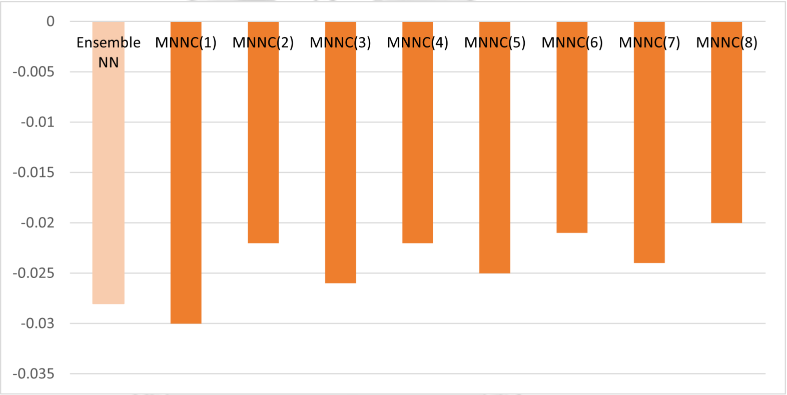


Figure 4.45: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

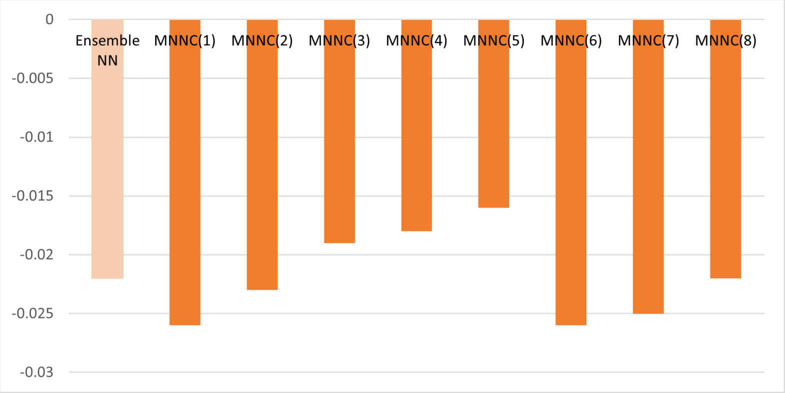


Figure 4.46: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

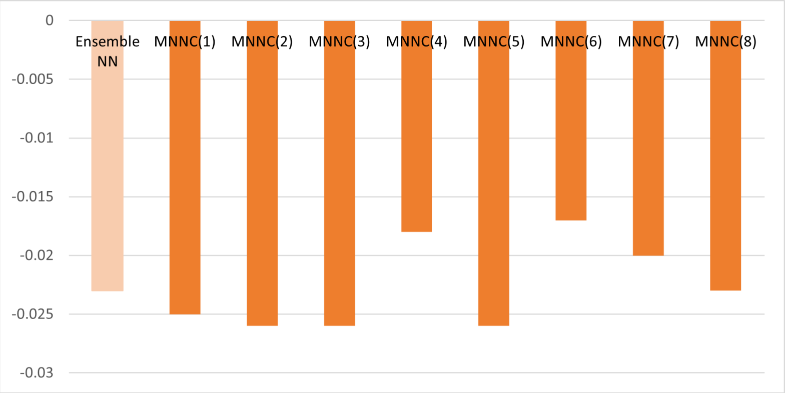


Figure 4.47: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

The recall of both ensemble NN and all versions of MNNC is consistently lower than that of Best *k*-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.48 to 4.52, respectively.

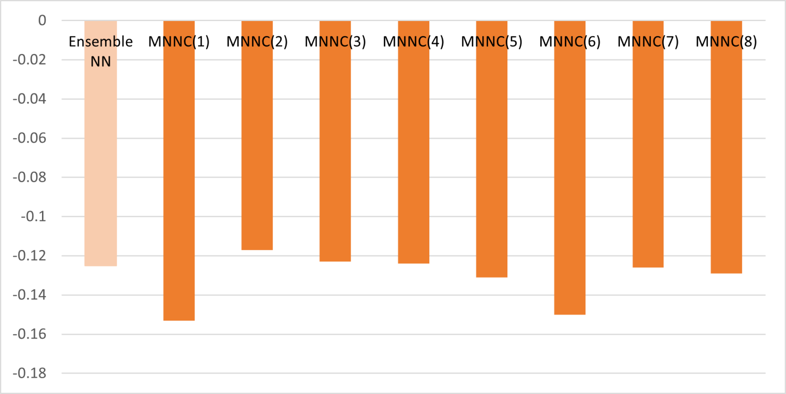


Figure 4.48: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

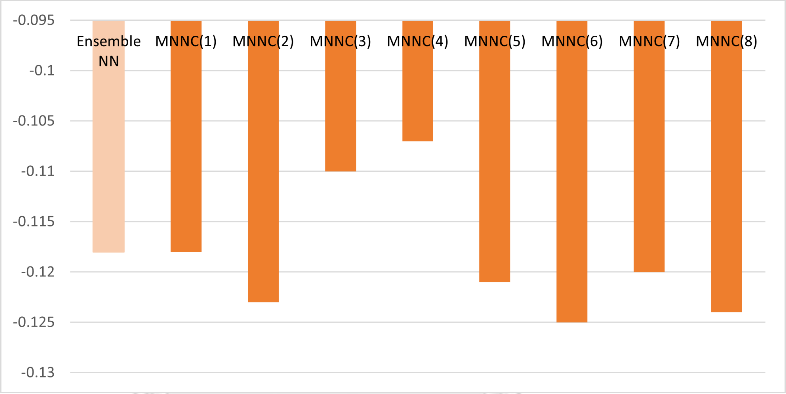


Figure 4.49: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

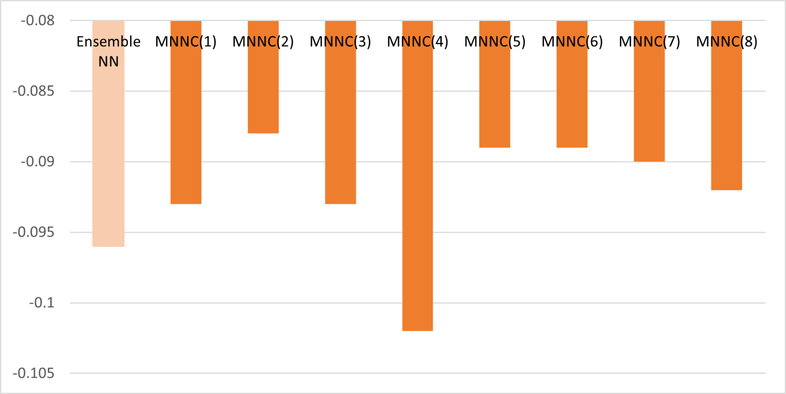


Figure 4.50: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

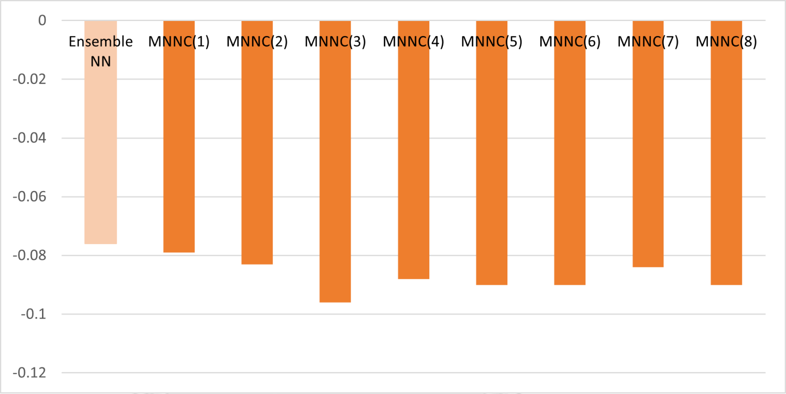


Figure 4.51: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

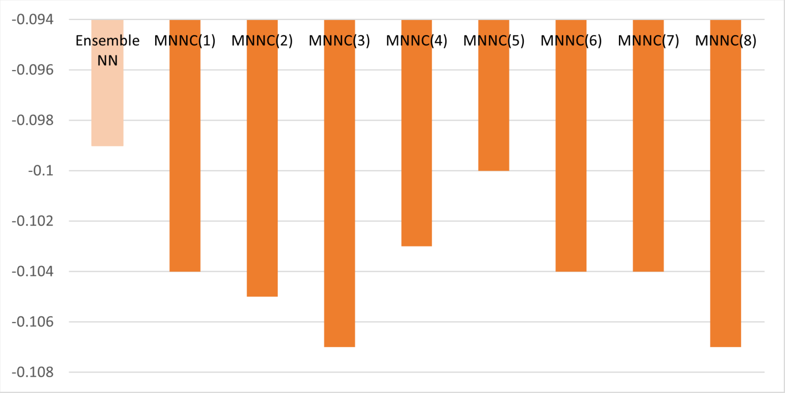


Figure 4.52: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Circle format

The recall of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.53 to 4.57, respectively.

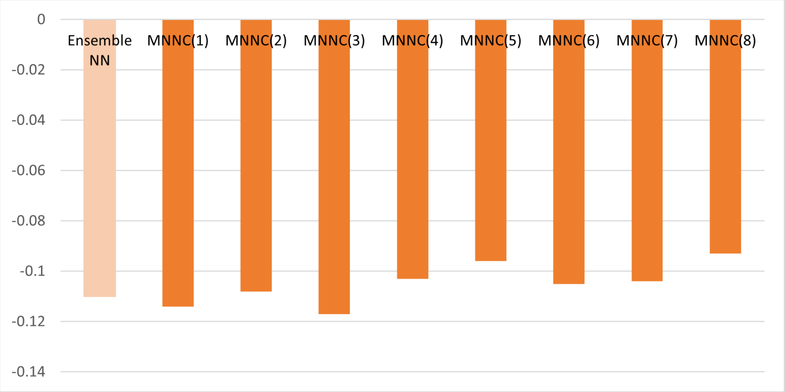


Figure 4.53: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

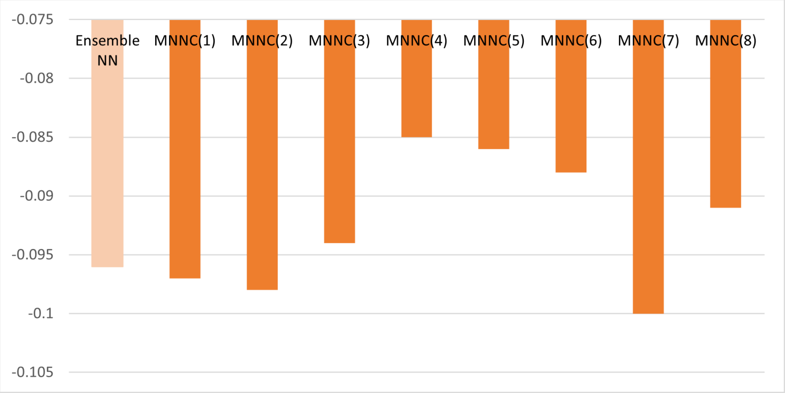


Figure 4.54: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

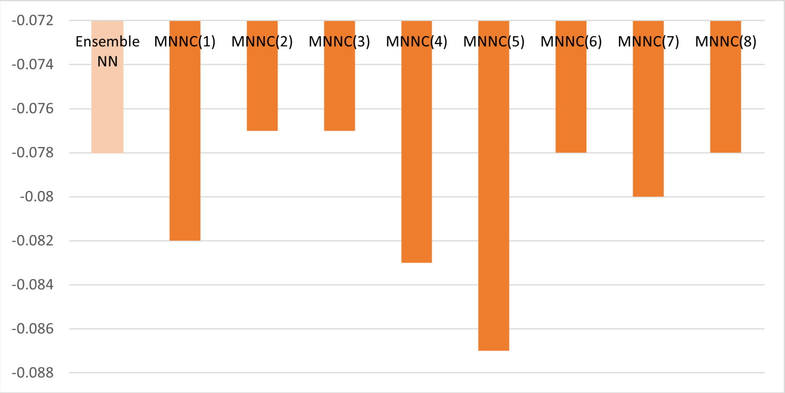


Figure 4.55: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

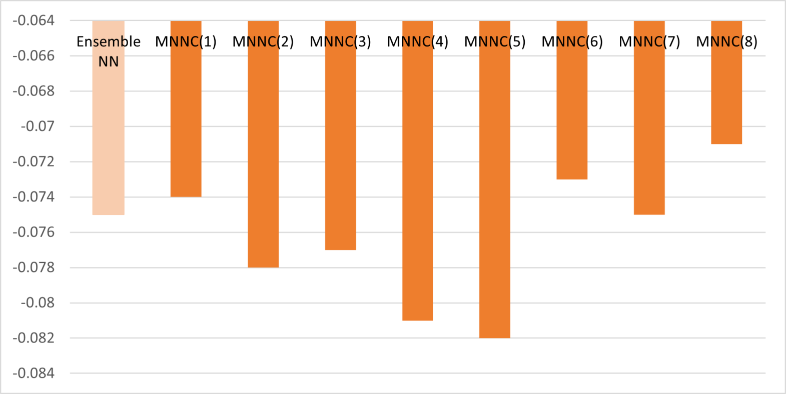


Figure 4.56: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

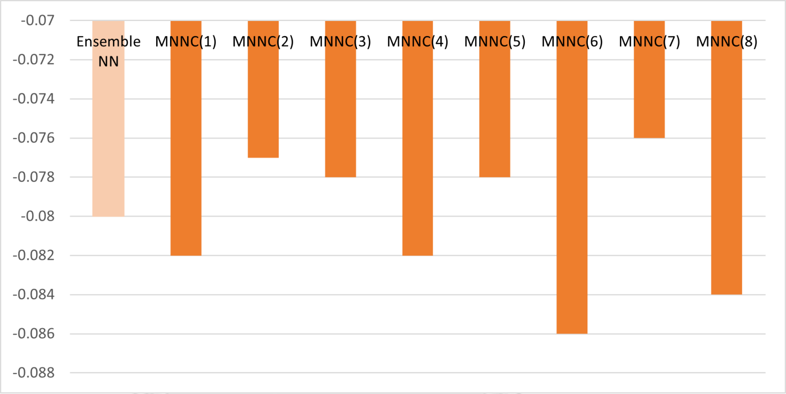


Figure 4.57: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

- Large overlap

  – Gaussian format

  The recall of both ensemble NN and all versions of MNNC is consistently lower than that of Best *k*-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.58 to 4.62, respectively.

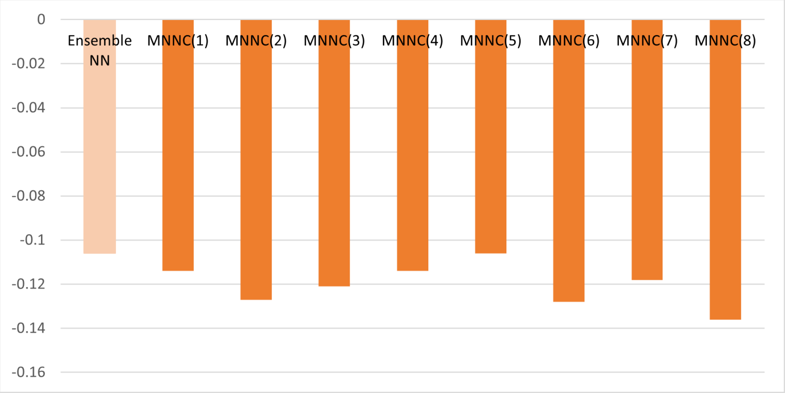


Figure 4.58: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

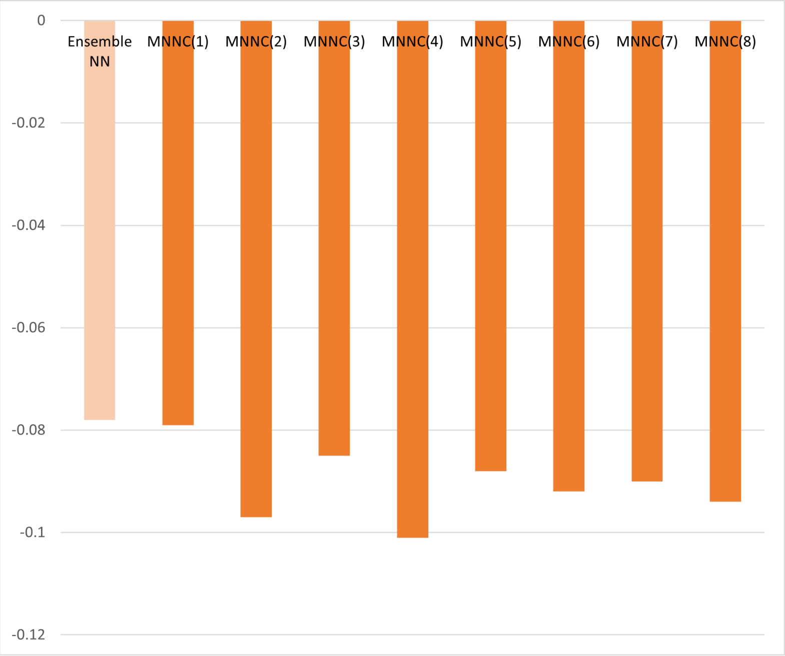


Figure 4.59: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

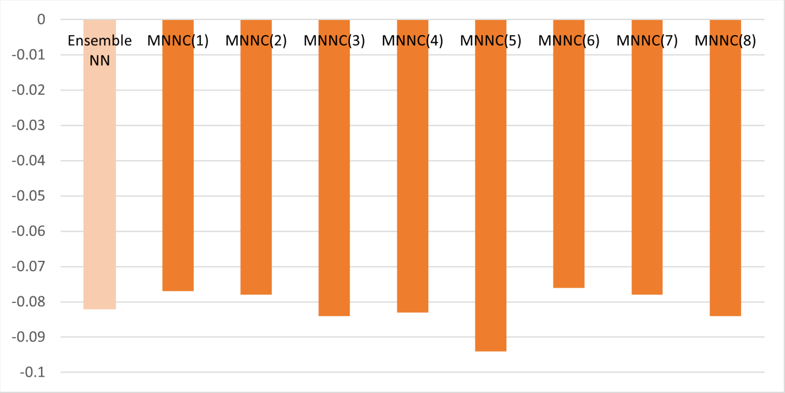


Figure 4.60: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

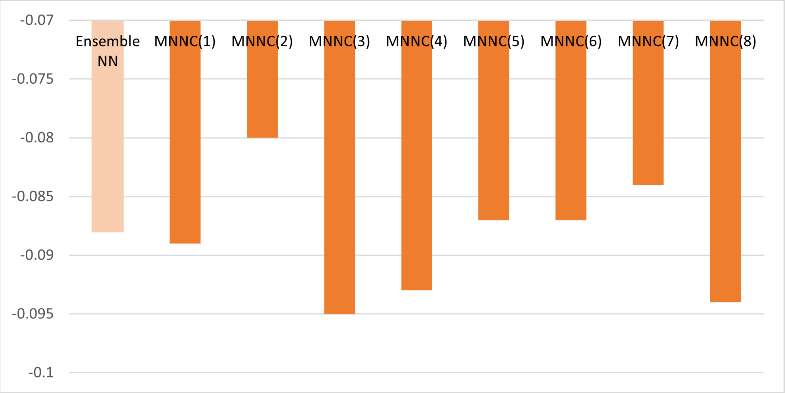


Figure 4.61: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

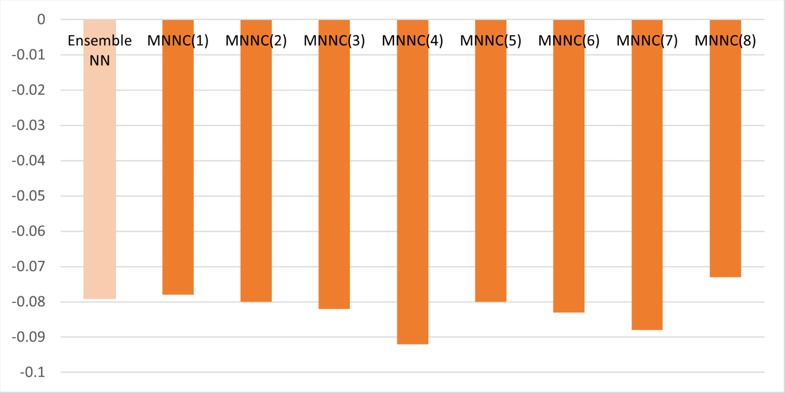


Figure 4.62: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

The recall of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.63 to 4.67, respectively.

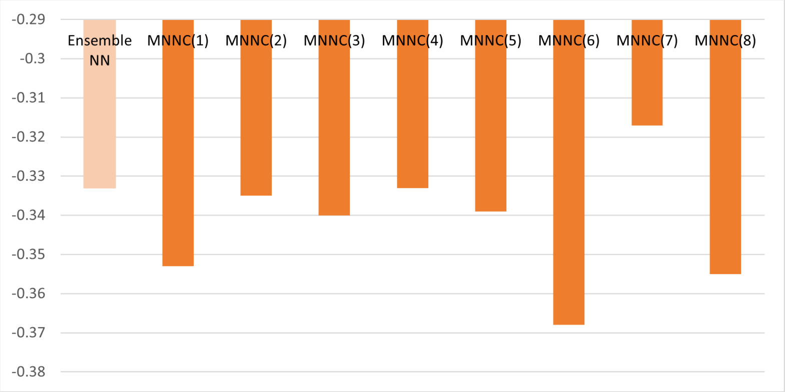


Figure 4.63: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

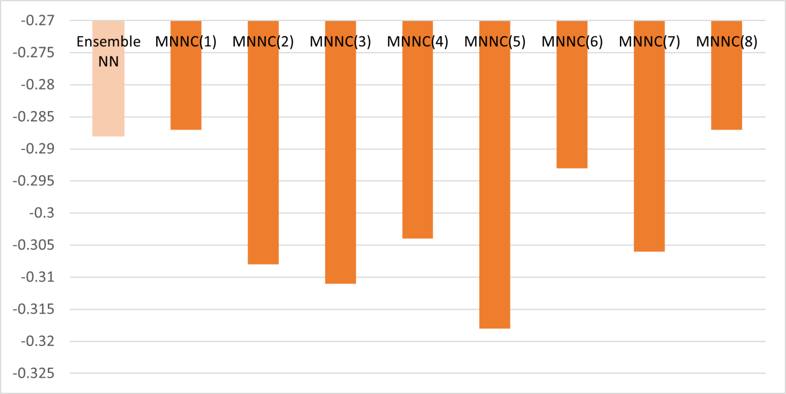


Figure 4.64: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

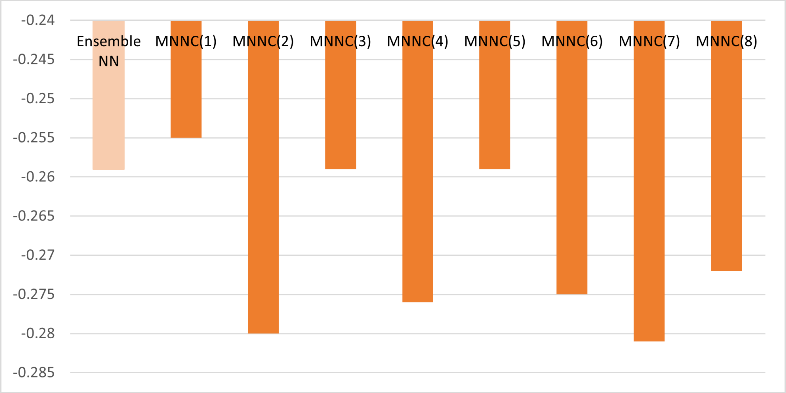


Figure 4.65: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

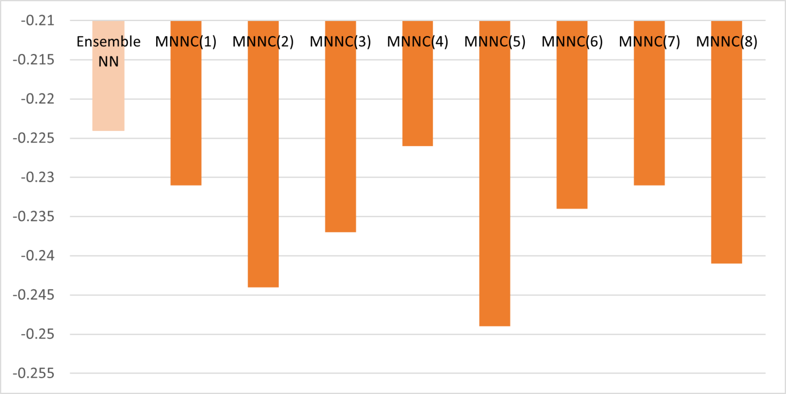


Figure 4.66: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

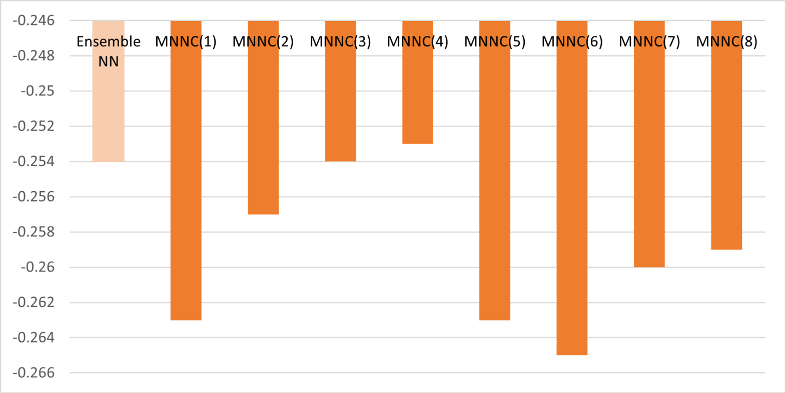


Figure 4.67: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Circle format

The recall of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.68 to 4.72, respectively.

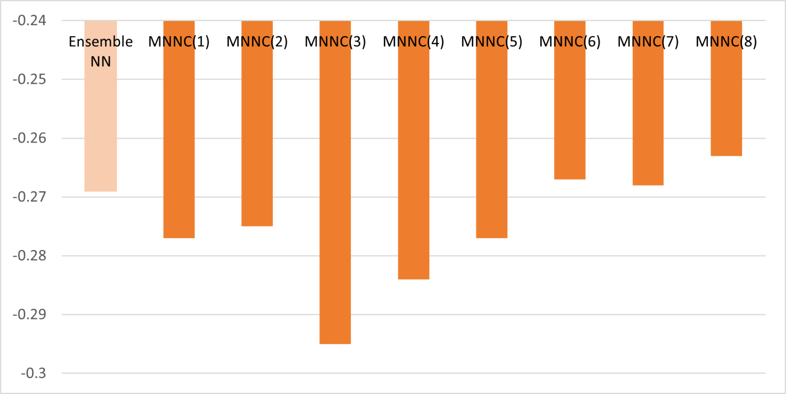


Figure 4.68: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

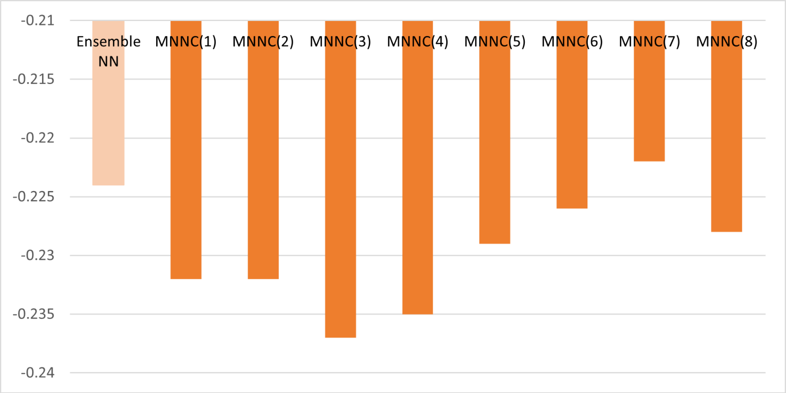


Figure 4.69: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

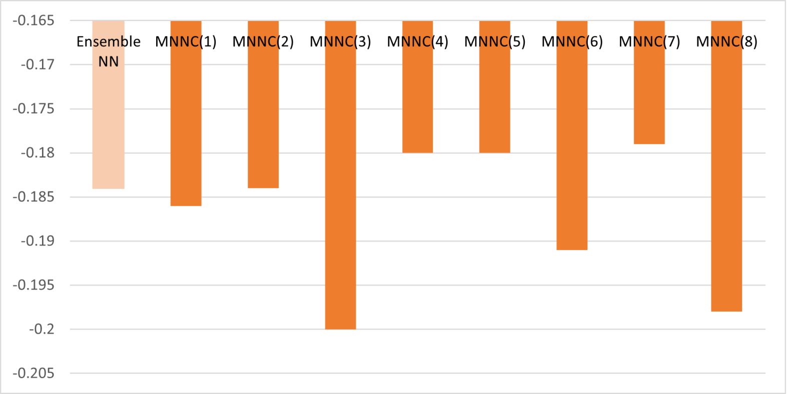


Figure 4.70: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

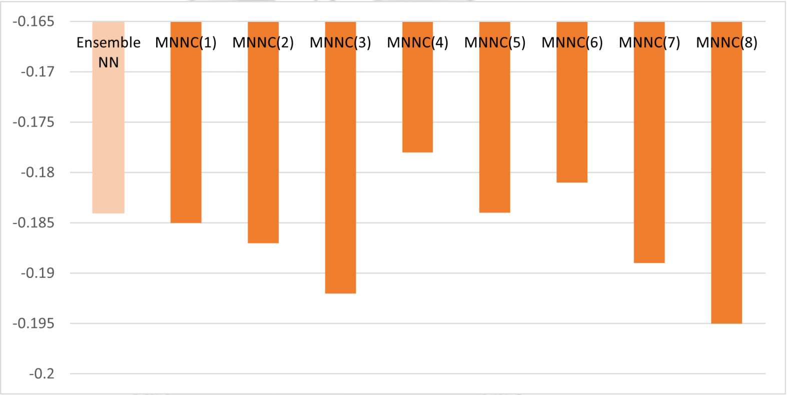


Figure 4.71: Differences in recall, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

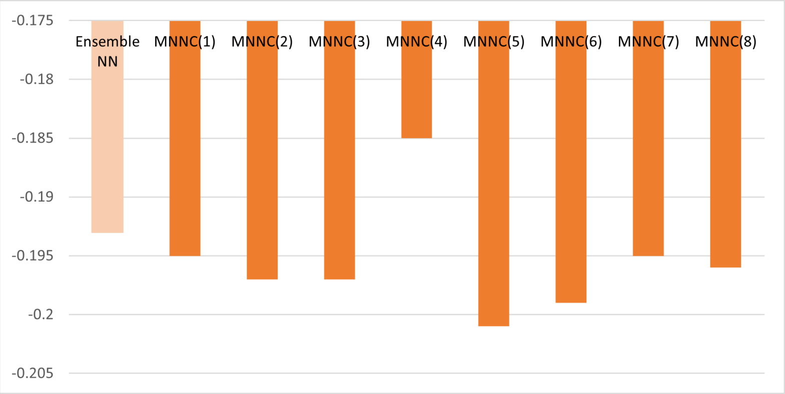


Figure 4.72: Differences in recall, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

4.2.1.3 F1-score

- No overlap
  - Gaussian format

    For datasets comprising 100, 200, 300, and 400 instances of class 1, the F1-score values for all iterations of ensemble NN and MNNC are consistent with the Best *k*-NN, both registering a F1-score of 1. However, when the dataset size is 500 instances of class 1, only ensemble NN and MNNC(6) demonstrate F1-score values equal to the Best *k*-NN. The remaining versions of MNNC exhibit lower F1-score values compared to Best *k*-NN and ensemble NN, as depicted in Figure 4.73.

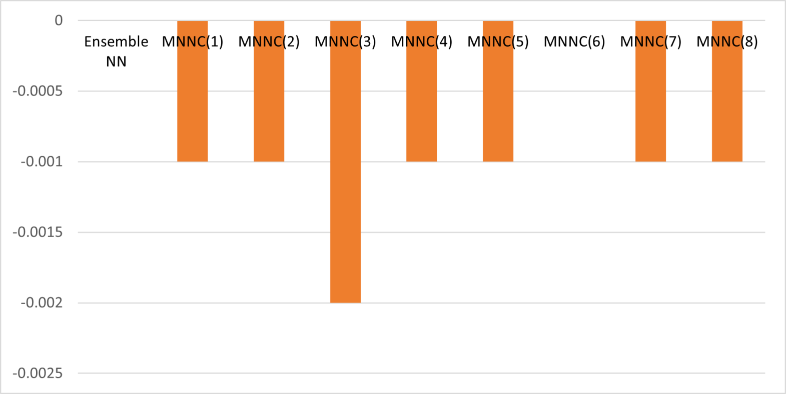


Figure 4.73: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

For datasets with 100 instances of class 1, the F1-score of both ensemble NN and all versions of MNNC is lower than that of Best k-NN, as illustrated in Figures 4.74.

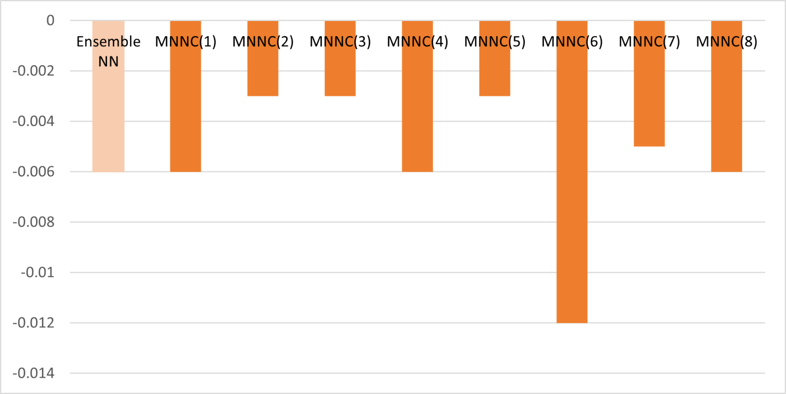


Figure 4.74: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

For datasets comprising 200 instances of class 1, the F1-score values for ensemble NN, MNNC(3), MNNC(4), and MNNC(7) are equivalent to that of Best k-NN, while the F1-score of the remaining versions of

MNNC is lower than Best *k*-NN, as illustrated in Figure 4.75.

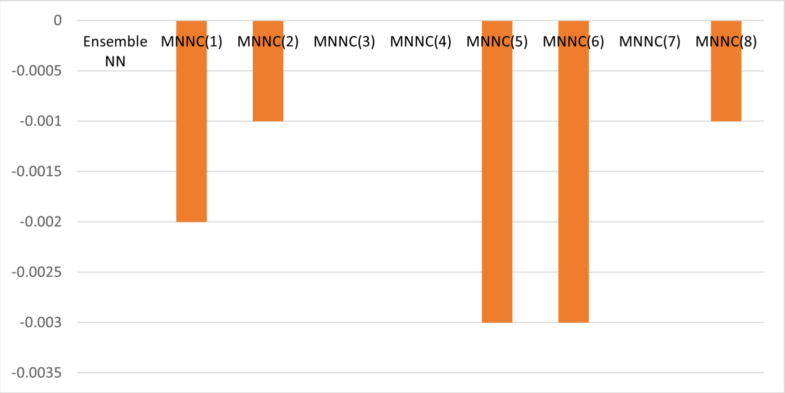


Figure 4.75: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

For datasets comprising 300 instances of class 1, the F1-score values for ensemble NN, MNNC(2), MNNC(4), and MNNC(5) are equivalent to that of Best *k*-NN, while the F1-score of the remaining versions of MNNC is lower than Best *k*-NN, as illustrated in Figure 4.76.

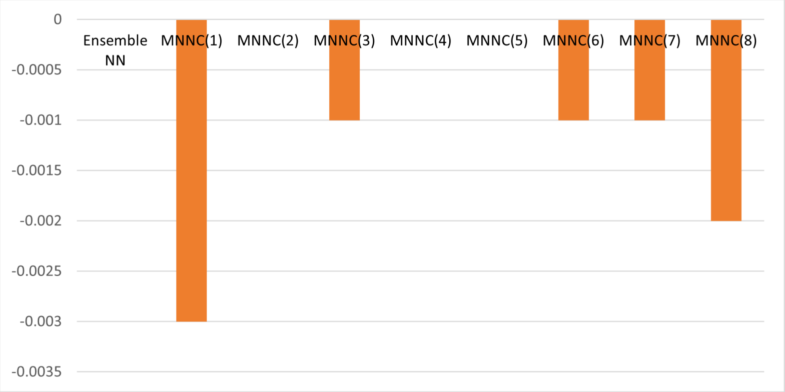


Figure 4.76: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

For datasets comprising 400 instances of class 1, the F1-score values for ensemble NN, MNNC(2), MNNC(3), MNNC(4), and MNNC(8) are equivalent to that of Best k-NN, while the F1-score of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.77.

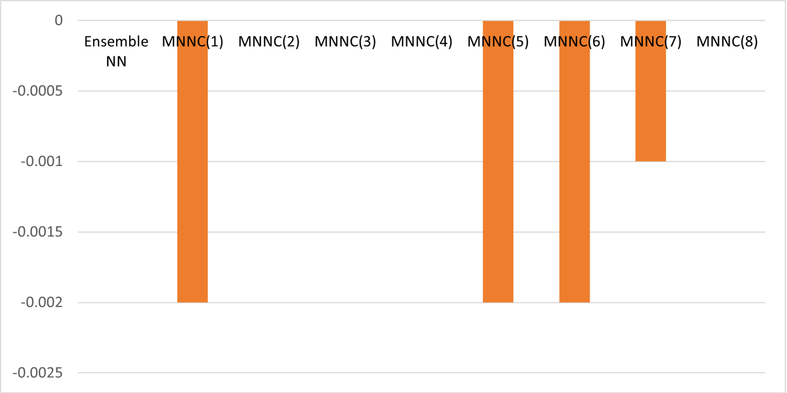


Figure 4.77: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

For datasets comprising 500 instances of class 1, the F1-score values for ensemble NN is equivalent to that of Best k-NN, while the F1-score of all versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.78.

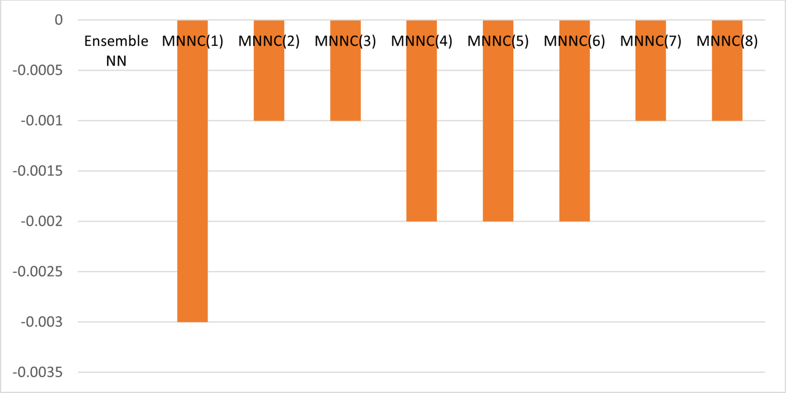


Figure 4.78: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Circle format

The F1-score of Best k-NN, Ensemble NN, and all MNNC exhibited perfect scores of 1 when evaluated on circle format with no overlap.

- Slight overlap

– Gaussian format

The F1-score of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.79 to 4.83, respectively.

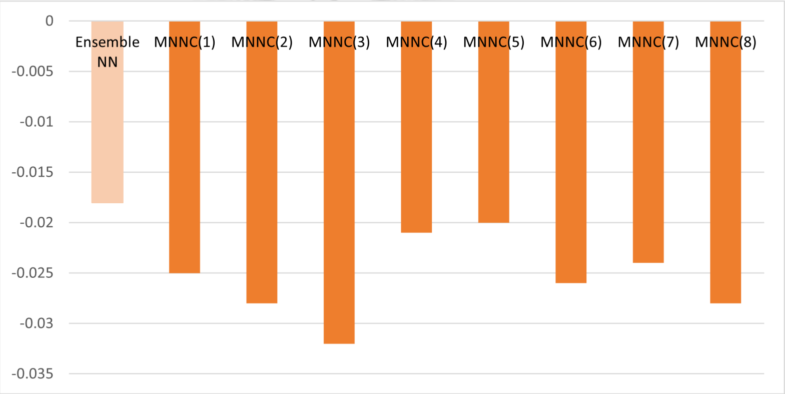


Figure 4.79: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

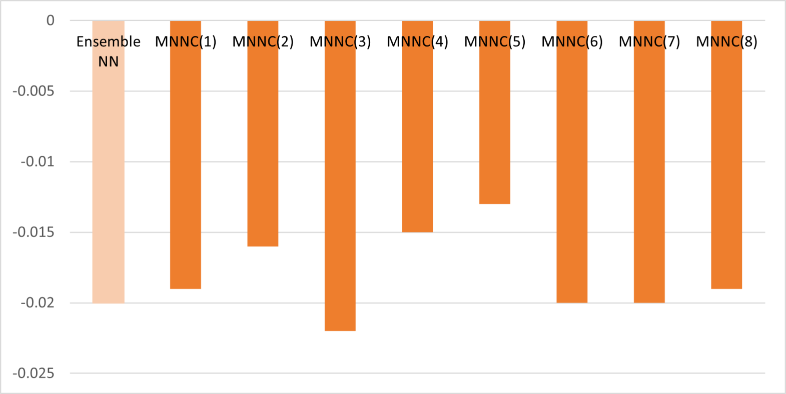


Figure 4.80: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

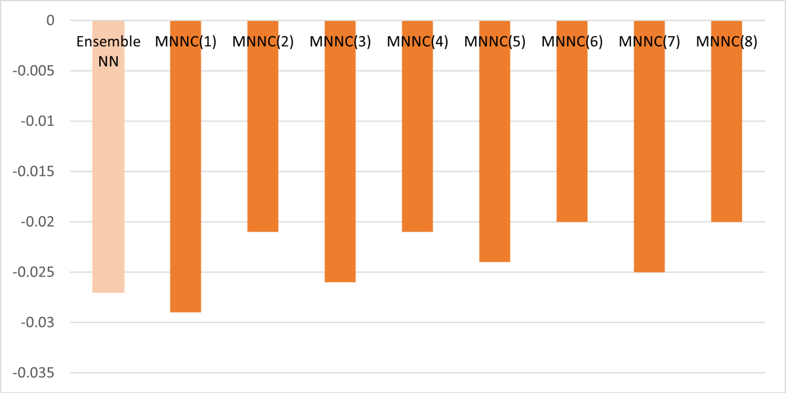


Figure 4.81: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

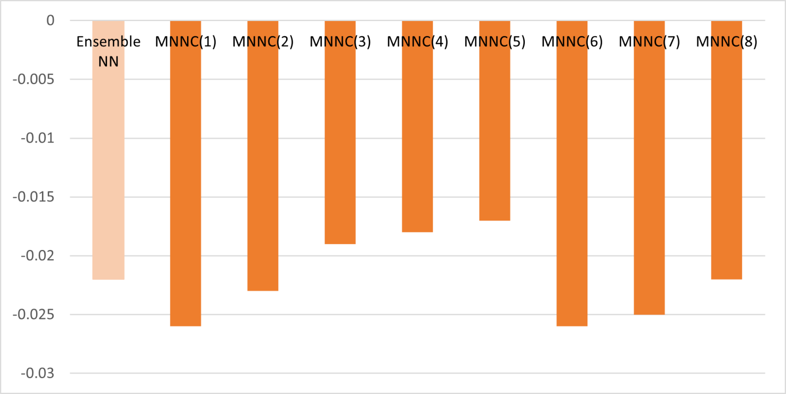


Figure 4.82: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

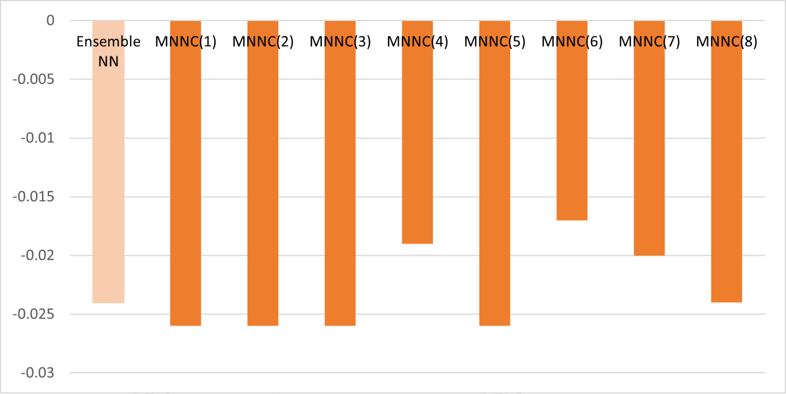


Figure 4.83: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

The F1-score of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.84 to 4.88, respectively.

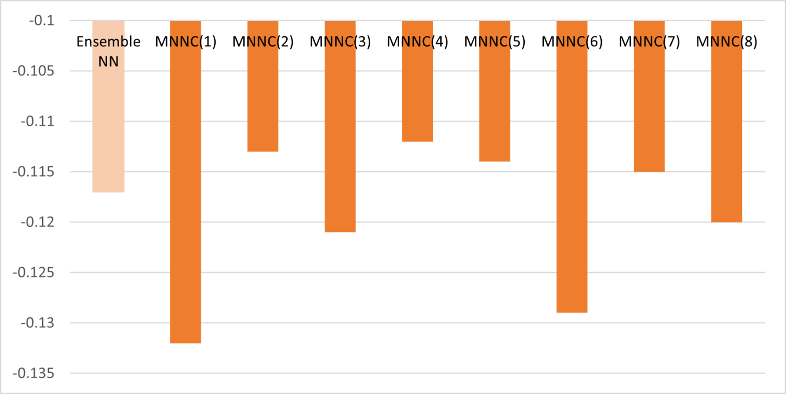


Figure 4.84: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

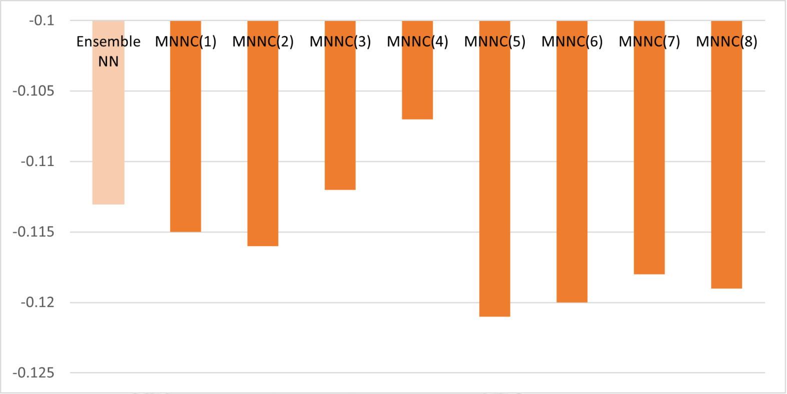


Figure 4.85: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

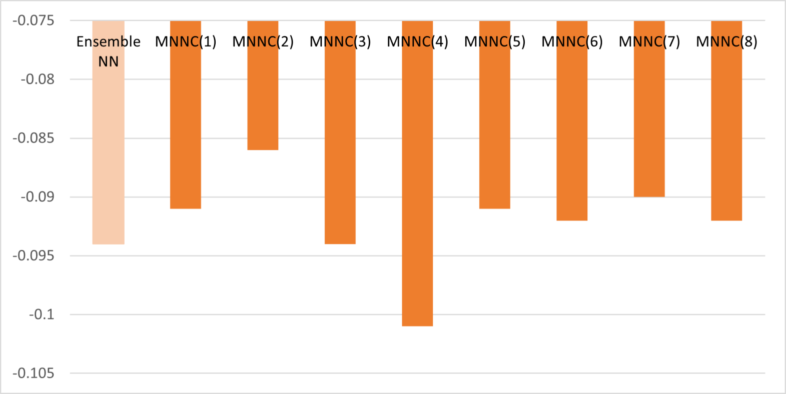


Figure 4.86: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

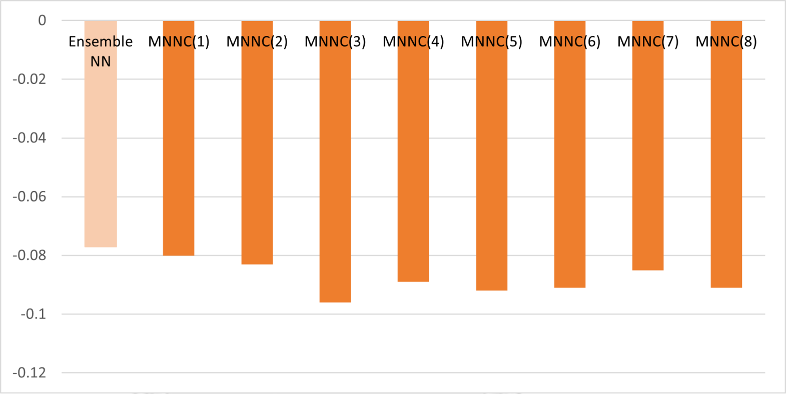


Figure 4.87: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

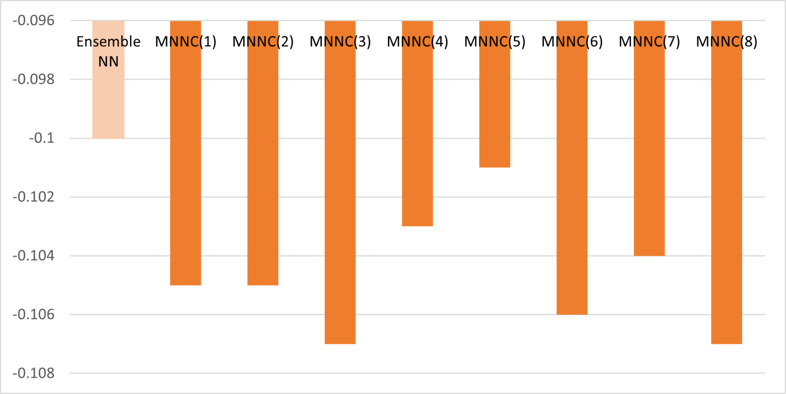


Figure 4.88: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

– Circle format

The F1-score of both ensemble NN and all versions of MNNC is consistently lower than that of Best *k*-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.89 to 4.93, respectively.

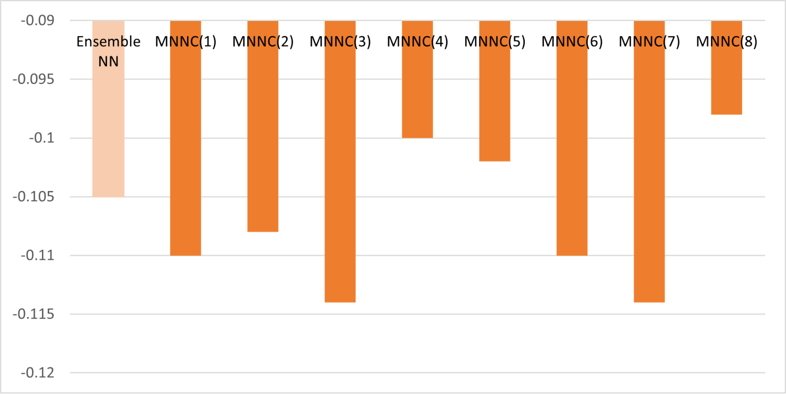


Figure 4.89: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

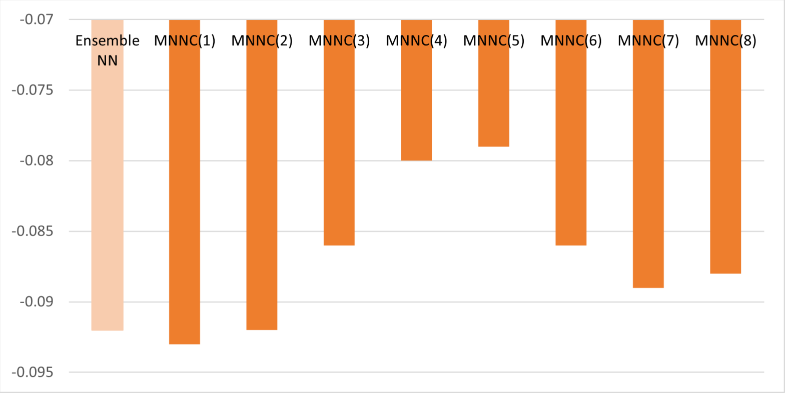


Figure 4.90: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

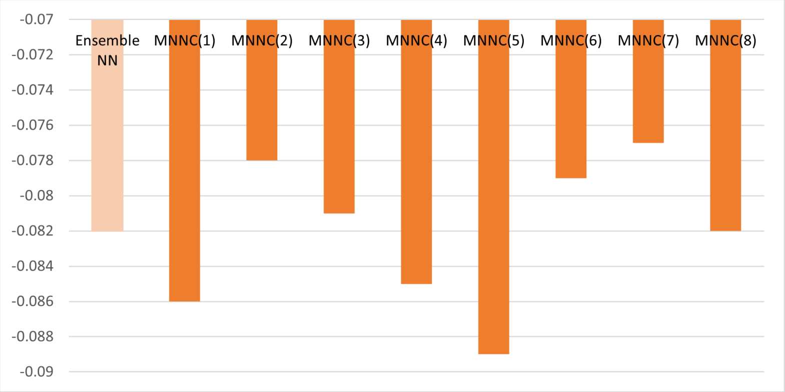


Figure 4.91: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

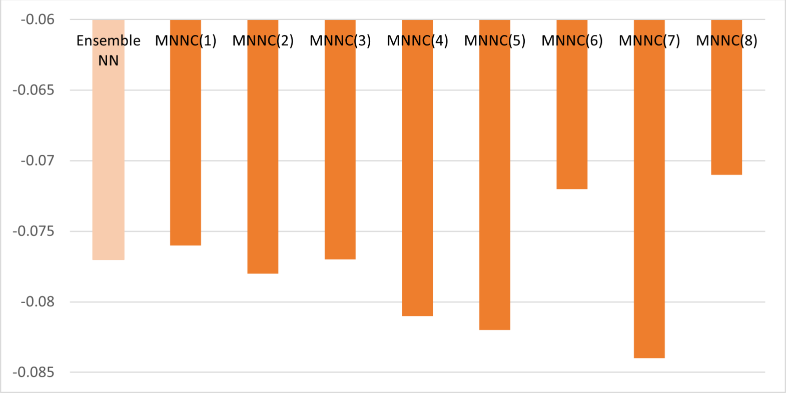


Figure 4.92: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

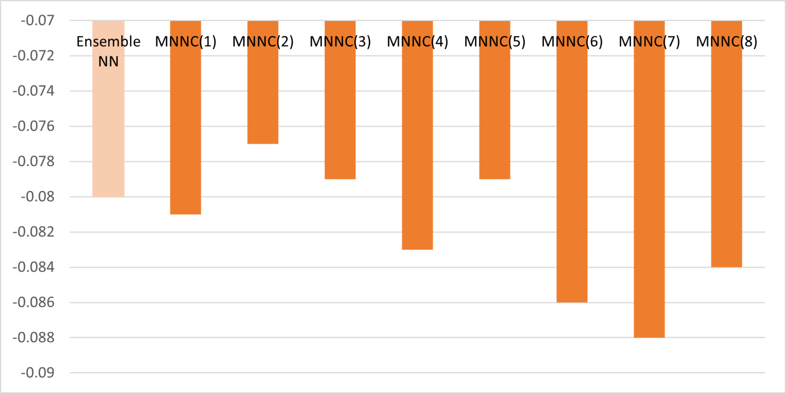


Figure 4.93: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Large overlap

  – Gaussian format

  The F1-score of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.94 to 4.98, respectively.

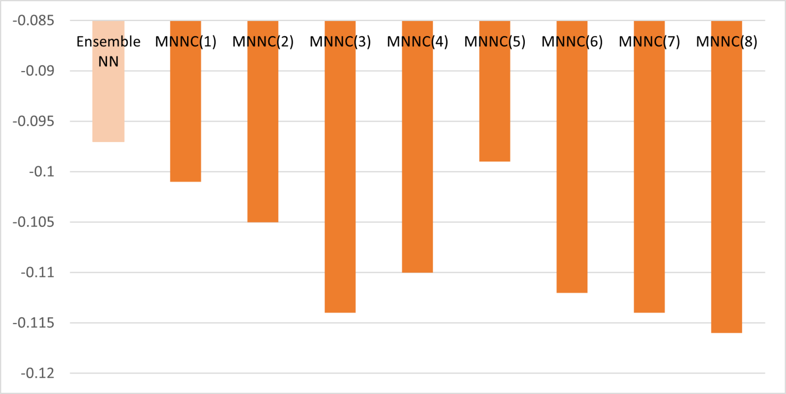


Figure 4.94: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

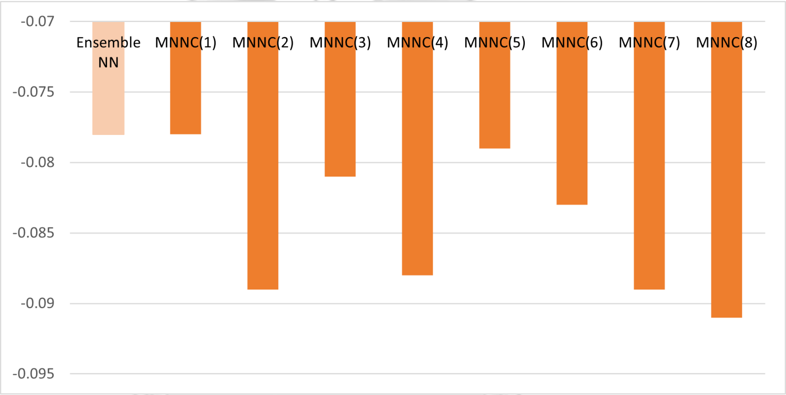


Figure 4.95: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

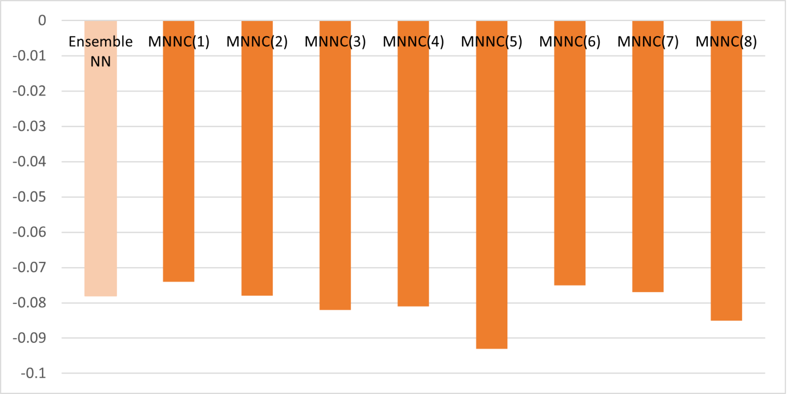


Figure 4.96: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

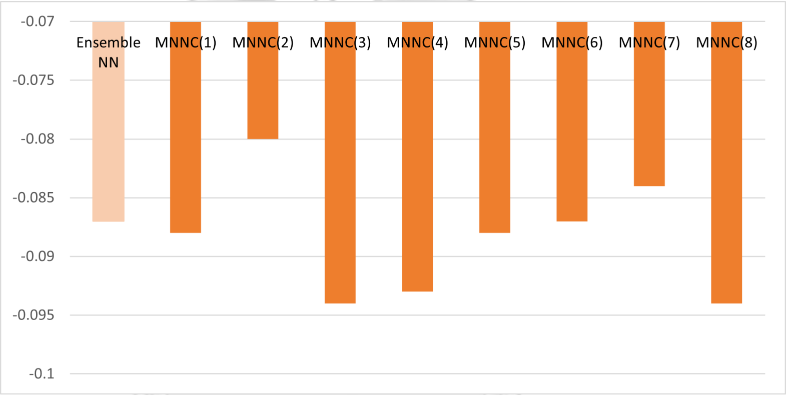


Figure 4.97: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

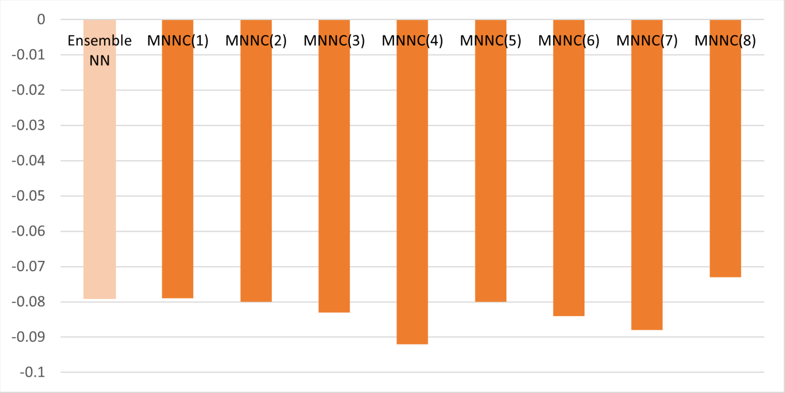


Figure 4.98: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Moon shaped format

  The F1-score of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.99 to 4.103, respectively.

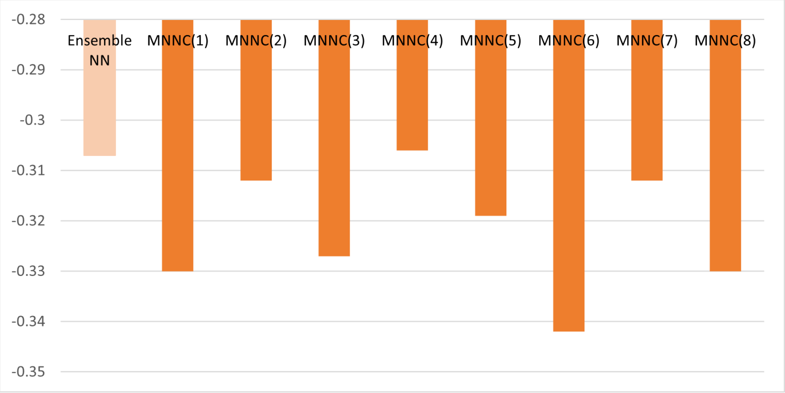


Figure 4.99: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

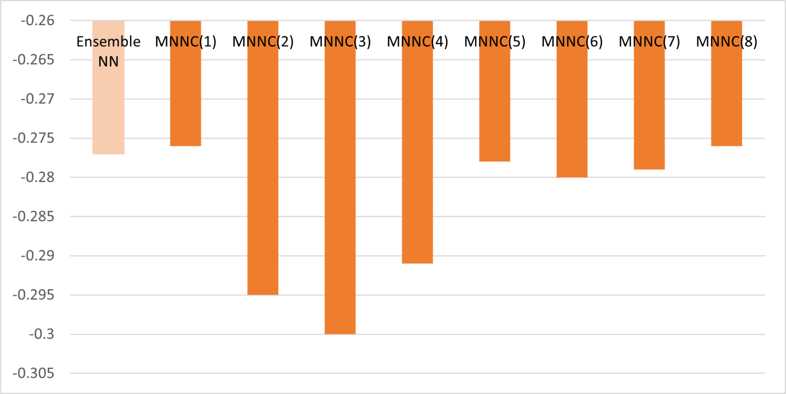


Figure 4.100: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

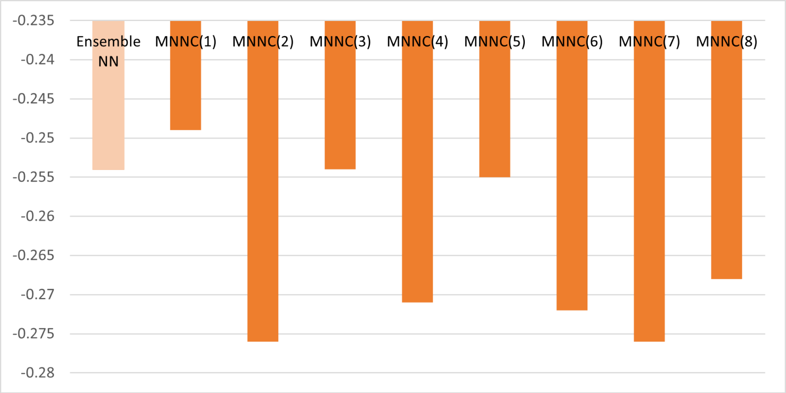


Figure 4.101: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

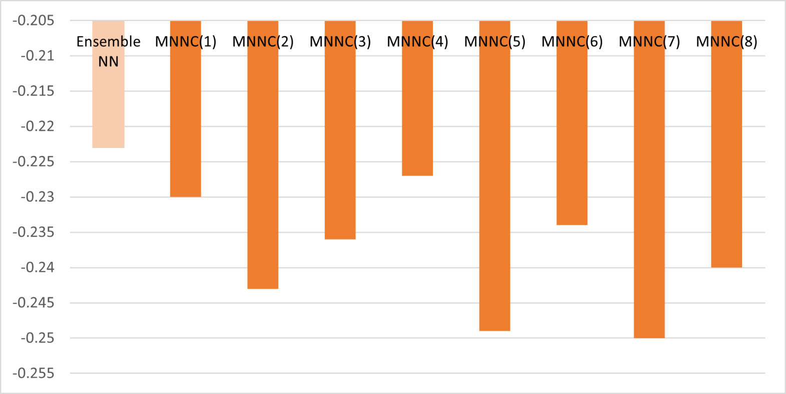


Figure 4.102: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

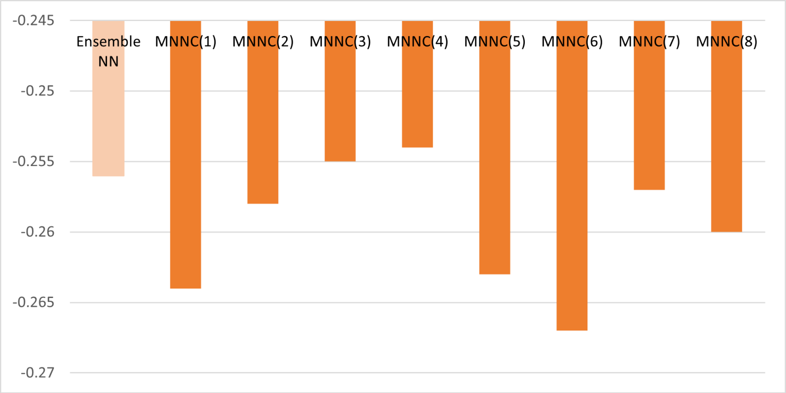


Figure 4.103: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Circle format

The F1-score of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.104 to 4.108, respectively.

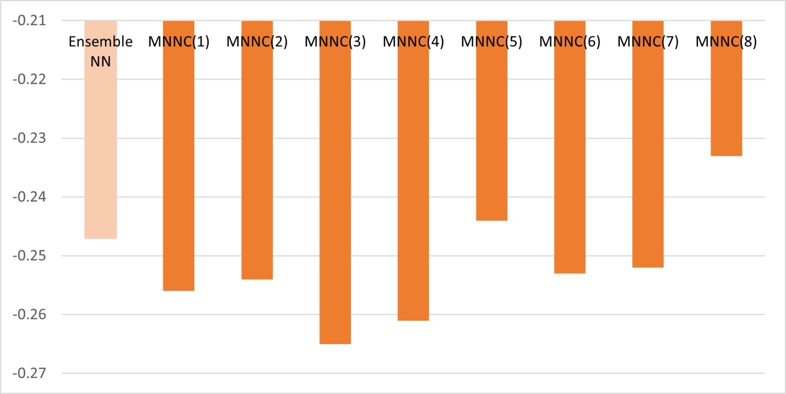


Figure 4.104: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

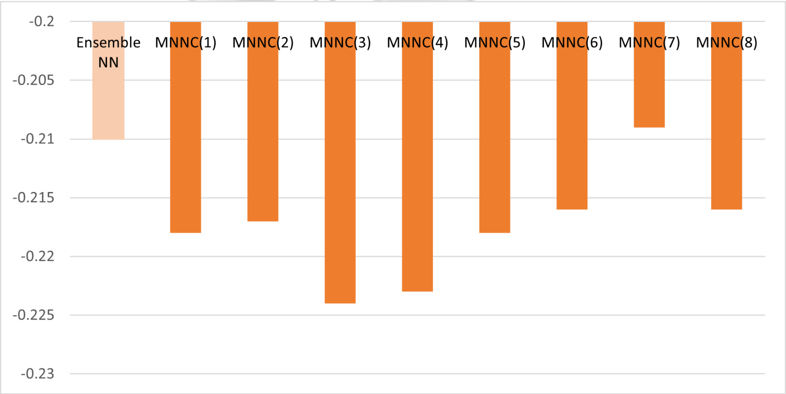


Figure 4.105: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

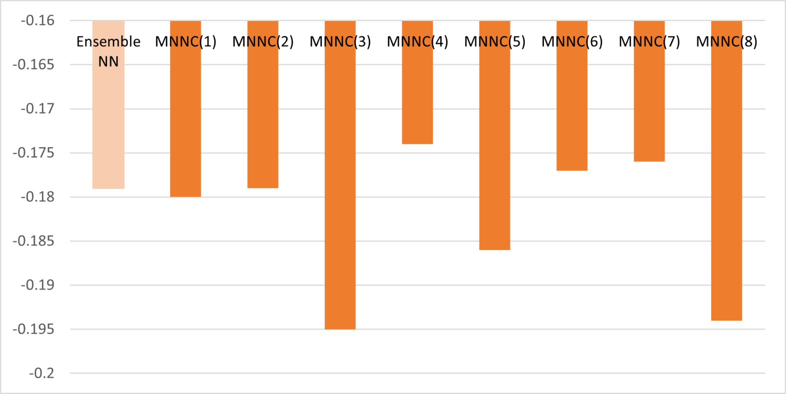


Figure 4.106: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

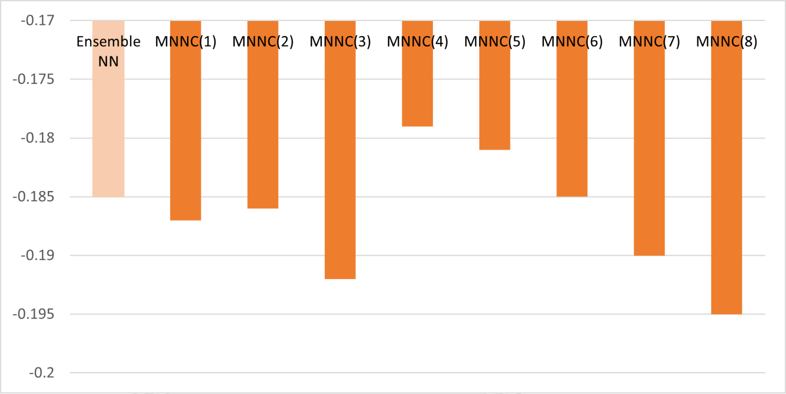


Figure 4.107: Differences in F1-score, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

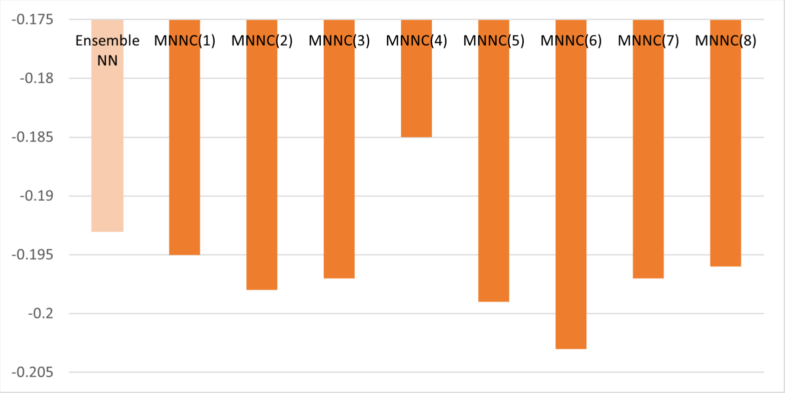


Figure 4.108: Differences in F1-score, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

4.2.1.4 Accuracy

- No overlap
  - Gaussian format

    For datasets comprising 100, 200, 300, and 400 instances of class 1, the accuracy values for all iterations of ensemble NN and MNNC are consistent with the Best k-NN, both registering a accuracy of 1. However, when the dataset size is 500 instances of class 1, only ensemble NN and MNNC(6) demonstrate accuracy values equal to the Best k-NN. The remaining versions of MNNC exhibit lower accuracy values compared to Best k-NN and ensemble NN, as depicted in Figure 4.109.

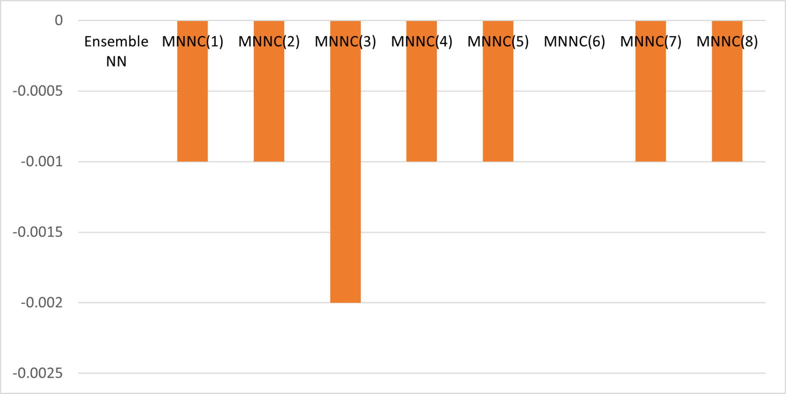


Figure 4.109: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Moon shaped format

  For datasets with 100 and 500 instances of class 1, the accuracy of both ensemble NN and all versions of MNNC is lower than that of Best k-NN, as illustrated in Figures 4.110 and 4.111, respectively.

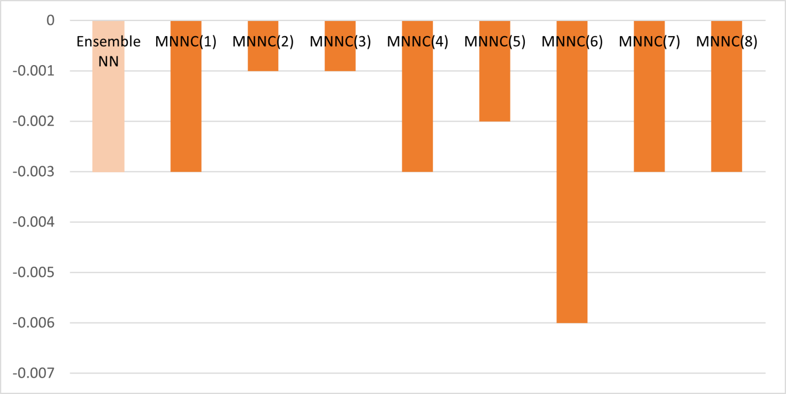


Figure 4.110: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

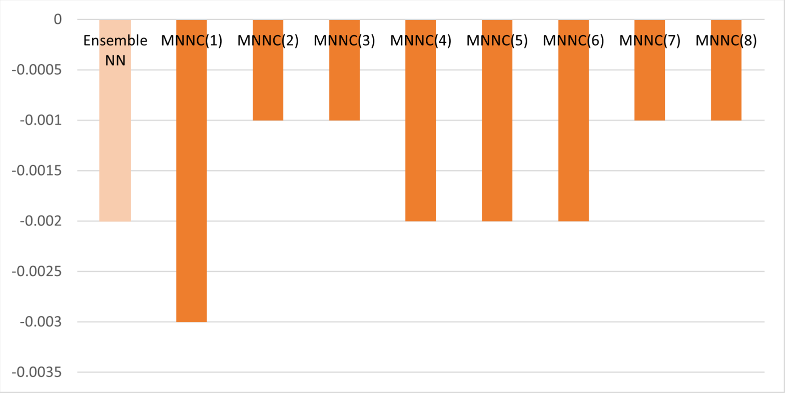


Figure 4.111: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

For datasets comprising 200 instances of class 1, the accuracy values for ensemble NN, MNNC(3), MNNC(4), and MNNC(7) are equivalent to that of Best *k*-NN, while the accuracy of the remaining versions of MNNC is lower than Best *k*-NN, as illustrated in Figure 4.112.

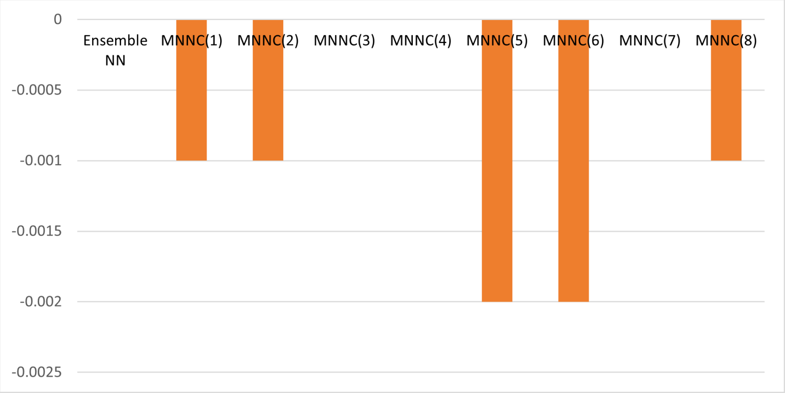


Figure 4.112: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

For datasets comprising 300 instances of class 1, the accuracy values for ensemble NN, MNNC(2), MNNC(3), MNNC(4), MNNC(5), MNNC(6) and MNNC(7) are equivalent to that of Best *k*-NN, while the accuracy of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.113.

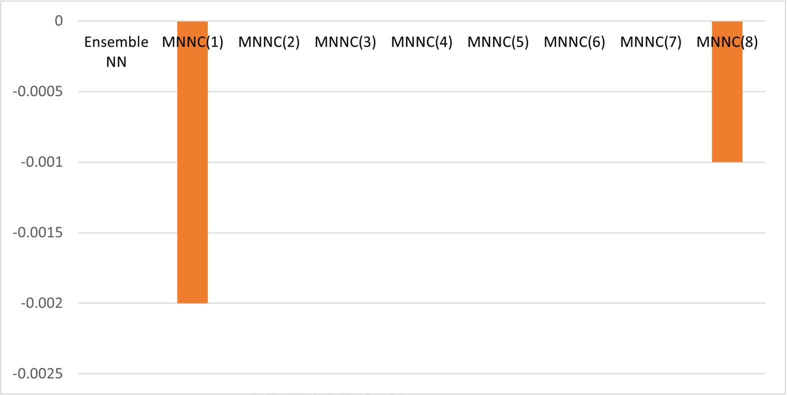


Figure 4.113: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

For datasets comprising 400 instances of class 1, the accuracy values for ensemble NN, MNNC(2), MNNC(3), MNNC(4), and MNNC(8) are equivalent to that of Best k-NN, while the accuracy of the remaining versions of MNNC is lower than Best k-NN, as illustrated in Figure 4.114.

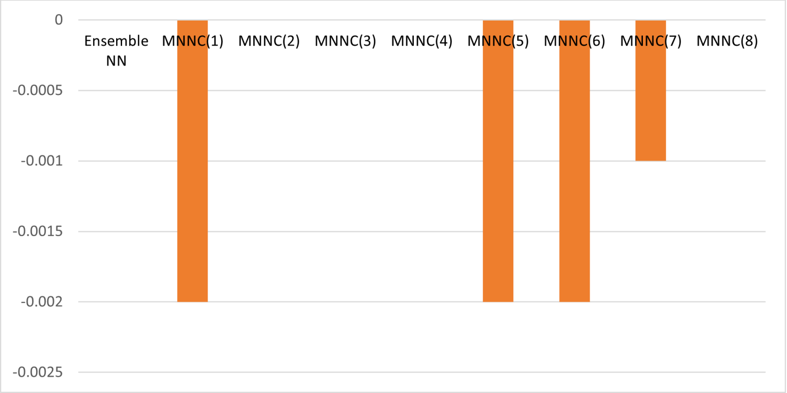


Figure 4.114: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

- Circle format

  The accuracy of Best k-NN, Ensemble NN, and all MNNC exhibited perfect scores of 1 when evaluated on circle format with no overlap.

- Slight overlap

  - Gaussian format

    The accuracy of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.115 to 4.119, respectively.

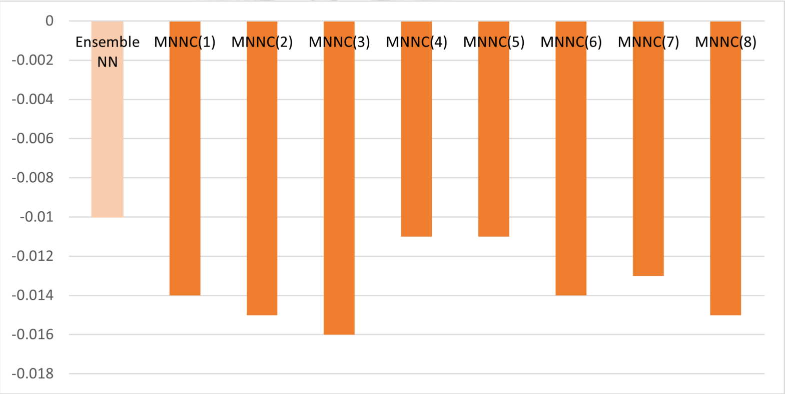


Figure 4.115: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

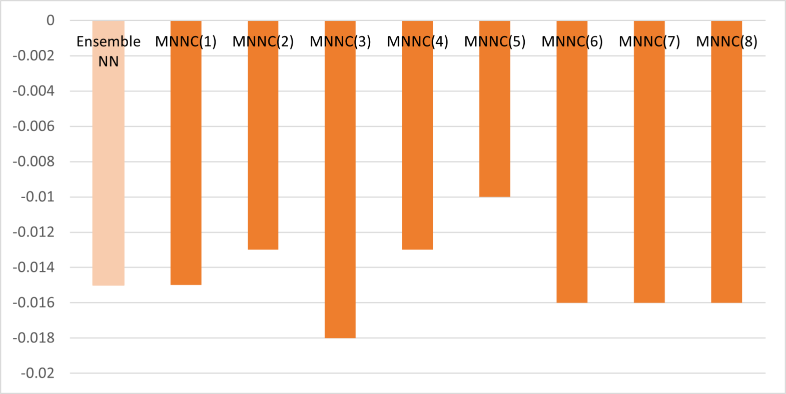


Figure 4.116: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

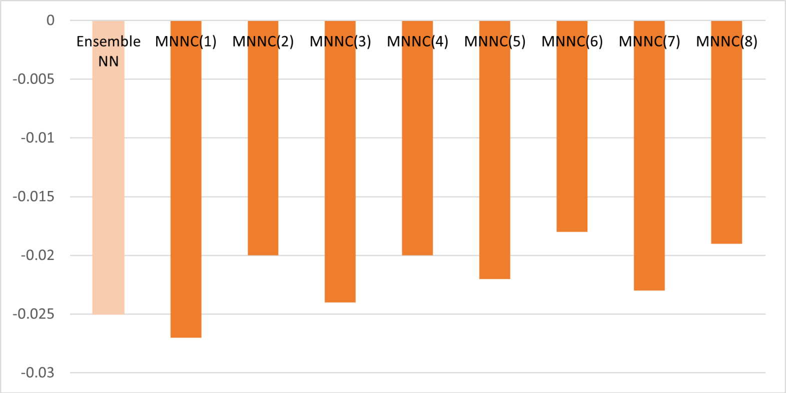


Figure 4.117: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

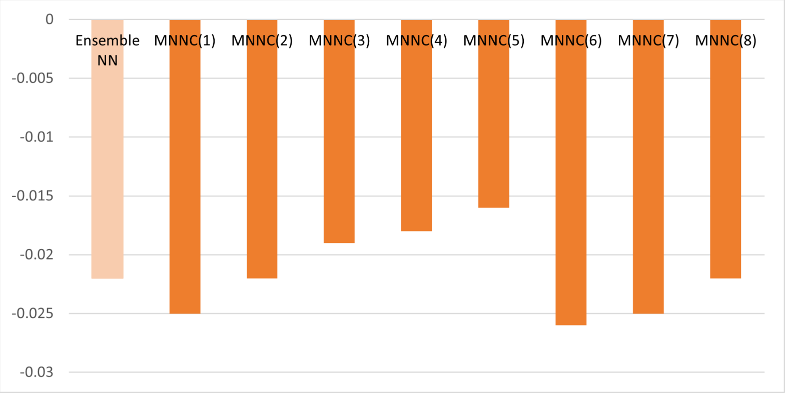


Figure 4.118: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

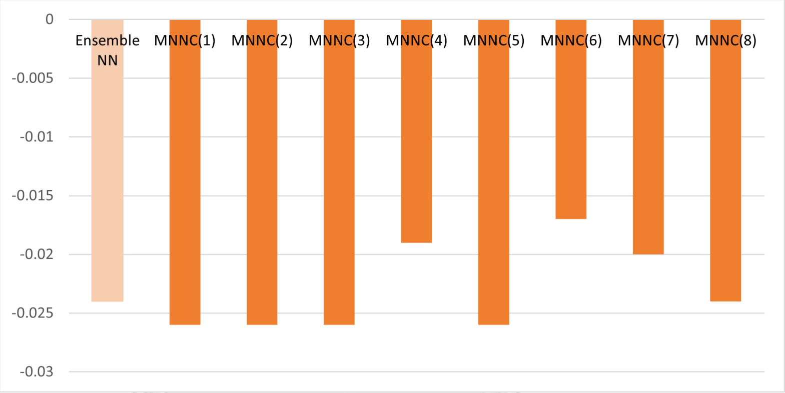


Figure 4.119: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

The accuracy of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.120 to 4.124, respectively.

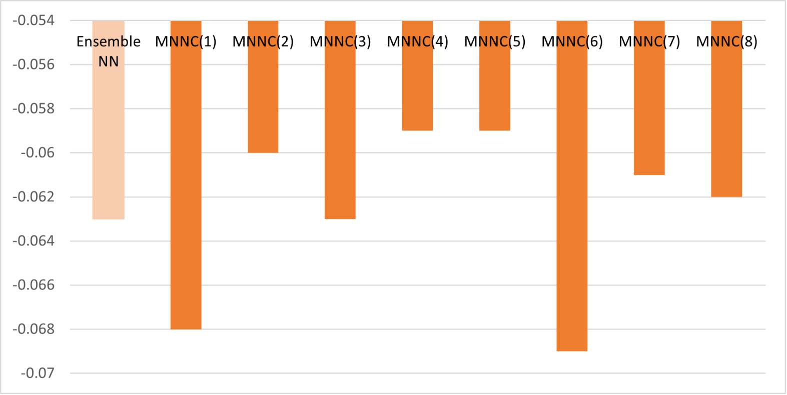


Figure 4.120: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

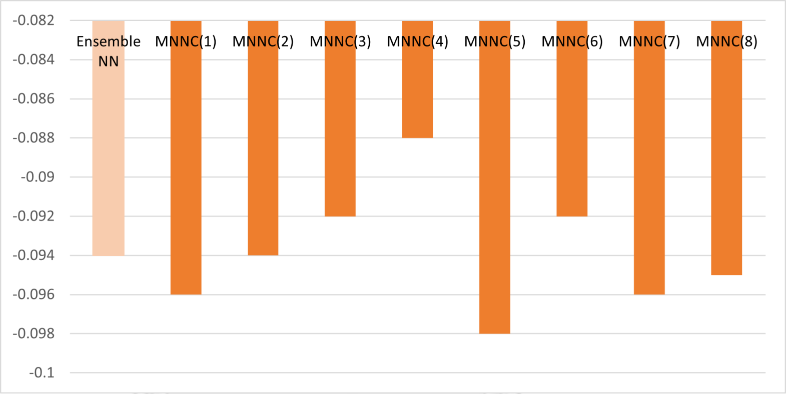


Figure 4.121: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

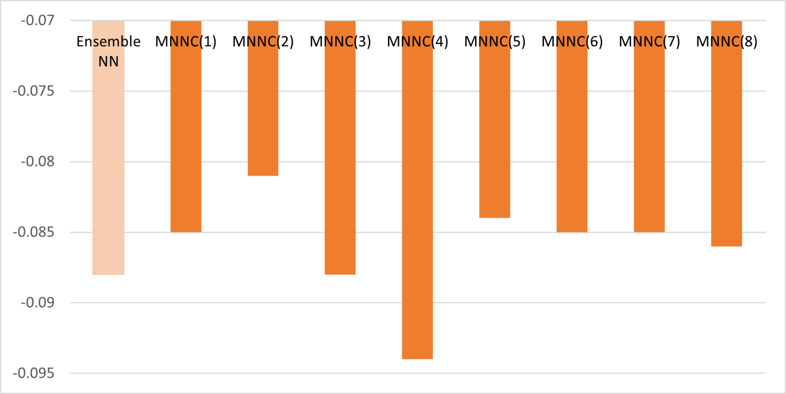


Figure 4.122: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 300 class 1 instances.

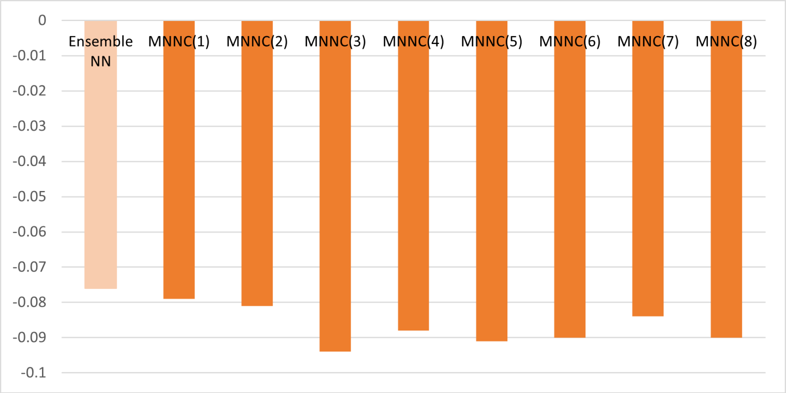


Figure 4.123: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

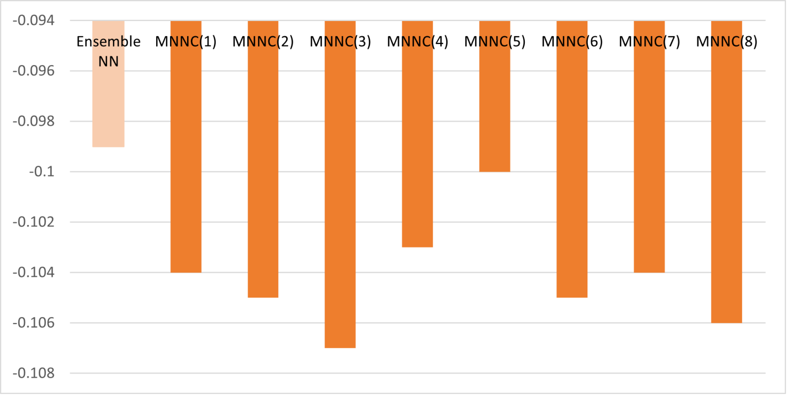


Figure 4.124: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Circle format

The accuracy of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.125 to 4.129, respectively.

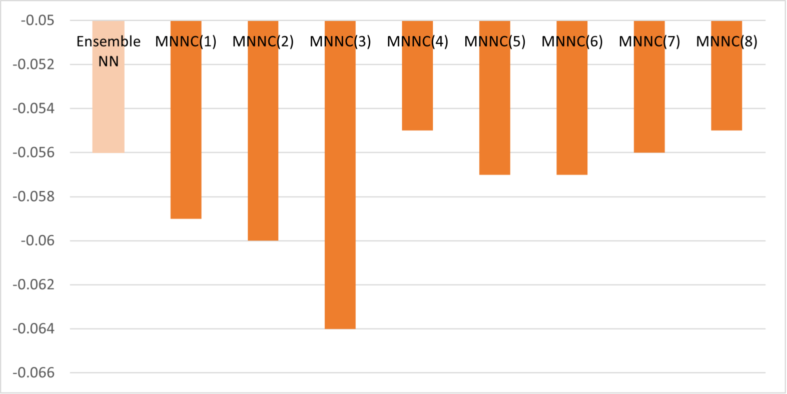


Figure 4.125: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

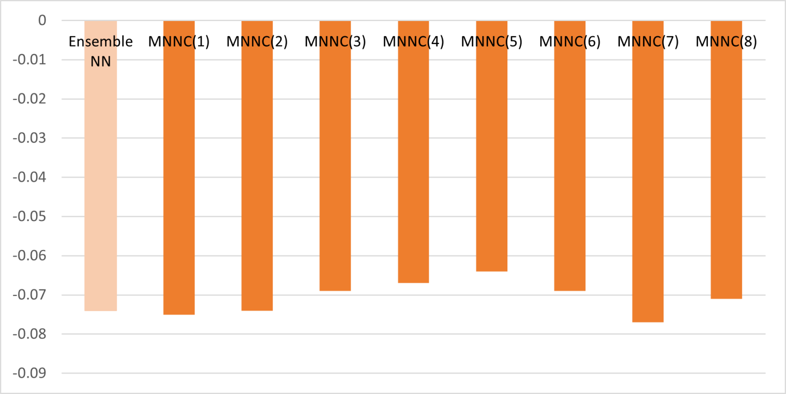


Figure 4.126: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

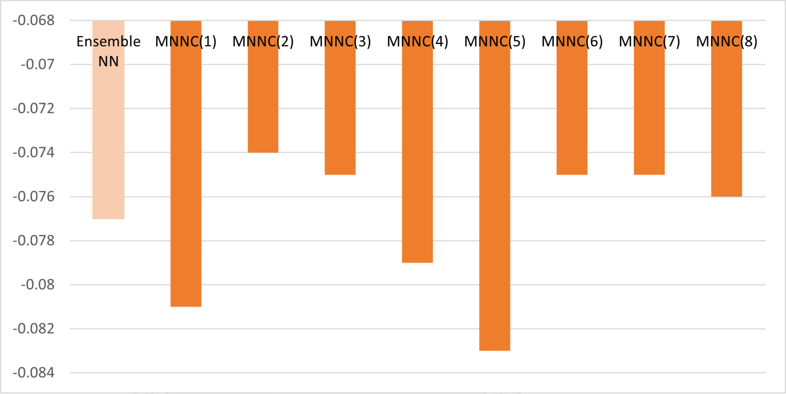


Figure 4.127: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

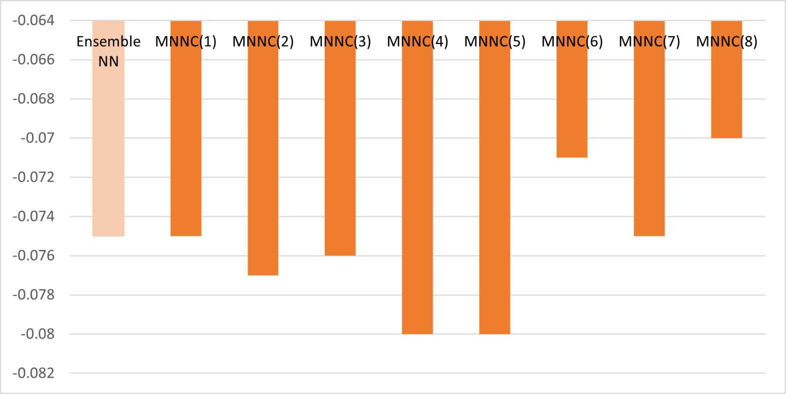


Figure 4.128: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 400 class 1 instances.

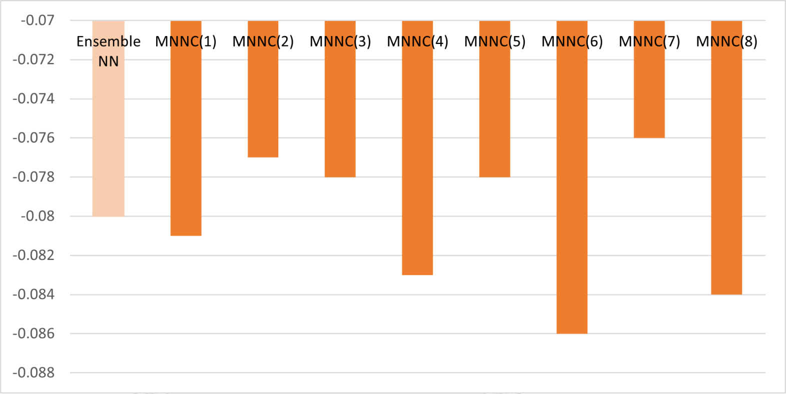


Figure 4.129: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

- Large overlap

  - Gaussian format

    The accuracy of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.130 to 4.134, respectively.

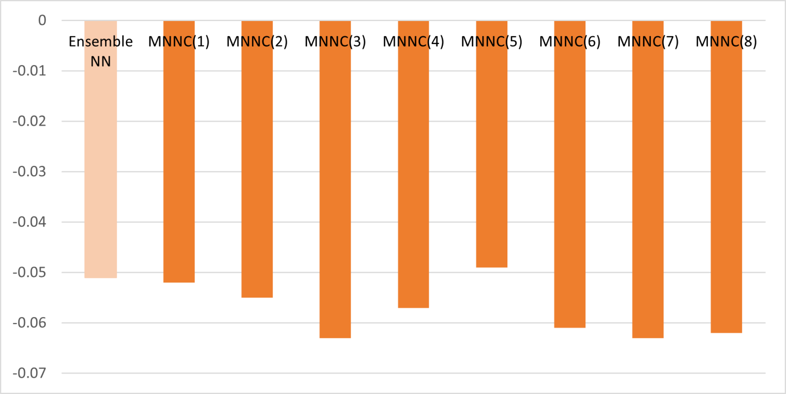


Figure 4.130: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

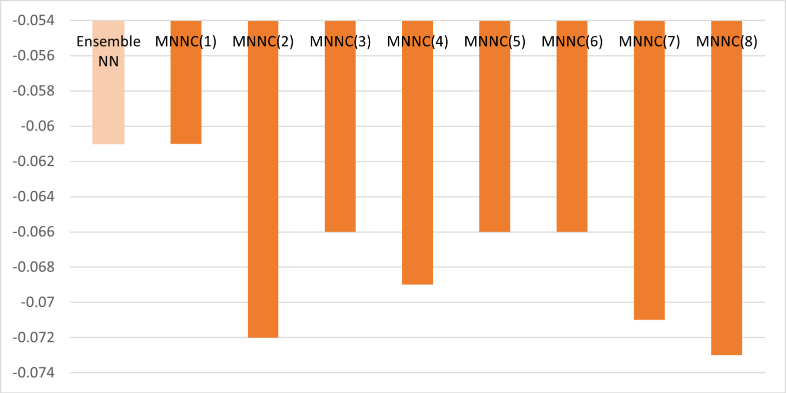


Figure 4.131: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 200 class 1 instances.

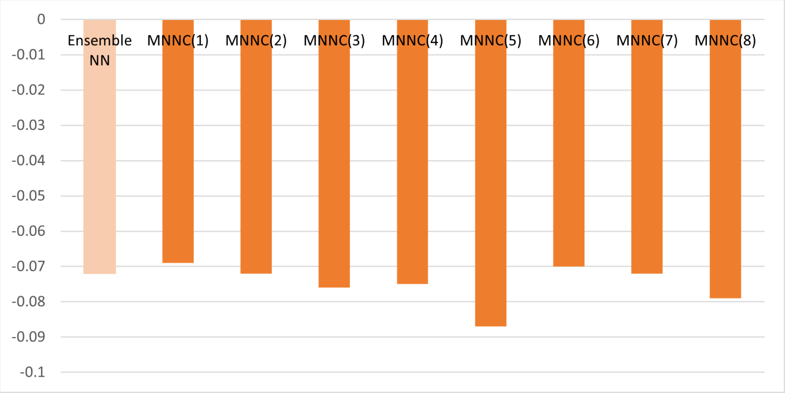


Figure 4.132: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

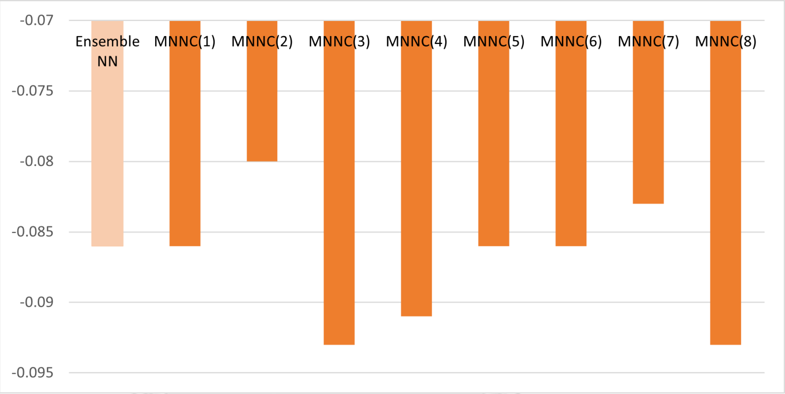


Figure 4.133: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

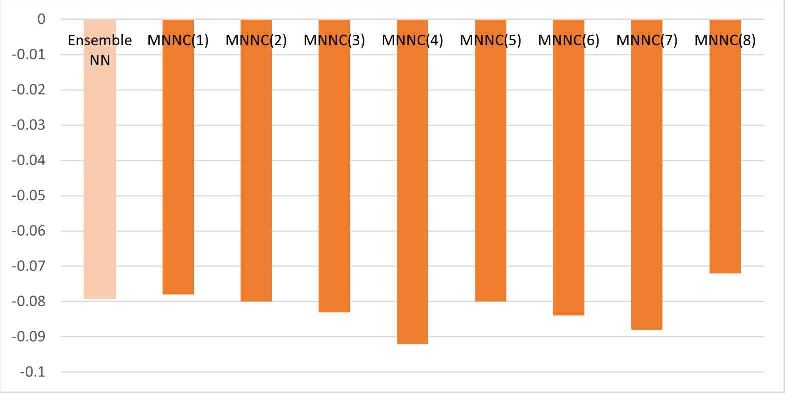


Figure 4.134: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 500 class 1 instances.

– Moon shaped format

The accuracy of both ensemble NN and all versions of MNNC is consistently lower than that of Best k-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.135 to 4.139, respectively.

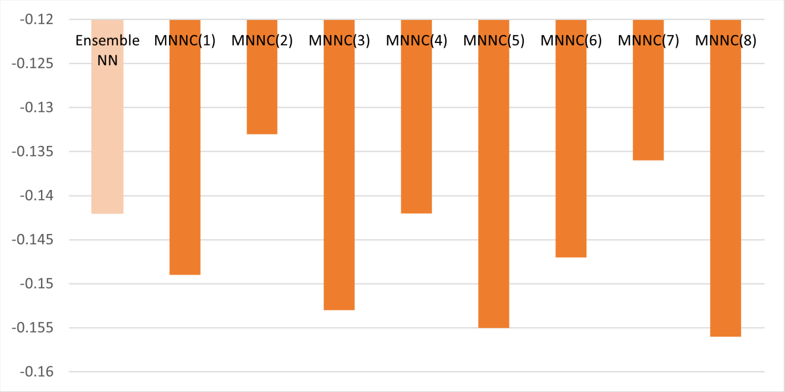


Figure 4.135: Differences in accuracy, between the Ensemble NN and Best k-NN, as well as among various versions of MNNC and Best k-NN for a dataset containing 100 class 1 instances.

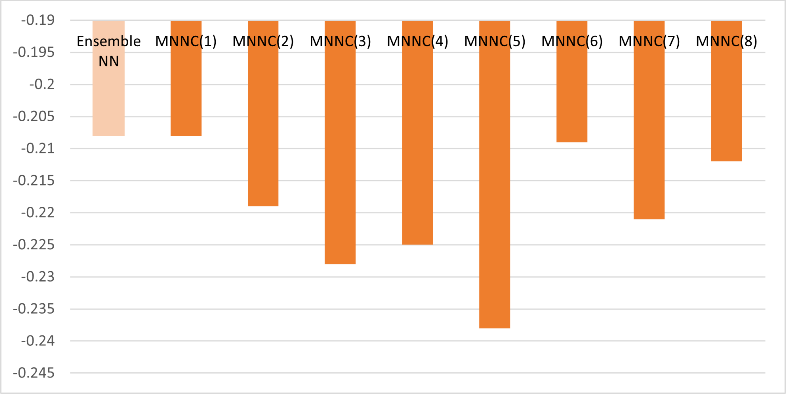


Figure 4.136: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

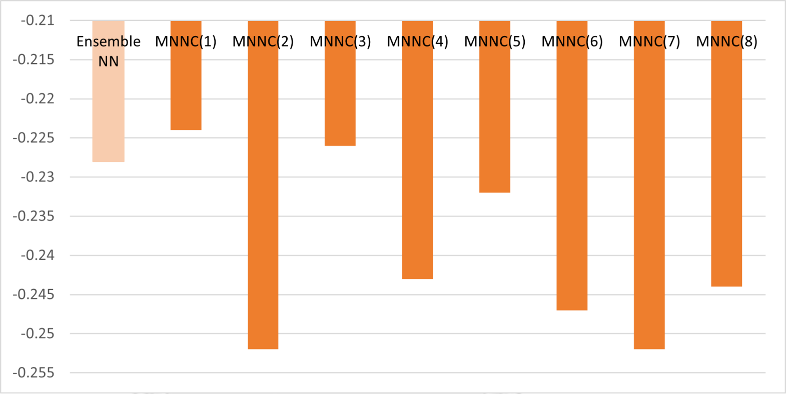


Figure 4.137: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

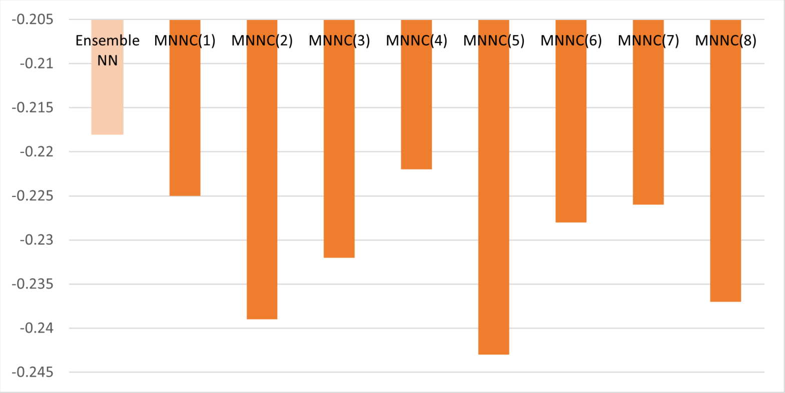


Figure 4.138: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

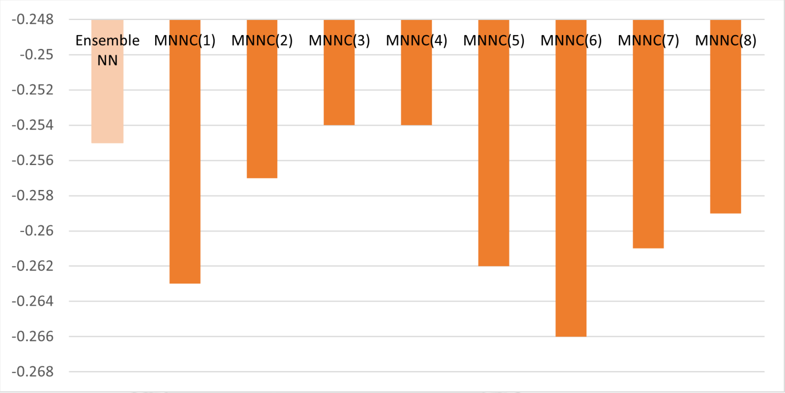


Figure 4.139: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

– Circle format

The accuracy of both ensemble NN and all versions of MNNC is consistently lower than that of Best *k*-NN across varying numbers of class 1 instances, ranging from 100 to 500, as depicted in Figures 4.140 to 4.144, respectively.

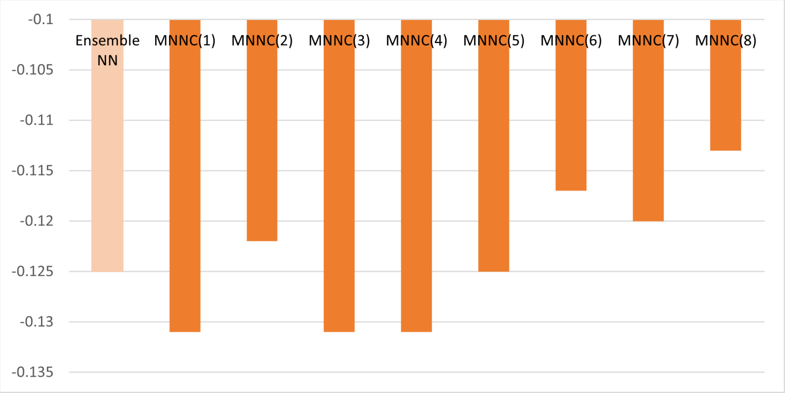


Figure 4.140: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 100 class 1 instances.

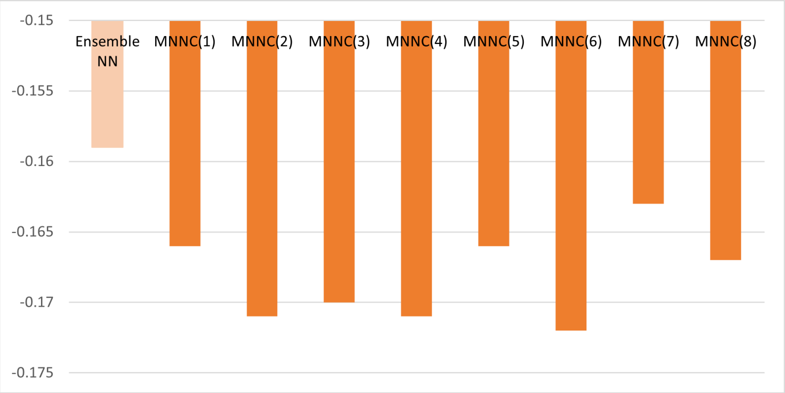


Figure 4.141: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 200 class 1 instances.

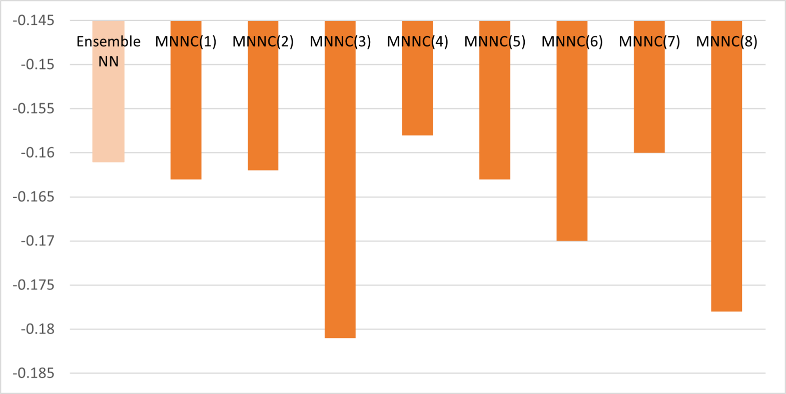


Figure 4.142: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 300 class 1 instances.

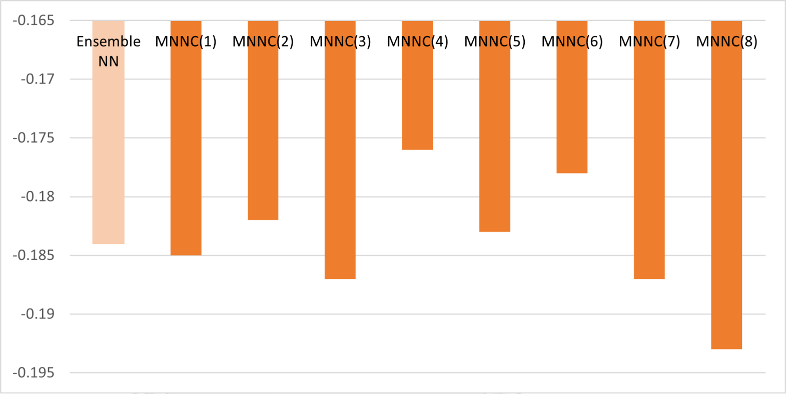


Figure 4.143: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 400 class 1 instances.

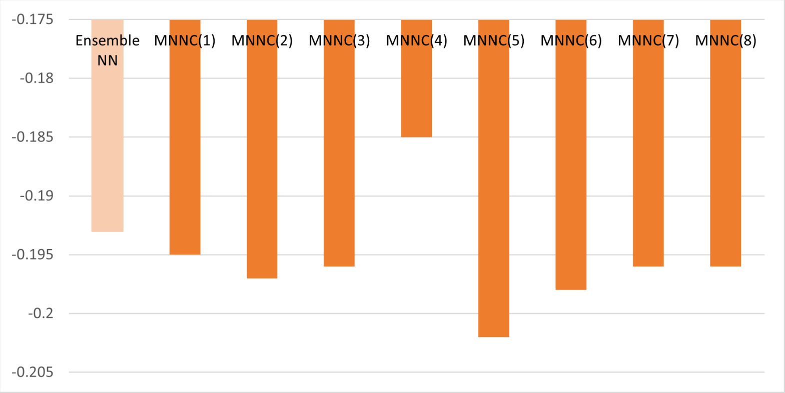


Figure 4.144: Differences in accuracy, between the Ensemble NN and Best *k*-NN, as well as among various versions of MNNC and Best *k*-NN for a dataset containing 500 class 1 instances.

4.2.2 UCI Dataset

In this subsection, the outcomes of the experiments conducted on the UCI dataset by each classifier are outlined across four tables. Specifically, table 4.1 showcases precision, table 4.2 features recall, table 4.3 presents the F1-score, and table 4.4 displays accuracy. The values within each table represent the mean ± standard deviation, derived from 10 repetitions of the experiment.

Data	Best *k*-NN	The ensemble NN	MNNC(1)	MNNC(2)	MNNC(3)	MNNC(4)	MNNC(5)	MNNC(6)	MNNC(7)	MNNC(8)
Wine	0.928±0.029	0.913±0.035	0.904±0.041	0.895±0.041	0.904±0.032	0.885±0.049	0.893±0.027	0.905±0.052	0.903±0.02	0.896±0.059
Sonar	0.848±0.041	0.821±0.057	0.736±0.073	0.765±0.056	0.786±0.074	0.816±0.047	0.807±0.045	0.738±0.053	0.717±0.059	0.713±0.051
Glass	0.869±0.036	0.849±0.037	0.862±0.039	0.867±0.049	0.869±0.052	0.867±0.046	0.869±0.067	0.867±0.061	0.869±0.039	0.865±0.041
Haberman	0.716±0.051	0.673±0.071	0.669±0.08	0.617±0.072	0.612±0.067	0.661±0.091	0.724±0.1	0.632±0.052	0.699±0.061	0.676±0.059
Liver	0.724±0.037	0.701±0.042	0.663±0.036	0.642±0.042	0.636±0.033	0.661±0.038	0.65±0.069	0.659±0.044	0.622±0.039	0.629±0.064
Ionosphere	0.884±0.027	0.865±0.027	0.896±0.023	0.883±0.025	0.896±0.025	0.882±0.04	0.885±0.031	0.889±0.039	0.882±0.04	0.881±0.034
Wholesale	0.891±0.016	0.875±0.018	0.881±0.023	0.845±0.031	0.861±0.037	0.867±0.022	0.855±0.031	0.848±0.036	0.866±0.037	0.87±0.024
Cancer	0.968±0.012	0.96±0.013	0.958±0.019	0.964±0.016	0.955±0.013	0.959±0.015	0.958±0.016	0.951±0.013	0.959±0.011	0.957±0.017
German	0.629±0.021	0.639±0.049	0.624±0.06	0.627±0.03	0.594±0.027	0.634±0.044	0.64±0.039	0.638±0.066	0.636±0.063	0.636±0.044
QSAR	0.798±0.019	0.772±0.02	0.737±0.025	0.768±0.017	0.774±0.022	0.767±0.025	0.769±0.033	0.747±0.016	0.767±0.027	0.756±0.022

Table 4.1: The precision of all version of the MNNC compared to other classifiers in UCI datasets.

Data	Best *k*-NN	The ensemble NN	MNNC(1)	MNNC(2)	MNNC(3)	MNNC(4)	MNNC(5)	MNNC(6)	MNNC(7)	MNNC(8)
Wine	0.921±0.045	0.904±0.045	0.888±0.034	0.908±0.036	0.895±0.042	0.875±0.039	0.875±0.035	0.881±0.048	0.911±0.036	0.902±0.052
Sonar	0.842±0.039	0.81±0.054	0.726±0.069	0.761±0.059	0.77±0.063	0.802±0.045	0.677±0.051	0.727±0.054	0.695±0.053	0.702±0.055
Glass	0.839±0.054	0.777±0.046	0.818±0.043	0.84±0.061	0.844±0.043	0.832±0.04	0.841±0.072	0.825±0.067	0.817±0.07	0.832±0.046
Haberman	0.647±0.028	0.61±0.049	0.597±0.045	0.563±0.033	0.566±0.04	0.591±0.053	0.596±0.045	0.567±0.032	0.592±0.03	0.567±0.036
Liver	0.704±0.031	0.677±0.036	0.651±0.041	0.64±0.042	0.624±0.031	0.645±0.03	0.612±0.064	0.648±0.038	0.612±0.032	0.621±0.056
Ionosphere	0.821±0.03	0.771±0.025	0.827±0.037	0.81±0.05	0.827±0.042	0.801±0.019	0.814±0.061	0.817±0.069	0.806±0.046	0.802±0.036
Wholesale	0.888±0.015	0.869±0.021	0.869±0.024	0.851±0.026	0.867±0.034	0.867±0.027	0.859±0.039	0.876±0.024	0.88±0.041	0.874±0.038
Cancer	0.971±0.014	0.959±0.014	0.961±0.02	0.969±0.015	0.956±0.018	0.959±0.012	0.958±0.014	0.96±0.015	0.965±0.009	0.965±0.015
German	0.607±0.021	0.55±0.02	0.55±0.023	0.555±0.016	0.543±0.014	0.562±0.014	0.559±0.014	0.556±0.023	0.537±0.019	0.563±0.019
QSAR	0.806±0.02	0.787±0.022	0.754±0.029	0.785±0.014	0.79±0.02	0.782±0.022	0.765±0.037	0.761±0.014	0.784±0.026	0.757±0.02

Table 4.2: The recall of all version of the MNNC compared to other classifiers in UCI datasets.

Data	Best *k*-NN	The ensemble NN	MNNC(1)	MNNC(2)	MNNC(3)	MNNC(4)	MNNC(5)	MNNC(6)	MNNC(7)	MNNC(8)
Wine	0.921±0.037	0.904±0.04	0.891±0.033	0.897±0.035	0.897±0.03	0.875±0.041	0.884±0.026	0.893±0.048	0.907±0.029	0.899±0.053
Sonar	0.841±0.04	0.807±0.056	0.723±0.068	0.76±0.057	0.769±0.067	0.802±0.046	0.736±0.052	0.732±0.056	0.706±0.056	0.707±0.056
Glass	0.845±0.05	0.794±0.049	0.829±0.044	0.849±0.053	0.854±0.044	0.843±0.037	0.855±0.07	0.845±0.064	0.842±0.063	0.848±0.039
Haberman	0.662±0.032	0.615±0.063	0.605±0.053	0.559±0.042	0.564±0.04	0.592±0.064	0.654±0.063	0.598±0.042	0.641±0.042	0.617±0.044
Liver	0.704±0.032	0.678±0.038	0.648±0.047	0.638±0.041	0.622±0.03	0.645±0.03	0.63±0.069	0.653±0.04	0.617±0.035	0.625±0.057
Ionosphere	0.837±0.03	0.788±0.028	0.844±0.035	0.825±0.047	0.844±0.038	0.819±0.03	0.848±0.054	0.851±0.069	0.842±0.044	0.84±0.037
Wholesale	0.888±0.012	0.871±0.018	0.873±0.021	0.847±0.028	0.862±0.03	0.865±0.022	0.857±0.033	0.862±0.029	0.873±0.039	0.872±0.03
Cancer	0.969±0.012	0.959±0.013	0.96±0.019	0.967±0.015	0.955±0.013	0.959±0.013	0.958±0.015	0.955±0.014	0.962±0.01	0.961±0.015
German	0.61±0.021	0.526±0.027	0.529±0.03	0.537±0.029	0.526±0.019	0.55±0.022	0.597±0.019	0.594±0.032	0.582±0.023	0.597±0.026
QSAR	0.8±0.019	0.777±0.02	0.742±0.026	0.772±0.016	0.779±0.022	0.772±0.024	0.767±0.034	0.754±0.016	0.775±0.027	0.756±0.026

Table 4.3: The F1-score of all version of the MNNC compared to other classifiers in UCI datasets.

Data	Best *k*-NN	The ensemble NN	MNNC(1)	MNNC(2)	MNNC(3)	MNNC(4)	MNNC(5)	MNNC(6)	MNNC(7)	MNNC(8)
Wine	0.929±0.031	0.913±0.034	0.902±0.028	0.913±0.03	0.911±0.03	0.901±0.041	0.923±0.02	0.902±0.042	0.929±0.025	0.913±0.05
Sonar	0.842±0.04	0.81±0.056	0.727±0.067	0.771±0.059	0.775±0.067	0.808±0.045	0.789±0.05	0.729±0.059	0.756±0.059	0.71±0.054
Glass	0.87±0.048	0.837±0.046	0.859±0.044	0.88±0.04	0.874±0.044	0.874±0.027	0.869±0.046	0.872±0.055	0.861±0.049	0.882±0.032
Haberman	0.79±0.033	0.764±0.053	0.766±0.045	0.723±0.033	0.721±0.04	0.752±0.043	0.762±0.037	0.742±0.038	0.762±0.038	0.738±0.034
Liver	0.722±0.032	0.7±0.034	0.668±0.037	0.655±0.039	0.645±0.03	0.666±0.028	0.631±0.064	0.669±0.041	0.633±0.04	0.645±0.053
Ionosphere	0.858±0.026	0.823±0.025	0.866±0.028	0.85±0.037	0.867±0.038	0.846±0.037	0.867±0.036	0.86±0.05	0.86±0.033	0.853±0.031
Wholesale	0.903±0.013	0.888±0.017	0.891±0.017	0.865±0.024	0.881±0.03	0.88±0.022	0.876±0.031	0.889±0.028	0.899±0.033	0.887±0.025
Cancer	0.973±0.01	0.964±0.011	0.964±0.017	0.97±0.013	0.96±0.013	0.963±0.011	0.961±0.014	0.963±0.012	0.966±0.009	0.968±0.015
German	0.691±0.015	0.698±0.012	0.694±0.018	0.697±0.032	0.686±0.019	0.704±0.02	0.716±0.015	0.703±0.019	0.706±0.019	0.712±0.016
QSAR	0.82±0.016	0.796±0.019	0.762±0.025	0.786±0.017	0.794±0.022	0.789±0.024	0.773±0.031	0.767±0.014	0.789±0.025	0.776±0.022

Table 4.4: The accuracy of all version of the MNNC compared to other classifiers in UCI datasets.

As evident from the data present in Table 4.1-4.4, the precision values for MNNC(1), MNNC(3), and MNNC(7) exceed those of the Best *k*-NN in the Glass dataset. Similarly, in the Haberman dataset, MNNC(5) demonstrates higher precision compared to the Best k-NN. Moving on to the Ionosphere dataset, MNNC(1), MNNC(3), MNNC(5), and MNNC(6) all exhibit higher precision than the Best k-NN. In the German dataset, precision values for MNNC(4), MNNC(5), MNNC(6), MNNC(7), and MNNC(8) surpass those of the Best k-NN.

Remarkably, all versions of MNNC outperform the ensemble NN in both the Glass and Ionosphere datasets. In the case of the Haberman dataset, MNNC(5), MNNC(7), and MNNC(8) exhibit higher precision than the ensemble NN. Furthermore, MNNC(1) surpasses the ensemble NN in the Wholesale dataset, while MNNC(2) outperforms it in the Cancer dataset. For the German dataset, MNNC(4) to MNNC(8) demonstrate higher precision than the ensemble NN. Notably, in the QSAR dataset, MNNC(3) outshines the ensemble NN.

The recall values for MNNC(2), MNNC(3), and MNNC(5) surpass those of the Best k-NN in the Glass dataset. In the Ionosphere dataset, MNNC(1) and MNNC(3) exhibit higher recall compared to the Best k-NN.

The recall values for MNNC(2), and MNNC(7) surpass those of the ensemble NN in the Wine dataset. Impressively, all versions of MNNC outperform the ensemble NN in both the Glass and Ionosphere datasets. Moving on to the Wholesale dataset, MNNC(1), MNNC(6), MNNC(7), and MNNC(8) demonstrate higher recall than the ensemble NN. Similarly, in the Cancer dataset, MNNC(1), MNNC(2), MNNC(4), MNNC(6), MNNC(7), and MNNC(8) outshine the ensemble NN. In the German dataset, MNNC(1), MNNC(2), MNNC(4), MNNC(5), MNNC(6), and MNNC(8) exhibit higher recall than the ensemble NN. Lastly, in the QSAR dataset, MNNC(3) surpasses the recall of the ensemble NN.

The F1-score for MNNC(2), MNNC(3), MNNC(5), MNNC(6), and MNNC(8) exceeds that of the Best k-NN in the Glass dataset. In the Haberman dataset, MNNC(5), and MNNC(7) showcase higher F1-scores compared to the Best k-NN. Turning to the Ionosphere dataset, MNNC(1), MNNC(3), MNNC(5), MNNC(6), MNNC(7), and MNNC(8) all demonstrate higher F1-scores than the Best k-NN.

Notably, in the Wine dataset, MNNC(7) exhibits a higher F1-score than the ensemble NN. Impressively, all versions of MNNC outperform the ensemble NN in the Glass, Ionosphere, and German datasets. In the Haberman dataset, MNNC(5), MNNC(7), and MNNC(8) surpass the ensemble NN in terms of F1-score. Similarly, in the Wholesale dataset, MNNC(1), MNNC(7), and MNNC(8) outshine the ensemble NN. For the Cancer dataset, MNNC(1), MNNC(2), MNNC(4), MNNC(7), and MNNC(8) exhibit higher F1-scores than the ensemble NN.

The accuracy of MNNC(7) surpasses that of the Best k-NN in the Wine dataset. In the Glass dataset, MNNC(2), MNNC(3), MNNC(4), MNNC(6), and MNNC(8) exhibit higher accuracy compared to the Best k-NN. Moving on to the Ionosphere dataset, MNNC(1), MNNC(3), MNNC(5), MNNC(6), and MNNC(7) showcase higher accuracy than the Best k-NN. In the German dataset, all versions of MNNC, except MNNC(3), demonstrate higher accuracy than the Best k-NN.

In the Wine dataset, MNNC(2), MNNC(5), MNNC(7), and MNNC(8) outperform the ensemble NN in terms of accuracy. Similarly, in the Glass and Ionosphere datasets, all versions of MNNC surpass the ensemble NN in accuracy. In the Haberman dataset, MNNC(1) exhibits higher accuracy than the ensemble NN. For the Wholesale dataset, MNNC(1), MNNC(6), MNNC(7), and MNNC(8) showcase higher accuracy compared to the ensemble NN. In the Cancer dataset, MNNC(1), MNNC(2), MNNC(7), and MNNC(8) outshine the ensemble NN. Lastly, in the German dataset, MNNC(4), MNNC(5), MNNC(6), MNNC(7), and MNNC(8) demonstrate higher accuracy than the ensemble NN.

4.3 The discussion of the MOF-guided conglomerate classifier

This chapter introduces the Mass ratio-variance Outlier Factors (MOF)-guided conglomerate nearest neighbor algorithm, a parameter-free approach for determining the number of neighbors. The MOF algorithm is employed to gauge the density of each test instance in the dataset, eliminating the need for manual parameterization. In experiments using synthesized datasets, the proposed algorithm demonstrates performance comparable to both the Best k-NN and the ensemble NN.

For two-class synthesized datasets, including the k-NN algorithm, the ensemble NN algorithm, and the MOF-guided conglomerate nearest neighbor algorithm, all exhibit similar precision, recall, F1-score, and accuracy when classifying unknown instances in the testing set. Although the MOF-guided conglomerate nearest neighbor consistently trails behind the k-NN in performance, particularly in datasets with overlapping or class imbalance issues.

In real-world datasets, the MOF-guided conglomerate nearest neighbor algorithm proves effective in predicting the class of unknown instances, performing at a level comparable to the Best k-NN. Notably, in specific datasets like Glass, Haberman, Ionosphere, and German, the MOF-guided conglomerate nearest neighbor classifier version 5 (MNNC(5)) outperforms the Best k-NN in precision. Furthermore, all versions of the MOF-guided conglomerate nearest neighbor classifier surpass the ensemble NN in terms of precision, recall, F1-score, and accuracy in the Glass and Ionosphere datasets.

The distinguishing feature of the conglomerate nearest neighbor lies in its ability to deliver competitive performance to the Best k-NN without the need for fine-tuning specific parameters, unlike the Best k-NN, which may require parameter optimization for optimal results. Additionally, the conglomerate nearest neighbor algorithm exhibits similar performance to the ensemble NN.

CHAPTER V

CONCLUSION

5.1 Conclusion and Discussion

This thesis introduces two algorithms based on the k nearest neighbor concept, with the objective of determining the optimal number of neighbors for each test instance, a factor influenced by the instance's position. The inclusion of Mass ratio-variance Outlier Factors (MOF), a density-based outlier search, enhances the algorithms.

In the CNNC method, the conglomerate nearest neighbor, the number of neighbors is established during the training phase by calculating MOF for each instance in the training dataset, with separate calculations for each class. The range of MOFs for each class is then segmented, and these segments are evenly divided into the largest integer less than or equal to the square root of the instances in the respective class, assigning values ranging from $k = 1$ to the square root of the number of instances in the respective class.

The inability of CNNC to predict classification results as anticipated may stem from the fact that calculating the MOF at a point in the training dataset, rather than at a point in the datatest set, does not effectively convey the location information of the test set. Consequently, the MNNC method, which calculates the MOF specifically for each test instance, addresses this limitation by providing a more accurate representation of the lacation of test instance test.

In the MNNC method, the MOF-guided conglomerate nearest neighbor, the number of neighbors is determined during the testing phase. When a test instance arrives, MOF is calculated without class separation and is considered in three cases:

In Case 1, when MOF is greater than or equal to 1, a single neighbor is employed, given that a high MOF value suggests the test instance is distantly positioned from other clusters. Opting for a small number of neighbors in this scenario mitigates the risk of inaccurate predictions.

In Case 2, if MOF falls within the range [a, 1), the number of neighbors is set as $\frac{\sqrt{n}}{2}$, with ‘a’ ranging from 0.01 to 0.7 and ‘n’ representing the instances in the training set. This adjustment considers the likelihood that a test instance resides on the periphery of clusters, allowing for more effective predictions with a moderate number of neighbors.

In Case 3, when MOF is less than ‘a’, the number of neighbors is set to $\sqrt{n}$, beneficial for instances within a cluster where using a larger number of neighbors enhances prediction accuracy.

Experimental results on synthesized datasets demonstrate that both methods predict test instance classes with performance similar to the Best k-NN and the ensemble NN. Our proposed method demonstrates predictive performance on par with the Best k-NN in non-overlapping data types. The effectiveness arises from the clear separation of data into distinct groups, a characteristic that significantly enhances the performance of classification in this type of data. In real-world datasets from the UCI dataset, both methods exhibit comparable performance to the Best k-NN and the ensemble NN. MNNC(5) stands out as the optimal choice among all experimental versions. It demonstrates precision that is superior to or equal to the best KNN in four datasets, greater or equal recall in two datasets, and greater or equal accuracy to the best KNN in two datasets, surpassing other MNNC versions.

Importantly, neither method requires parameter determination, addressing the need to choose the number of neighbors in traditional kNN algorithms, where different neighbor numbers yield varying results. In our current information-rich world where data evolves swiftly, the process of identifying optimal parameter values for each dataset could be a time-consuming endeavor. Thus, the absence of parameters in these methods translates to time savings during the parameter tuning phase.

The computational speed of each method varies, ranging from the fastest to the slowest as follows: the Best k-NN, ensemble NN, CNNC, and MNNC. The Best k-NN method is straightforward, selecting the k that yields the highest F1-score among k values from 1, 3, ..., the square root of the training data. The ensemble NN iteratively adjusts k continuously from 1, 3, ..., the square root of the training data, and then votes based on the majority of prediction results, making it more complex than the Best k-NN method. The CNNC method involves calculating the MOF for every instance in the training data, including range division and determining the number of neighbors, contributing to increased processing time. Lastly, MNNC takes the longest time, as it requires recalculating the MOF for each test instance when it is presented.

5.2 Future work

Future research efforts could investigate expanding this study to include multiclass classification, incorporating categorical datasets, and applying it to different metrics, such as weighted-attribute distance.

REFERENCES

[1] E. Alpaydin. *Introduction to machine learning.* MIT Press, Cambridge, USA, 2014.

[2] P. Changsakul, S. Boonsiri, and K. Sinapiromsaran. Mass-ratio-variance based outlier factor. *IEEE*, page 15, 2021.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, (16):321–357, 2002.

[4] G. H. Chen and D. Shah. Explaining the success of nearest neighbor methods in prediction. *Foundations and Trends® in Machine Learning*, 10(5-6):337–588, 2018.

[5] Z. Hajizadeh, M. Taheri, and M. Z. Jahromi. Nearest neighbor classification with locally weighted distance for imbalanced data. *International Journal of Computer and Communication Engineering*, 3(2):81, 2014.

[6] A. B. Hassanat, M. A. Abbadi, G. A. Altarawneh, and A. A. Alhasanat. Solving the problem of the k parameter in the knn classifier using an ensemble learning approach. *International Journal of Computer Science and Information Security*, 12(8):34–39, 2018.

[7] T. Kohonen. The self-organizing map. *IEEE*, 78(9):1464–1480, 1990.

[8] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160(1):3–24, 2007.

[9] E. Kriminger, J. C. Príncipe, and C. Lakshminarayan. Nearest neighbor distributions for imbalanced classification. *IEEE*, pages 1–5, 2012.

[10] Y. Li and X. Zhang. Improving k nearest neighbor with exemplar generalization for imbalanced classification. *Springer*, pages 321–332, 2011.

[11] C. Liu, L. Cao, and S. Y. Philip. A hybrid coupled k-nearest neighbor algorithm on imbalance data. *IEEE*, pages 2011–2018, 2014.

[12] H. C. Mandhare and S. R. Idate. A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques. *IEEE*, pages 931–935, 2017.

[13] S. Ougiaroglou, A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos, and T. Welzer-Druzovec. Adaptive k-nearest-neighbor classification using a dynamic number of nearest neighbors. *Springer*, 11:66–82, 207.

[14] M. Sarkar and T. Y. Leong. Application of k-nearest neighbors algorithm on breast cancer diagnosis problem. *American Medical Informatics Association*, page 759, 2000.

[15] B. Tang and H. He. A local density-based approach for outlier detection. *Neurocomputing*, (241):171–180, 2017.

[16] K. Taunk, S. De, S. Verma, and A. Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. *IEEE*, pages 1255–1260, 2019.

[17] A. Torres-García, O. Mendoza-Montoya, C. Reyes-García, and L. Villaseñor-Pineda. *Biosignal Processing and Classification Using Computational Learning and Intelligence*. Elsevier, Amsterdam, Netherlands, 2021.

[18] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng. Learning k for knn classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(3):1–19, 2017.

[19] X. F. Zhong, S. Z. Guo, L. Gao, H. Shan, and J. H. Zheng. An improved k-nn classification with dynamic k. *Proceedings of the 9th International Conference on Machine Learning and Computing*, 9:211–216, 2017.

APPENDICES

APPENDIX A : The result of synthesized dataset in chapter 3, conglomerate nearest neighbor classifier.

No overlap

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	1	1	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1	1	1
500	300	1	1	0.998±0.002	1	1	0.999±0.001	1	1	0.998±0.002	1	1	0.999±0.002
500	400	1	0.999±0.001	0.999±0.001	1	0.999±0.001	0.999±0.001	1	0.999±0.001	0.999±0.001	1	0.999±0.001	0.999±0.001
500	500	1	0.999±0.002	1	1	0.999±0.002	1	1	0.999±0.002	1	1	0.999±0.002	1

Table 1: The precision, recall, F1-score, and accuracy of CNNC in Gaussian with no overlap compared to other classifiers.

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.998±0.002	0.998±0.002	0.998±0.002	0.992±0.011	0.989±0.014	0.991±0.011	0.995±0.006	0.993±0.009	0.995±0.006	0.997±0.003	0.997±0.004	0.997±0.003
500	200	1	0.999±0.001	0.999±0.001	1	0.998±0.004	0.99±0.004	1	0.998±0.002	0.998±0.002	1	0.999±0.002	0.999±0.002
500	300	1	0.999±0.002	0.998±0.003	1	0.999±0.001	0.999±0.002	1	0.999±0.002	0.999±0.002	1	0.999±0.001	0.999±0.002
500	400	1	0.999±0.001	1	1	0.999±0.002	1	1	0.999±0.001	1	1	0.999±0.001	1
500	500	1	0.998±0.002	0.997±0.002	1	0.998±0.002	0.997±0.003	1	0.999±0.002	0.997±0.003	1	0.998±0.002	0.997±0.003

Table 2: The precision, recall, F1-score, and accuracy of CNNC in moon shaped with no overlap compared to other classifiers.

Synthesized data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	1	1	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1	1	1
500	500	1	1	1	1	1	1	1	1	1	1	1	1

Table 3: The precision, recall, F1-score, and accuracy of CNNC in circle with no overlap compared to other classifiers.

Slight overlap

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.979±0.015	0.978±0.014	0.976±0.014	0.988±0.013	0.984±0.012	0.975±0.019	0.983±0.012	0.981±0.010	0.975±0.013	0.990±0.007	0.989±0.006	0.986±0.008
500	200	0.978±0.011	0.978±0.012	0.976±0.012	0.981±0.010	0.976±0.011	0.973±0.011	0.979±0.010	0.977±0.010	0.974±0.013	0.98±0.008	0.982±0.008	0.98±0.011
500	300	0.980±0.010	0.978±0.010	0.972±0.009	0.980±0.011	0.975±0.013	0.972±0.010	0.980±0.010	0.976±0.011	0.972±0.009	0.982±0.009	0.978±0.010	0.97±0.008
500	400	0.987±0.007	0.981±0.009	0.980±0.008	0.988±0.007	0.980±0.010	0.979±0.008	0.987±0.007	0.981±0.009	0.980±0.008	0.988±0.007	0.981±0.009	0.980±0.008
500	500	0.986±0.006	0.979±0.008	0.978±0.011	0.986±0.006	0.979±0.007	0.978±0.011	0.986±0.006	0.979±0.008	0.978±0.011	0.986±0.007	0.979±0.008	0.978±0.011

Table 4: The precision, recall, F1-score, and accuracy of CNNC in Gaussian with slight overlap compared to other classifiers.

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.926±0.031	0.904±0.035	0.907±0.036	0.899±0.030	0.866±0.042	0.875±0.045	0.911±0.025	0.881±0.031	0.888±0.033	0.952±0.013	0.938±0.015	0.941±0.016
500	200	0.901±0.017	0.893±0.019	0.891±0.024	0.892±0.020	0.884±0.023	0.879±0.023	0.896±0.018	0.888±0.019	0.884±0.022	0.918±0.013	0.912±0.014	0.909±0.016
500	300	0.919±0.014	0.909±0.016	0.911±0.013	0.915±0.014	0.900±0.00716	0.905±0.015	0.916±0.013	0.903±0.015	0.906±0.014	0.921±0.013	0.909±0.015	0.912±0.014
500	400	0.916±0.023	0.909±0.023	0.906±0.022	0.916±0.021	0.90±0.0208	0.90±0.0205	0.915±0.022	0.908±0.022	0.905±0.021	0.916±0.022	0.909±0.021	0.906±0.021
500	500	0.908±0.020	0.897±0.017	0.900±0.016	0.908±0.020	0.89±0.016	0.900±0.016	0.908±0.020	0.897±0.017	0.900±0.016	0.908±0.020	0.897±0.016	0.900±0.016

Table 5: The precision, recall, F1-score, and accuracy of CNNC in moon shaped with slight overlap compared to other classifiers.

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.915±0.026	0.910±0.023	0.903±0.028	0.933±0.028	0.907±0.037	0.903±0.041	0.924±0.026	0.908±0.028	0.902±0.033	0.957±0.017	0.95±0.017	0.947±0.021
500	200	0.919±0.017	0.911±0.018	0.908±0.014	0.911±0.019	0.902±0.021	0.902±0.023	0.914±0.017	0.906±0.019	0.904±0.017	0.933±0.012	0.926±0.014	0.925±0.013
500	300	0.924±0.014	0.916±0.017	0.916±0.019	0.927±0.011	0.917±0.013	0.917±0.014	0.925±0.012	0.916±0.014	0.916±0.016	0.93±0.012	0.922±0.014	0.922±0.015
500	400	0.929±0.013	0.920±0.013	0.917±0.013	0.927±0.013	0.917±0.012	0.917±0.013	0.927±0.012	0.918±0.012	0.916±0.013	0.928±0.012	0.92±0.011	0.917±0.012
500	500	0.928±0.018	0.921±0.016	0.920±0.019	0.928±0.018	0.922±0.016	0.920±0.018	0.928±0.018	0.921±0.016	0.920±0.018	0.928±0.018	0.921±0.016	0.920±0.018

Table 6: The precision, recall, F1-score, and accuracy of CNNC in circle with slight overlap compared to other classifiers.

Large overlap

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.931±0.030	0.919±0.037	0.909±0.027	0.931±0.019	0.904±0.032	0.91±0.031	0.931±0.023	0.911±0.032	0.911±0.027	0.962±0.013	0.952±0.017	0.950±0.015
500	200	0.942±0.022	0.932±0.024	0.925±0.021	0.931±0.024	0.911±0.029	0.911±0.028	0.936±0.020	0.921±0.025	0.916±0.022	0.947±0.018	0.935±0.021	0.932±0.019
500	300	0.924±0.018	0.915±0.022	0.921±0.019	0.924±0.019	0.912±0.022	0.920±0.020	0.923±0.018	0.913±0.022	0.920±0.020	0.928±0.017	0.919±0.021	0.925±0.019
500	400	0.923±0.016	0.916±0.013	0.919±0.018	0.923±0.014	0.913±0.014	0.917±0.016	0.923±0.015	0.915±0.013	0.917±0.017	0.924±0.015	0.917±0.013	0.919±0.017
500	500	0.914±0.014	0.908±0.014	0.91±0.012	0.914±0.013	0.907±0.014	0.909±0.012	0.914±0.013	0.907±0.014	0.909±0.012	0.914±0.013	0.907±0.014	0.909±0.011

Table 7: The precision, recall, F1-score, and accuracy of CNNC in Gaussian with large overlap compared to other classifiers.

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.788±0.052	0.767±0.058	0.747±0.05	0.714±0.054	0.65±0.044	0.662±0.036	0.734±0.046	0.676±0.048	0.684±0.036	0.865±0.029	0.852±0.023	0.847±0.023
500	200	0.766±0.019	0.708±0.024	0.767±0.026	0.729±0.016	0.708±0.018	0.727±0.027	0.742±0.015	0.721±0.018	0.741±0.027	0.803±0.016	0.791±0.015	0.803±0.022
500	300	0.764±0.027	0.761±0.025	0.754±0.031	0.751±0.028	0.736±0.025	0.734±0.03	0.755±0.029	0.743±0.026	0.74±0.032	0.777±0.031	0.771±0.027	0.766±0.034
500	400	0.785±0.017	0.774±0.014	0.779±0.022	0.779±0.019	0.765±0.016	0.773±0.024	0.78±0.018	0.767±0.016	0.774±0.023	0.784±0.017	0.772±0.013	0.779±0.021
500	500	0.758±0.013	0.744±0.014	0.746±0.017	0.759±0.012	0.743±0.015	0.746±0.017	0.758±0.013	0.741±0.015	0.745±0.017	0.758±0.012	0.742±0.015	0.745±0.017

Table 8: The precision, recall, F1-score, and accuracy of CNNC in moon shaped with large overlap compared to other classifiers.

Data		Precision			Recall			F1-score			Accuracy		
#C0	#C1	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	Ensemble NN	CNNC	Best *k*-NN	The ensemble NN	CNNC
500	100	0.818±0.045	0.824±0.069	0.81±0.057	0.77±0.04	0.716±0.048	0.712±0.04	0.788±0.035	0.748±0.043	0.742±0.038	0.889±0.018	0.881±0.018	0.877±0.019
500	200	0.82±0.043	0.811±0.036	0.803±0.045	0.783±0.037	0.757±0.038	0.756±0.042	0.796±0.037	0.773±0.037	0.77±0.042	0.839±0.041	0.827±0.038	0.823±0.043
500	300	0.831±0.019	0.823±0.018	0.819±0.025	0.818±0.022	0.805±0.019	0.807±0.024	0.823±0.02	0.811±0.018	0.811±0.024	0.837±0.017	0.827±0.017	0.826±0.021
500	400	0.83±0.021	0.816±0.024	0.821±0.02	0.832±0.022	0.815±0.025	0.822±0.02	0.83±0.021	0.815±0.024	0.821±0.019	0.832±0.021	0.817±0.024	0.823±0.02
500	500	0.833±0.023	0.821±0.026	0.819±0.029	0.832±0.023	0.82±0.026	0.817±0.029	0.832±0.023	0.819±0.026	0.816±0.029	0.832±0.023	0.82±0.026	0.817±0.029

Table 9: The precision, recall, F1-score, and accuracy of CNNC in circle with large overlap compared to other classifiers.

APPENDIX B : The precision of synthesized dataset in chapter 4, MOF-guided conglomerate nearest neighbor classifier.

No overlap

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1±0
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	0.999±0.002	0.999±0.002	0.998±0.002	0.999±0.002	0.999±0.002	1	0.999±0.002	0.999±0.002

Table 10: The precision of MNNC in Gaussian with no overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.998±0.002	0.998±0.003	0.998±0.002	0.999±0.002	0.999±0.002	0.998±0.003	0.996±0.004	0.999±0.002	0.998±0.002	0.998±0.004
500	200	1	1	0.999±0.002	1.002	1	1	0.998±0.002	0.998±0.002	1	1
500	300	1	1	0.997±0.003	1	0.999±0.002	0.999±0.002	0.999±0.002	1	0.999±0.002	0.998±0.003
500	400	1	1	0.998±0.003	0.999±0.002	1	1	0.998±0.002	0.998±0.003	0.999±0.002	1
500	500	1	0.998±0.002	0.997±0.003	0.999±0.002	0.999±0.002	0.998±0.003	0.998±0.002	0.998±0.003	0.999±0.002	0.999±0.002

Table 11: The precision of MNNC in moon shaped with no overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	1	1	1	1	1	1	1	1

Table 12: The precision of MNNC in circle with no overlap compared to other classifiers.

Slight overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.984±0.016	0.982±0.018	0.979±0.018	0.977±0.018	0.967±0.017	0.979±0.015	0.977±0.021	0.979±0.015	0.982±0.012	0.975±0.02
500	200	0.985±0.007	0.979±0.009	0.981±0.008	0.985±0.011	0.979±0.008	0.984±0.01	0.982±0.007	0.986±0.011	0.979±0.008	0.98±0.008
500	300	0.98±0.01	0.974±0.011	0.972±0.011	0.98±0.01	0.974±0.011	0.98±0.01	0.981±0.006	0.977±0.01	0.975±0.01	0.979±0.015
500	400	0.986±0.008	0.978±0.009	0.975±0.012	0.978±0.009	0.98±0.011	0.982±0.01	0.974±0.009	0.983±0.008	0.974±0.009	0.979±0.011
500	500	0.983±0.007	0.976±0.008	0.974±0.008	0.974±0.006	0.974±0.008	0.981±0.005	0.983±0.005	0.974±0.011	0.98±0.01	0.977±0.01

Table 13: The precision of MNNC in Gaussian with slight overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.914±0.033	0.896±0.032	0.901±0.042	0.896±0.04	0.89±0.046	0.907±0.042	0.902±0.034	0.911±0.042	0.9±0.041	0.893±0.043
500	200	0.9±0.018	0.894±0.018	0.89±0.026	0.895±0.019	0.888±0.022	0.895±0.02	0.886±0.027	0.881±0.018	0.886±0.025	0.888±0.018
500	300	0.918±0.02	0.907±0.022	0.911±0.022	0.917±0.016	0.905±0.017	0.9±0.021	0.907±0.014	0.908±0.014	0.911±0.011	0.908±0.028
500	400	0.93±0.019	0.923±0.018	0.92±0.014	0.918±0.02	0.905±0.014	0.911±0.018	0.91±0.013	0.907±0.012	0.915±0.013	0.909±0.015
500	500	0.911±0.01	0.901±0.009	0.896±0.01	0.896±0.02	0.893±0.011	0.898±0.027	0.894±0.02	0.9±0.025	0.896±0.02	0.894±0.016

Table 14: The precision of MNNC in moon shaped with slight overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.925±0.035	0.903±0.05	0.899±0.05	0.897±0.03	0.891±0.016	0.905±0.029	0.889±0.038	0.898±0.034	0.908±0.028	0.899±0.035
500	200	0.922±0.022	0.916±0.023	0.915±0.017	0.916±0.031	0.923±0.025	0.926±0.025	0.917±0.019	0.929±0.028	0.907±0.019	0.917±0.019
500	300	0.924±0.011	0.916±0.013	0.912±0.011	0.92±0.015	0.916±0.015	0.915±0.014	0.92±0.017	0.91±0.02	0.918±0.021	0.916±0.02
500	400	0.933±0.009	0.923±0.012	0.922±0.012	0.923±0.026	0.923±0.01	0.919±0.01	0.93±0.015	0.919±0.016	0.924±0.014	0.93±0.019
500	500	0.929±0.011	0.921±0.012	0.92±0.015	0.923±0.013	0.922±0.012	0.918±0.012	0.914±0.011	0.921±0.008	0.925±0.007	0.916±0.014

Table 15: The precision of MNNC in circle with slight overlap compared to other classifiers.

Large overlap

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.931±0.032	0.915±0.041	0.914±0.045	0.925±0.032	0.897±0.03	0.899±0.026	0.911±0.038	0.912±0.04	0.894±0.036	0.914±0.032
500	200	0.933±0.018	0.924±0.023	0.925±0.021	0.922±0.021	0.924±0.028	0.93±0.018	0.928±0.009	0.931±0.013	0.913±0.021	0.913±0.026
500	300	0.934±0.016	0.927±0.016	0.93±0.015	0.923±0.023	0.92±0.012	0.922±0.019	0.927±0.018	0.91±0.02	0.925±0.012	0.916±0.015
500	400	0.923±0.014	0.915±0.011	0.915±0.01	0.92±0.012	0.908±0.015	0.908±0.011	0.912±0.025	0.913±0.013	0.917±0.013	0.906±0.01
500	500	0.93±0.012	0.922±0.013	0.922±0.015	0.921±0.016	0.917±0.01	0.909±0.01	0.916±0.011	0.92±0.025	0.913±0.015	0.928±0.012

Table 16: The precision of MNNC in Gaussian with large overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.781±0.064	0.76±0.06	0.733±0.053	0.751±0.063	0.714±0.074	0.766±0.071	0.77±0.086	0.732±0.05	0.787±0.074	0.751±0.069
500	200	0.764±0.032	0.747±0.027	0.746±0.038	0.732±0.045	0.731±0.049	0.737±0.04	0.749±0.025	0.726±0.04	0.733±0.031	0.756±0.025
500	300	0.787±0.031	0.758±0.03	0.763±0.041	0.738±0.038	0.756±0.035	0.738±0.031	0.74±0.021	0.758±0.02	0.734±0.024	0.742±0.027
500	400	0.795±0.025	0.783±0.026	0.776±0.029	0.759±0.017	0.768±0.026	0.775±0.026	0.77±0.018	0.753±0.04	0.773±0.027	0.762±0.026
500	500	0.758±0.013	0.746±0.019	0.738±0.02	0.744±0.029	0.745±0.015	0.747±0.029	0.735±0.02	0.738±0.011	0.74±0.02	0.741±0.025

Table 17: The precision of MNNC in moon shaped with large overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.806±0.036	0.794±0.042	0.781±0.036	0.789±0.055	0.795±0.031	0.795±0.051	0.802±0.035	0.79±0.055	0.778±0.032	0.782±0.063
500	200	0.835±0.029	0.818±0.028	0.809±0.029	0.813±0.043	0.802±0.033	0.797±0.038	0.795±0.033	0.809±0.034	0.814±0.026	0.793±0.04
500	300	0.847±0.024	0.831±0.018	0.829±0.031	0.832±0.03	0.817±0.02	0.839±0.022	0.824±0.04	0.831±0.025	0.83±0.03	0.814±0.024
500	400	0.833±0.028	0.815±0.03	0.814±0.034	0.816±0.016	0.809±0.023	0.823±0.016	0.823±0.015	0.817±0.028	0.81±0.023	0.806±0.019
500	500	0.823±0.022	0.807±0.025	0.805±0.026	0.803±0.022	0.804±0.017	0.815±0.015	0.803±0.012	0.799±0.022	0.804±0.015	0.805±0.014

Table 18: The precision of MNNC in circle with large overlap compared to other classifiers.

APPENDIX C : The recall of synthesized dataset in chapter 4, MOF-guided conglomerate nearest neighbor classifier.

No overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	0.999±0.002	0.999±0.002	0.998±0.002	0.999±0.002	0.999±0.002	1	0.999±0.002	0.999±0.002

Table 19: The recall of MNNC in Gaussian with no overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.993±0.009	0.991±0.013	0.991±0.009	0.995±0.01	0.996±0.009	0.991±0.013	0.991±0.017	0.995±0.008	0.992±0.01	0.991±0.015
500	200	1	1	0.998±0.005	0.999±0.002	1	1	0.999±0.003	0.996±0.005	1	0.999±0.003
500	300	1	1	0.997±0.003	1	1	1	0.999±0.002	1	1	0.999±0.002
500	400	1	1	0.998±0.003	1	1	1	1	0.998±0.002	0.999±0.002	1
500	500	1	0.998±0.002	0.997±0.003	0.999±0.002	0.999±0.002	0.998±0.003	0.999±0.002	0.998±0.003	0.999±0.002	0.999±0.001

Table 20: The recall of MNNC in moon shaped with no overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	1	1	1	1	1	1	1	1

Table 21: The recall of MNNC in circle with no overlap compared to other classifiers.

Slight overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.983±0.014	0.983±0.014	0.971±0.019	0.967±0.028	0.97±0.019	0.979±0.013	0.982±0.014	0.972±0.029	0.97±0.022	0.97±0.026
500	200	0.988±0.009	0.981±0.009	0.981±0.009	0.984±0.01	0.977±0.012	0.985±0.01	0.988±0.014	0.979±0.009	0.981±0.011	0.982±0.007
500	300	0.979±0.01	0.972±0.013	0.97±0.013	0.978±0.012	0.974±0.012	0.978±0.009	0.975±0.01	0.979±0.009	0.976±0.005	0.98±0.013
500	400	0.986±0.008	0.978±0.011	0.974±0.013	0.977±0.009	0.981±0.012	0.982±0.01	0.984±0.008	0.974±0.009	0.975±0.009	0.978±0.011
500	500	0.983±0.007	0.977±0.008	0.975±0.008	0.974±0.006	0.974±0.008	0.982±0.005	0.974±0.011	0.983±0.005	0.98±0.01	0.977±0.01

Table 22: The recall of MNNC in Gaussian with slight overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.895±0.025	0.875±0.03	0.847±0.045	0.883±0.055	0.877±0.052	0.876±0.043	0.869±0.027	0.85±0.042	0.874±0.028	0.871±0.043
500	200	0.895±0.021	0.882±0.018	0.882±0.025	0.877±0.016	0.89±0.033	0.893±0.036	0.879±0.028	0.875±0.035	0.88±0.024	0.876±0.026
500	300	0.917±0.019	0.904±0.021	0.907±0.021	0.912±0.022	0.907±0.021	0.898±0.026	0.911±0.012	0.911±0.014	0.91±0.013	0.908±0.022
500	400	0.932±0.019	0.924±0.018	0.921±0.015	0.917±0.021	0.904±0.014	0.912±0.019	0.91±0.013	0.91±0.013	0.916±0.015	0.91±0.014
500	500	0.912±0.01	0.901±0.01	0.896±0.01	0.895±0.02	0.893±0.012	0.897±0.026	0.9±0.025	0.896±0.02	0.896±0.02	0.893±0.015

Table 23: The recall of MNNC in moon shaped with slight overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.913±0.035	0.89±0.037	0.886±0.035	0.892±0.025	0.883±0.03	0.897±0.04	0.904±0.042	0.895±0.038	0.896±0.03	0.907±0.023
500	200	0.912±0.019	0.904±0.021	0.903±0.018	0.902±0.035	0.906±0.025	0.915±0.03	0.914±0.022	0.912±0.026	0.9±0.015	0.909±0.018
500	300	0.929±0.012	0.922±0.017	0.918±0.014	0.923±0.014	0.923±0.018	0.917±0.015	0.913±0.019	0.922±0.022	0.92±0.019	0.922±0.02
500	400	0.935±0.007	0.925±0.008	0.926±0.008	0.922±0.026	0.923±0.009	0.919±0.009	0.918±0.017	0.927±0.017	0.925±0.013	0.929±0.018
500	500	0.929±0.012	0.92±0.012	0.918±0.016	0.923±0.013	0.922±0.012	0.918±0.011	0.922±0.011	0.914±0.01	0.924±0.008	0.916±0.014

Table 24: The recall of MNNC in circle with slight overlap compared to other classifiers.

Large overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.913±0.037	0.894±0.048	0.886±0.053	0.873±0.041	0.879±0.028	0.886±0.042	0.894±0.031	0.872±0.043	0.882±0.022	0.864±0.027
500	200	0.927±0.02	0.922±0.021	0.921±0.026	0.903±0.02	0.915±0.039	0.899±0.03	0.912±0.013	0.908±0.023	0.91±0.034	0.906±0.025
500	300	0.928±0.015	0.918±0.016	0.923±0.013	0.922±0.019	0.916±0.021	0.917±0.02	0.906±0.018	0.924±0.017	0.922±0.015	0.916±0.015
500	400	0.92±0.012	0.912±0.01	0.911±0.009	0.92±0.013	0.905±0.018	0.907±0.012	0.913±0.014	0.913±0.026	0.916±0.014	0.906±0.011
500	500	0.93±0.011	0.921±0.013	0.922±0.014	0.92±0.015	0.918±0.018	0.908±0.01	0.92±0.025	0.917±0.012	0.912±0.014	0.927±0.013

Table 25: The recall of MNNC in Gaussian with large overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.712±0.033	0.667±0.027	0.647±0.04	0.665±0.046	0.66±0.058	0.667±0.035	0.661±0.045	0.632±0.052	0.683±0.046	0.645±0.03
500	200	0.724±0.031	0.712±0.033	0.713±0.04	0.692±0.028	0.689±0.036	0.696±0.025	0.682±0.033	0.707±0.021	0.694±0.025	0.713±0.04
500	300	0.765±0.026	0.741±0.027	0.745±0.033	0.72±0.037	0.741±0.036	0.724±0.032	0.741±0.025	0.725±0.028	0.719±0.022	0.728±0.022
500	400	0.79±0.025	0.776±0.024	0.769±0.027	0.756±0.015	0.763±0.029	0.774±0.025	0.751±0.039	0.766±0.021	0.769±0.027	0.759±0.027
500	500	0.758±0.011	0.746±0.017	0.737±0.017	0.743±0.028	0.746±0.016	0.747±0.029	0.737±0.011	0.735±0.02	0.74±0.021	0.741±0.025

Table 26: The recall of MNNC in moon shaped with large overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.756±0.038	0.731±0.046	0.723±0.044	0.725±0.033	0.705±0.028	0.716±0.049	0.723±0.053	0.733±0.063	0.732±0.044	0.737±0.057
500	200	0.792±0.03	0.776±0.027	0.768±0.03	0.768±0.024	0.763±0.039	0.765±0.026	0.771±0.024	0.774±0.027	0.778±0.028	0.772±0.034
500	300	0.83±0.022	0.816±0.012	0.814±0.029	0.816±0.028	0.8±0.021	0.82±0.027	0.82±0.03	0.809±0.041	0.821±0.029	0.802±0.02
500	400	0.834±0.027	0.816±0.029	0.815±0.032	0.813±0.015	0.808±0.021	0.822±0.016	0.816±0.03	0.819±0.017	0.811±0.023	0.805±0.019
500	500	0.822±0.022	0.807±0.024	0.805±0.026	0.803±0.022	0.803±0.018	0.815±0.016	0.799±0.022	0.801±0.01	0.805±0.014	0.804±0.014

Table 27: The recall of MNNC in circle with large overlap compared to other classifiers.

APPENDIX D : The F1-score of synthesized dataset in chapter 4, MOF-guided conglomerate nearest neighbor classifier.

No overlap

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	0.999±0.002	0.999±0.002	0.998±0.002	0.999±0.002	0.999±0.002	1	0.999±0.002	0.999±0.002

Table 28: The F1-score of MNNC in Gaussian with no overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.995±0.006	0.994±0.008	0.994±0.006	0.997±0.006	0.997±0.005	0.994±0.008	0.997±0.005	0.988±0.013	0.995±0.006	0.994±0.011
500	200	1	1	0.998±0.003	0.999±0.003	1	1	0.997±0.004	0.997±0.003	1	0.999±0.002
500	300	1	1	0.997±0.003	1	0.999±0.002	1	1	0.999±0.002	0.999±0.002	0.998±0.003
500	400	1	1	0.998±0.003	1	1	1	0.998±0.002	0.998±0.002	0.999±0.002	1
500	500	1	1	0.997±0.003	0.999±0.002	0.999±0.002	0.998±0.003	0.998±0.003	0.998±0.002	0.999±0.002	0.999±0.002

Table 29: The F1-score of MNNC in moon shaped with no overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1±0
500	200	1	1	1	1	1	1	1	1±0	1	1±0
500	300	1	1	1	1	1	1	1	1±0	1	1±0
500	400	1	1	1	1	1	1	1	1±0	1	1±0
500	500	1	1	1	1	1	1	1	1±0	1	1±0

Table 30: The F1-score of MNNC in circle with no overlap compared to other classifiers.

Slight overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.983±0.012	0.982±0.014	0.975±0.016	0.972±0.022	0.968±0.01	0.979±0.01	0.98±0.011	0.974±0.021	0.976±0.015	0.972±0.018
500	200	0.986±0.007	0.98±0.008	0.981±0.008	0.984±0.01	0.978±0.009	0.985±0.009	0.987±0.011	0.98±0.007	0.98±0.009	0.981±0.005
500	300	0.979±0.01	0.973±0.012	0.971±0.011	0.979±0.011	0.974±0.011	0.979±0.009	0.976±0.01	0.98±0.007	0.975±0.008	0.98±0.014
500	400	0.986±0.008	0.978±0.01	0.974±0.013	0.977±0.009	0.981±0.011	0.982±0.01	0.983±0.008	0.974±0.009	0.975±0.009	0.978±0.011
500	500	0.983±0.007	0.976±0.008	0.974±0.008	0.974±0.006	0.974±0.008	0.981±0.005	0.974±0.011	0.983±0.005	0.98±0.01	0.976±0.01

Table 31: The F1-score of MNNC in Gaussian with slight overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.902±0.02	0.883±0.02	0.868±0.038	0.887±0.044	0.879±0.042	0.888±0.034	0.886±0.021	0.871±0.03	0.885±0.027	0.88±0.039
500	200	0.897±0.018	0.887±0.016	0.885±0.024	0.884±0.014	0.888±0.024	0.893±0.027	0.879±0.023	0.88±0.031	0.882±0.023	0.881±0.022
500	300	0.918±0.019	0.906±0.021	0.909±0.021	0.914±0.019	0.906±0.019	0.899±0.024	0.909±0.012	0.908±0.013	0.91±0.011	0.908±0.025
500	400	0.931±0.019	0.923±0.018	0.92±0.014	0.917±0.021	0.904±0.014	0.911±0.019	0.908±0.012	0.909±0.013	0.915±0.014	0.909±0.014
500	500	0.911±0.01	0.9±0.01	0.895±0.01	0.895±0.021	0.893±0.011	0.897±0.026	0.899±0.025	0.894±0.021	0.896±0.02	0.893±0.015

Table 32: The F1-score of MNNC in moon shaped with slight overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.918±0.032	0.895±0.037	0.89±0.035	0.892±0.013	0.886±0.019	0.9±0.031	0.898±0.027	0.89±0.033	0.886±0.023	0.902±0.024
500	200	0.915±0.017	0.908±0.019	0.907±0.014	0.908±0.032	0.914±0.023	0.92±0.026	0.921±0.024	0.914±0.02	0.911±0.027	0.912±0.018
500	300	0.926±0.01	0.918±0.014	0.914±0.011	0.922±0.015	0.919±0.016	0.915±0.013	0.911±0.019	0.921±0.019	0.923±0.013	0.918±0.02
500	400	0.933±0.008	0.923±0.01	0.924±0.011	0.922±0.026	0.923±0.01	0.919±0.01	0.918±0.017	0.928±0.016	0.916±0.014	0.929±0.019
500	500	0.929±0.011	0.92±0.012	0.919±0.016	0.923±0.013	0.921±0.012	0.917±0.012	0.921±0.012	0.914±0.01	0.912±0.015	0.916±0.014

Table 33: The F1-score of MNNC in circle with slight overlap compared to other classifiers.

Large overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.921±0.033	0.903±0.043	0.899±0.048	0.895±0.032	0.886±0.026	0.89±0.026	0.901±0.027	0.888±0.039	0.882±0.052	0.884±0.017
500	200	0.929±0.016	0.922±0.02	0.922±0.021	0.911±0.018	0.919±0.034	0.912±0.025	0.921±0.011	0.917±0.017	0.902±0.027	0.909±0.023
500	300	0.931±0.015	0.922±0.016	0.926±0.013	0.922±0.021	0.918±0.022	0.919±0.019	0.907±0.017	0.925±0.017	0.917±0.022	0.915±0.013
500	400	0.921±0.012	0.913±0.01	0.912±0.009	0.92±0.012	0.906±0.018	0.907±0.012	0.912±0.014	0.913±0.025	0.907±0.027	0.906±0.01
500	500	0.929±0.012	0.921±0.013	0.921±0.015	0.92±0.015	0.917±0.018	0.908±0.01	0.92±0.025	0.916±0.011	0.914±0.021	0.927±0.012

Table 34: The F1-score of MNNC in Gaussian with large overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.733±0.031	0.693±0.026	0.67±0.043	0.688±0.045	0.673±0.055	0.694±0.041	0.681±0.046	0.658±0.065	0.688±0.032	0.67±0.034
500	200	0.737±0.032	0.723±0.033	0.724±0.041	0.705±0.032	0.7±0.04	0.709±0.028	0.722±0.035	0.72±0.021	0.721±0.026	0.724±0.037
500	300	0.772±0.027	0.746±0.028	0.751±0.035	0.724±0.037	0.746±0.036	0.729±0.03	0.745±0.024	0.728±0.025	0.724±0.03	0.732±0.023
500	400	0.791±0.025	0.777±0.024	0.77±0.027	0.757±0.016	0.764±0.029	0.773±0.025	0.751±0.039	0.766±0.019	0.75±0.023	0.76±0.027
500	500	0.757±0.012	0.744±0.018	0.736±0.018	0.742±0.028	0.745±0.016	0.746±0.028	0.737±0.011	0.733±0.019	0.743±0.016	0.74±0.025

Table 35: The F1-score of MNNC in moon shaped with large overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.776±0.035	0.753±0.042	0.744±0.039	0.746±0.033	0.735±0.026	0.739±0.046	0.756±0.052	0.747±0.053	0.748±0.032	0.767±0.057
500	200	0.808±0.029	0.79±0.026	0.782±0.029	0.783±0.027	0.776±0.034	0.777±0.029	0.782±0.028	0.784±0.025	0.791±0.026	0.784±0.037
500	300	0.836±0.022	0.821±0.014	0.82±0.03	0.821±0.028	0.805±0.021	0.826±0.025	0.814±0.041	0.823±0.027	0.824±0.03	0.806±0.02
500	400	0.833±0.028	0.815±0.03	0.813±0.033	0.814±0.015	0.808±0.022	0.821±0.015	0.819±0.016	0.815±0.03	0.81±0.023	0.805±0.019
500	500	0.822±0.022	0.807±0.025	0.805±0.026	0.802±0.022	0.803±0.018	0.815±0.016	0.801±0.011	0.797±0.021	0.803±0.016	0.804±0.014

Table 36: The F1-score of MNNC in circle with large overlap compared to other classifiers.

APPENDIX E : The accuracy of synthesized dataset in chapter 4, MOF-guided conglomerate nearest neighbor classifier.

No overlap

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	0.999±0.002	0.999±0.002	0.998±0.002	0.999±0.002	0.999±0.002	1	0.999±0.002	0.999±0.002

Table 37: The accuracy of MNNC in Gaussian with no overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.997±0.003	0.997±0.005	0.997±0.004	0.999±0.003	0.999±0.003	0.997±0.005	0.998±0.003	0.994±0.007	0.997±0.003	0.997±0.007
500	200	1	1	0.999±0.002	0.999±0.001	1	1	0.998±0.003	0.998±0.003	1	0.999±0.002
500	300	1	1	0.998±0.003	1	1	1	1	1	1	0.999±0.002
500	400	1	1	0.998±0.003	1	1	1	0.998±0.002	0.998±0.002	0.999±0.002	1
500	500	1	0.998±0.002	0.997±0.003	0.999±0.002	0.999±0.002	0.998±0.003	0.998±0.003	0.998±0.002	0.999±0.002	0.999±0.002

Table 38: The accuracy of MNNC in moon shaped with no overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	1	1	1	1	1	1	1	1	1	1
500	200	1	1	1	1	1	1	1	1	1	1
500	300	1	1	1	1	1	1	1	1	1	1
500	400	1	1	1	1	1	1	1	1	1	1
500	500	1	1	1	1	1	1	1	1	1	1

Table 39: The accuracy of MNNC in circle with no overlap compared to other classifiers.

Slight overlap

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.991±0.007	0.99±0.008	0.986±0.009	0.985±0.011	0.984±0.005	0.989±0.006	0.989±0.006	0.986±0.011	0.987±0.008	0.985±0.009
500	200	0.989±0.006	0.985±0.006	0.985±0.006	0.987±0.008	0.982±0.007	0.987±0.008	0.99±0.009	0.984±0.006	0.984±0.008	0.984±0.005
500	300	0.981±0.009	0.975±0.011	0.973±0.011	0.98±0.01	0.976±0.011	0.98±0.009	0.978±0.009	0.982±0.007	0.977±0.007	0.981±0.013
500	400	0.986±0.008	0.978±0.01	0.975±0.013	0.978±0.009	0.981±0.011	0.982±0.01	0.984±0.008	0.974±0.009	0.975±0.009	0.978±0.011
500	500	0.983±0.007	0.976±0.008	0.974±0.008	0.974±0.006	0.974±0.008	0.981±0.005	0.974±0.011	0.983±0.005	0.98±0.01	0.976±0.01

Table 40: The accuracy of MNNC in Gaussian with slight overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.947±0.012	0.937±0.012	0.932±0.018	0.94±0.024	0.937±0.023	0.941±0.016	0.941±0.013	0.931±0.016	0.939±0.016	0.938±0.02
500	200	0.913±0.017	0.906±0.015	0.904±0.022	0.906±0.014	0.908±0.025	0.912±0.022	0.902±0.018	0.908±0.023	0.904±0.019	0.905±0.018
500	300	0.923±0.018	0.912±0.02	0.915±0.02	0.919±0.018	0.912±0.017	0.906±0.021	0.916±0.011	0.915±0.013	0.915±0.011	0.914±0.023
500	400	0.932±0.019	0.924±0.018	0.921±0.014	0.919±0.02	0.906±0.014	0.912±0.019	0.909±0.012	0.91±0.013	0.916±0.013	0.91±0.013
500	500	0.911±0.01	0.901±0.01	0.896±0.01	0.895±0.021	0.893±0.011	0.897±0.026	0.9±0.025	0.895±0.021	0.896±0.02	0.894±0.016

Table 41: The accuracy of MNNC in moon shaped with slight overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.957±0.017	0.944±0.02	0.941±0.02	0.94±0.01	0.936±0.008	0.945±0.017	0.943±0.018	0.943±0.014	0.944±0.012	0.945±0.014
500	200	0.931±0.016	0.926±0.018	0.925±0.014	0.926±0.023	0.931±0.018	0.933±0.022	0.936±0.022	0.931±0.017	0.923±0.015	0.929±0.015
500	300	0.93±0.009	0.923±0.014	0.919±0.012	0.926±0.013	0.925±0.014	0.921±0.012	0.917±0.017	0.925±0.017	0.925±0.018	0.924±0.017
500	400	0.935±0.008	0.925±0.01	0.925±0.011	0.923±0.025	0.924±0.01	0.92±0.01	0.92±0.016	0.929±0.016	0.925±0.013	0.93±0.019
500	500	0.929±0.011	0.92±0.012	0.919±0.016	0.923±0.013	0.922±0.012	0.917±0.012	0.922±0.011	0.914±0.01	0.924±0.008	0.916±0.014

Table 42: The accuracy of MNNC in circle with slight overlap compared to other classifiers.

Large overlap

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.959±0.018	0.949±0.024	0.948±0.024	0.945±0.016	0.937±0.018	0.943±0.011	0.951±0.015	0.939±0.021	0.937±0.013	0.938±0.013
500	200	0.945±0.011	0.939±0.013	0.939±0.013	0.928±0.017	0.934±0.022	0.931±0.018	0.934±0.009	0.934±0.015	0.929±0.021	0.927±0.018
500	300	0.936±0.015	0.928±0.015	0.931±0.013	0.928±0.019	0.924±0.02	0.925±0.018	0.913±0.016	0.93±0.016	0.928±0.012	0.921±0.011
500	400	0.922±0.012	0.914±0.01	0.914±0.009	0.92±0.012	0.907±0.018	0.909±0.011	0.914±0.014	0.914±0.024	0.917±0.013	0.907±0.01
500	500	0.93±0.012	0.921±0.012	0.922±0.015	0.92±0.015	0.917±0.018	0.908±0.01	0.92±0.025	0.916±0.011	0.912±0.015	0.928±0.012

Table 43: The accuracy of MNNC in Gaussian with large overlap compared to other classifiers.

#C0	#C1	Best *k*-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.869±0.019	0.858±0.018	0.851±0.015	0.867±0.021	0.847±0.025	0.858±0.021	0.845±0.029	0.853±0.037	0.864±0.025	0.844±0.028
500	200	0.802±0.023	0.792±0.023	0.792±0.029	0.781±0.029	0.772±0.039	0.775±0.028	0.762±0.034	0.791±0.021	0.779±0.022	0.788±0.031
500	300	0.796±0.026	0.772±0.027	0.776±0.034	0.748±0.034	0.774±0.029	0.757±0.026	0.768±0.025	0.753±0.024	0.748±0.021	0.756±0.02
500	400	0.795±0.025	0.782±0.025	0.775±0.027	0.761±0.016	0.768±0.03	0.778±0.025	0.757±0.037	0.772±0.018	0.774±0.026	0.763±0.027
500	500	0.758±0.012	0.745±0.018	0.737±0.018	0.743±0.027	0.746±0.016	0.746±0.028	0.738±0.011	0.734±0.019	0.739±0.02	0.741±0.026

Table 44: The accuracy of MNNC in moon shaped with large overlap compared to other classifiers.

#C0	#C1	Best k-NN	The ensemble NN	MCCN (1)	MCCN (2)	MCCN (3)	MCCN (4)	MCCN (5)	MCCN (6)	MCCN (7)	MCCN (8)
500	100	0.882±0.029	0.875±0.026	0.869±0.025	0.878±0.018	0.869±0.018	0.869±0.023	0.875±0.03	0.883±0.03	0.88±0.02	0.887±0.027
500	200	0.854±0.021	0.841±0.018	0.834±0.021	0.829±0.026	0.83±0.022	0.829±0.022	0.834±0.016	0.828±0.028	0.837±0.022	0.833±0.031
500	300	0.852±0.02	0.839±0.015	0.837±0.028	0.838±0.025	0.819±0.019	0.842±0.025	0.837±0.026	0.83±0.035	0.84±0.028	0.822±0.018
500	400	0.834±0.027	0.816±0.029	0.815±0.033	0.818±0.015	0.813±0.02	0.824±0.015	0.817±0.029	0.822±0.016	0.813±0.021	0.807±0.02
500	500	0.823±0.022	0.807±0.024	0.805±0.026	0.803±0.022	0.804±0.018	0.815±0.015	0.798±0.021	0.802±0.011	0.804±0.015	0.804±0.014

Table 45: The accuracy of MNNC in circle with large overlap compared to other classifiers.

BIOGRAPHY

Name	Ms. Patcharasiri Fuangfoo
Date of Birth	October 23, 1998
Place of Birth	Chiangrai, Thailand
Educations	B.Sc. (Mathematics) (Second Class Honours), Chulalongkorn University, 2020
Scholarships	H.M. the King Bhumibhol Adulyadej's 72nd Birthday Anniversary Scholarship