جامعة الإسراء

Isra University

Department of Information Technology

PREDICTION OF MISSING DATA TECHNIQUE TO IMPROVE BIG DATA CLASSIFICATION

Prepared By

Huda Hussain

Supervised By

Dr. Aysh Alhroob

This Thesis is submitted to the Faculty of Information Technology as a Partial Fulfilment of the Requirement for Master Degree in

Software Engineering

August, 2020

The undersigned have examined the thesis entitled "Prediction of Missing Data Technique to Improve Big Data Classification" presented by Student Name, a candidate for the degree Master of Science in Software Engineering and hereby certify that it is worthy of acceptance.

عايش الحروب 31-8-2020 Dr. Aysh Albroob

Date

د. إيــاد الزبيــدي

Ayad 2/9/2020 Dr. Ayad Alzobaydi

Date

Dr. Ghaith M Jaradat

Date 6/9/2020

جامعة الاسراء

إقرار تفويض

أنا هدى حسين ـ افوض جامعة الاسراء بتزويد نسخ من رسالتي ورقيا والكترونيا للمكتبات او المنظمات او الهيئات او المؤسسات المعنية بالأبحاث والدراسات العليا عند طلبها.

هدى حسين

التوقيع : huda. hussein. m

التاريخ : 6/9/2020

AUTHORIZATION STATEMENT

I Huda Hussain, authorize Isra University to provide hard copies or soft copies of my thesis to libraries, institutions, or individuals upon their request.

Huda Hussain

Signature: huda. hussein. m

Date: 6/9/2020

الإهداء

الى من فاضت علي~ من رافدي~ عطائها وحبها واظلت علي~ بسعاف ثقتها الى من حملتني على جناحي النجاح والكفاح

وحلقت بي الى جنان التفوق المعلقات لأتعلم ان ليس كل سقوط هو نهاية الطريق

بل هو بداية لحلمي الجديد دمتي لي دائما وابدآ ملجئي واماني وفخري

الى امي......

....

الى من علمني ان أبهى النجاح

هو الوصول الى نهاية طريق الصعاب

وان ازدياد العلم هو السر لبريق الجمال

الى من ملك اطيب القلوب وهو اشد الرجال

واورثني أشرف الانساب وأعظم الاحلام

لروحك ابي......

....

الى الروح التي علمتني ان الحضور قد يعني الغياب المكتمل او الرحيل الابدي

الذي لا تقاطعه عودة ولا يشوبه وصال الى ملهمي وهدفي الى من هو شموخي

وفخري وعزي وعزتي وخطى دربي لتحقيق احلامي دامت كلماتك قناديل تضيء لي دربي

لروحك عمي.....

الشكر والامتنان

﴿ يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ بِمَا تَعْمَلُونَ خَبِيرٌ ﴾

المجادلة (11)

﴿الْحَمْدُ لِلَّهِ الَّذِي أَنْزَلَ عَلَىٰ عَبْدِهِ الْكِتَابَ وَلَمْ يَجْعَلْ لَهُ عِوَجًا ﴾

الكهف (1)

اشكر ربي الذي انعم علي بلطفه وكرمة ونعامه وتوفيقه فله الحمد دائما وابد واسئلة التوفيق الدائم

ولمشرفي الدكتور القدير الذي اشاد الي وافاض علي~ من معالمه ومعرفته ودعمه الكبير الدكتور (عايش الحروب)

لكل من منحني من علمه وازاد الي بالمعرفة ...اساتذتي الدكاترة الكرام

لمن علموني ان مرارة الصعاب

تحلوها دفئ قلوبهم

وعثرات الطريق

اتجاوزها بتشجيعهم

والايام المظلمة

قد تلونها امالهم بمستقبلي

الذين كانوا وسيبقون رفقاء الروح والاخوة

الذين ولدتهم لي ارحام الايام

الى اخي كرار وصديقاتي واصدقائي..

DEDICATION

To my mother the origin of my success,

To the soul of my father and my uncle who taught me the meaning of life,

To my brother,

May Allah bless them …

ACKNOWLEDGMENT

My greatest adoration and thanks to The Great Almighty God who enabled me to complete this act of faith. To my family, thank you for encouraging me in all my pursuits and inspiring me to follow my dreams. I am especially grateful to my mother, who supported me emotionally. I always knew that you believed in me and wanted the best for me. Thank you for teaching me that my job in life was to learn, to be happy, and to know and understand myself; only then could I know and understand others. For her endless love, prayers, sacrifice, and support to pursue and successfully complete my master study.

Foremost, I would like to express my sincere gratitude to my supervisor Dr. Aysh Alhroob, for taking out time to ensure qualitative supervision of this research. Thanks a lot for your support and frequent feedback.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABREVIATIONS

ANN	Artificial Neural Network
DD	Diabetes Disease
GMM	Gaussian Mixture Modelling
KDD	Knowledge Discovery in Databases
DM	Data Mining
MV	Missing Value
PIDD	Pima Indian Diabetes Disease
MAR	Missing at Random
FA	Firefly Algorithm
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
KNN	K Nearest Neighbors
SVM	Support Vector Machine
NBC	Naive Bayesian Classifier
RBF	Radial Basis Function
MSE	Mean Square Error
RBF	Radial Basis Function
LR	Linear Regression
NB	Naive Bayesian
NP	Nondeterministic Polynomial time

ABSTRACT

Designing an early prediction systems-based machine learning model (for diabetes disease) is an emerging research area, increasing day by day due to the increasing of the diabetes cases all around the world. Missing values in medical datasets in general, and diabetes disease in particular is an issue faces the machine learning models and case studies. The imputation method is needed for estimating the missing values is a preprocessing step, should be implemented before classifying the cases in the dataset. In this study, a new imputation algorithm based on Firefly Algorithm (FA) is proposed, which is called Imputation Algorithm based Firefly Algorithm (IFA). In order to evaluate the proposed IFA algorithm, a classifier is needed as a fitness function, which generates the classification accuracy of the generated dataset and should be maximized. Therefore, the accuracy is obtained using three different classifiers: K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayesian Classifier (NBC). Pima Indian Diabetes Disease (PIDD) is the main dataset used in this study for estimating the missing values and evaluate IFA. The proposed algorithm is evaluated using two types of experiments, first experiments validated the generated datasets using k-fold cross validation (K=5). While the second experiment the validation is done using holdout validation, where the generated dataset is divided into training set (65%) and testing set (35%). The obtained results showed that the IFA-SVM was ranked the best based the average of ten run times, while IFA-NBC ranked the worst. Moreover, IFA with all classifiers had the best accuracies as compared to the four popular techniques, which proved that the optimization algorithm as an imputation algorithm is better than the statistical methods in this study. In conclusion, FA algorithm can be used for estimating missing values PIDD and medical datasets in general.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND

Data mining (DM), also known as Knowledge Discovery in Databases (KDD), is *"the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"* Data is generated every day in great amounts as millions of people generate large amount of data over the web through various means, such as social media, banks applications, mobile applications, governmental offices, university portals, etc. With the large network for connected devices, the volume of data grows exponentially and the organization of this big data volume and its pre-processing for automatic extraction of useful information gives rise to a new branch of science called (DM) [1], [2].

During DM processes, the quality of the considered data determines the quality of its outcome; hence, data pre-processing is an important step towards achieving clean and quality data and determines the success of the mining process. Data pre-processing is the major step in KDD process as it decreases data complexity and gives better conditions to subsequent data analysis. Data pre-processing aids in understanding the nature of the data, thereby allowing accurate and efficient data analysis. The next important step of KDD is the data itself. The input data must be prepared in a suitable format and structure that will fit each DM task perfectly. Raw data is not expected to be perfect without pre-processing. Since good DM models usually require good, structured data, it is important that the data quality is improved via thorough data cleansing. The data values must be correct and consistent as missing data is a major problem during DM processes, especially when occurring in large amounts; however, it is not all attributes (instances) with missing values can be removed from the sample [3].

Most of the existing datasets in the world, both governmental and non- governmental datasets contain attributes with some missing values (MVs). Missing values are the values of the attributes lost during the recording process. These values are lost due to various reasons, such as errors during manual data entry, incorrect measurement, and equipment failure. Clean data preparation process usually involves a pre-processing step where the data is first prepared and cleaned for use in knowledge extraction process. One of the easy ways of handling MVs is to delete the attributes that contain them from the data set. However, this is not a good method when dealing with data that contains a large amount of records with MVs as it will result to bias during the inference. In the presence of MVs, data analysis is a difficult task as it will expose the analyst to serious problems; in fact, if handled in a non-professional manner, it can lead to bias during data analysis and cause ambiguous conclusions; it can also limit the generalizability of the study outcome [4], [5].

In various fields, incomplete data is a major problem during data analysis. Missing values can be encountered for various reasons, such as failure to provide answers to some survey question, planned missing values, dropout, latent variables, intermittent missed measurements, and equipment failure. In fact, more than just one type of MVs can be encountered in many studies. Hence, MVs should be handled appropriately in the inference for the parameters of interest. Most of the techniques of handling MVs normally fails to account for the MVS-related uncertainty and this failure can lead to biased estimates and over-confident inferences. In DM processes, MVs are normally associated with three common types of problems which are a) loss of efficiency, b) data handling and analysis complexity, and c) unfairness due to differences between the complete data and the missing data [4], [6].

Many methods have been proposed for optimization processes, but the most efficient ones have been achieved by using the optimization algorithms. Reliance on optimization algorithms is due to their ease of implementation, cost-effectiveness, and ability to offer results near to real ones. Hence, the improvement of the digital filter is reliant on the performance of the optimizer. Optimization algorithms can be classified into different types and each one has a certain level of advantage over the other. Some of these algorithms have been inspired from the evolutionary theory, therefore, they are called “Evolutionary Algorithms” such as Genetic Algorithms (GA), Genetic Programming (GP),and Differential Evolution (DE). On the other hand, another type of optimization algorithm have been inspired from the living behaviour of some animals, insect, or even human, therefore, these algorithms are called “Swarm Intelligence (SI). There are tens of SI algorithms have been suggested in the literature, such as Particle Swarm Optimization (PSO), which have been inspired from the movement of the birds or fishes, Ant Colony Optimizer (ACO) which has been inspired from the unique style of movement of ants from the colonies to the food sources. Artificial Bees Colony (ABC), Cuckoo Search Algorithm (CSA), Grey Wolf Optimizer (GWO) and so many more are all examples of SI algorithms.

In this study, the major focus is on the Firefly optimization algorithm (FA) ; it is a nature-based meta-heuristic that can solve complex mathematical problems with close to ideal results based on the right choice of the parameter values with respect to the considered problem[7], [8]. FA algorithm as other optimization algorithms contains several controlling parameters for balancing between the global search and local search abilities. To be more specific, it contains four different parameters (Randomization Factor (α), Attractiveness (β), absorption coefficient (γ), and reduction factor (δ). The values of these parameters effect of the searching performance of the algorithms, therefore, they need to be tuned in order to balance between the search capabilities mentioned above. However, in this research, the default values for these parameters have been utilized. FA is used in this study as an imputation algorithm, meaning that it is used for filling the missing values in the dataset. The targeted case study in this thesis is the Pima Indian Diabetes Disease (PIDD) which is mainly about diabetes disease, because PIDD dataset is very popular in the biomedical case studies, and also it contains a lot of missing values.

1.2 PROBLEM STATEMENT

During decision-making processes, the challenge posed by missing data is more evident, especially in the on-line applications where data must be used as generated. Hence, decision making processes have recently been dependent on the use of computational intelligence techniques, such as neural networks and other pattern recognition techniques. However, decision making processes cannot continue in situations where some variables are not measured, and a major problem is that the standard computational intelligence techniques cannot effectively process input data with MVs and cannot perform regression or classification tasks[3], [4], [6].

Finding the solution to missing data problem is a tedious task in most applications and this is not usually considered in most decision-making tasks. Therefore, this demands for quick and perhaps inefficient techniques to handle missing data-related problems. This creates both computational and conceptual problems, raising the need for resources, such as methodologies and theoretical frameworks that can lead to an appearance of completeness [9], [10]. Most times, inefficient techniques are employed because there is limited time to find better techniques to handle missing data by the time they are observed, leading to the use of inefficient techniques such as case deletions. Unfortunately, some of the commonly used approaches cause more harm than good as they normally produce biased and unreliable results.

This study seeks to answer the following questions:

1- How to design an imputation algorithm based on Firefly Algorithm?

2- How to impute the missing values in PIDD dataset using Firefly Algorithm?

3- How to evaluate the performance of the proposed algorithm?

1.3 THE AIM AND OBJECTIVES

Designing an imputation algorithm based on Firefly Algorithm for imputation the missing values is the main contribution of this study. The proposed algorithm is used to enhance the classification performance based on PIDD dataset. The objectives of this research are:

1) To design an imputation algorithm based on Firefly Algorithm with three different classification models, KNN, SVM, and NBC.
2) To impute the missing values of PIDD dataset using the proposed algorithm.
3) To test and validate the proposed algorithm using different evaluation metrices.

1.4 THE SCOPE

This study is limited to the imputation of the missing values in the medical datasets. The type of the proposed algorithm is an optimization algorithm which is Firefly Algorithm. The targeted medical case study is the diabetes disease based on the popular dataset called (PIDD).

1.5 RESEARCH METHODOLOGY

The research methodology of this thesis is organized in five different phases. These phases are presented in Figure 1-1 below, which are:

Phase 1: Problem Understanding

In this phase, a review on the most recent and important studies is implemented in order to clarify the problem of missing values in the datasets in general and applying machine learning models on PIDD dataset in particular. Moreover, the three main classifiers used in this thesis are explained in detail, which K Nearest Neighbors (KNN), Support Vector Machine (SVM) and Naïve Bayesian Classifier (NBC).

Phase 2: Analyzing PIDD Dataset

The dataset used in this study is (PIDD). This dataset is studied and analyzed in terms of the types of the features, the ranges [Max, Min] of each feature, the histogram, the density, and Pearson Correlation Coefficient. Additionally, the missing values in PIDD are counted for each feature.

Phase 3: Designing the Imputation Algorithm

In this phase, the proposed imputation algorithm is designed and explained in detail. The algorithm includes several stages, one of these stages is the (FA) which is the main contribution of this study.

Phase 4: Implementation

In this phase, the proposed imputation algorithm including FA is implemented using MATLAB programming language. The version used in this study is 2018b, installed in Windows 10 operating system.

Phase 5: Evaluation

In order to validated and evaluate the proposed imputation algorithm, the evaluation phase is required. In this phase, several experiments are implemented with different scenarios based on different iterations and swarm sizes. Each experiment is executed with different classification model (i.e., KNN, SVM, or NBC). The results are recorded in terms of Accuracy, MSE, Sensitivity, and Specificity.

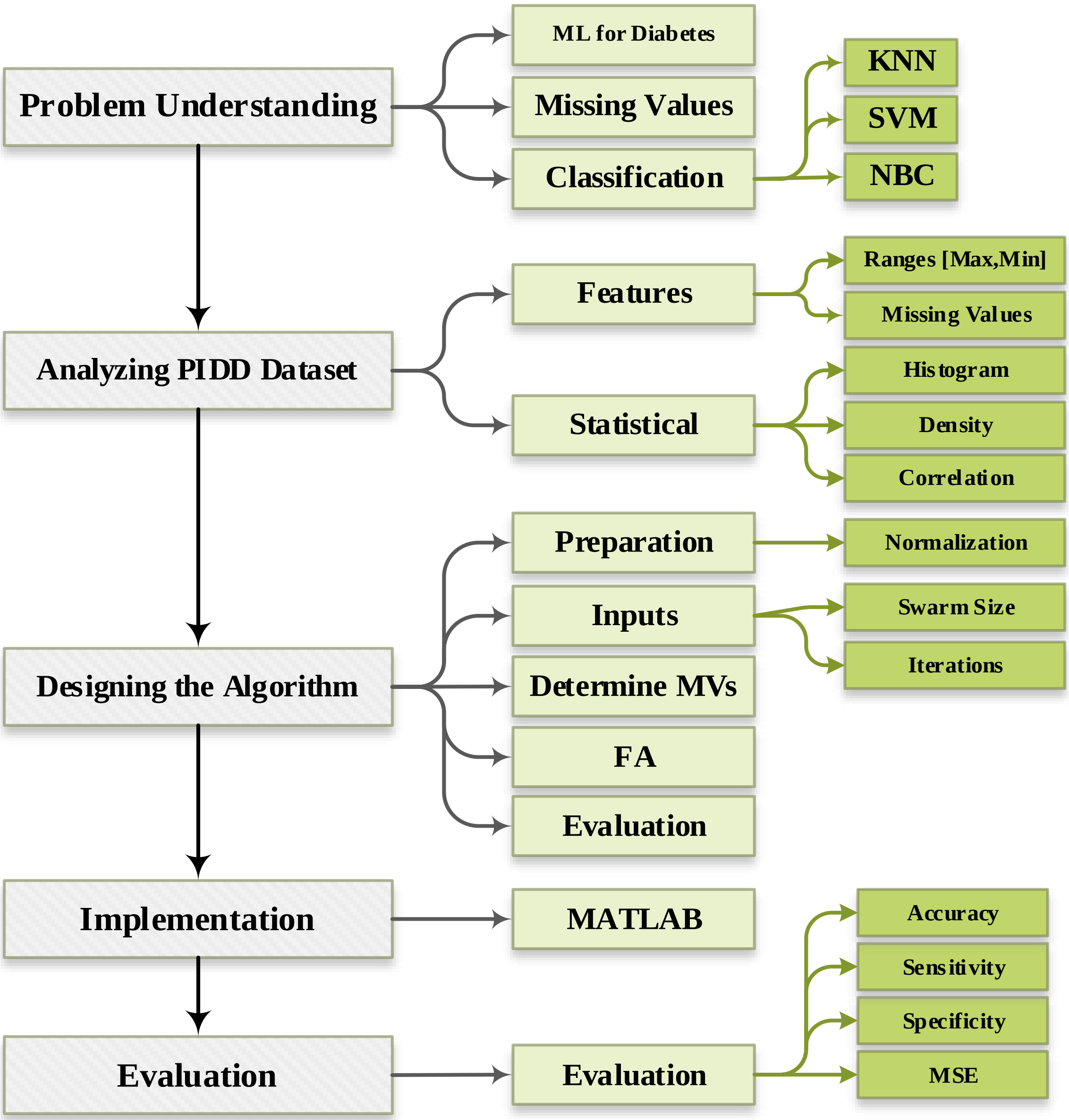


Figure 1-1 The main phases of the methodology

1.6 THESIS ORGANIZATION

Chapter two presents the background on missing values, and summarized the most important related works on the machine learning models for diabetes disease. In addition, KNN, SVM and NBC are presented. Chapter three explains proposed algorithm, including the main steps of FA. Chapter Four the PIDD dataset is presented and analyzed in this chapter. In addition, the proposed imputation algorithm is evaluated, and examined based on different scenarios. Finally, chapter five concludes the outcomes from the proposed algorithm, and presents the recommendations for the future works.

CHAPTER TWO

BACKGROUND AND RELATED WORKS

2.1 INTRODUCTION

In this chapter, a theoretical background on the main concepts and topics used in this study are covered and reviewed. In the first section, the missing values mechanisms are explained in details, followed by explaining the most common missing data imputation methods. Then, the machine learning classifiers used in the proposed algorithm and the evaluation process are explained in details. At the end of this chapter, the diabetes disease based machine learning models are reviewed.

2.2 MISSING DATA MECHANISMS

Missing data can best be handled by first considering how the data points become missing. There are three mechanisms of missing data; these are Missing Completely at Random, Missing at Random, and Non-Ignorable Case [4], [6], [11], [12].

2.2.1 Missing Completely at Random

This situation is encountered if the probability of the MV for variable X is not related to the value X itself or to any other variable in the complete data set. This implies that the missing data depends not on the variable of interest or on any other variable contained in the data set. In other words, the missing data values are simple random samples of all data values in the database. For instance, the missing data for age as a variable will be considered MCAR if the MV is not related to age or to the values of any other missing or observed variable in the database [13]. Another instance of MCAR is a situation where people that do not report their income are the same as those who do; here, income is considered as MCAR. In this case, there is no difference between cases with complete data and cases with incomplete data. In Table 2-1, the missingness of the missing value in x_4 is said to be MCAR if it does not depend on x_1, x_2 and x_3 and the variable x_4 itself.

Table 2-1 Example of MCAR

Sample	x_1	x_2	x_3	x_4	x_5
1	11.2	2.1	30.1	145.1	-6.4
2	N/A	1.9	27.2	N/A	-9.4
3	12.5	1.6	N/A	N/A	1.9
4	15.4	2.2	22.8	161.1	-2.4
5	17.2	1.8	19.9	N/A	N/A

2.2.2 Missing at Random

This case is encountered when the missing data probability on a particular variable X is dependent on the other variables but does not depend on X itself. For instance, if the probability of missing income is dependent on marital status but the probability of missing income in each category of marital status is not related to the value of income, then, income will be considered as MAR in this case. Although there are clear differences between cases with incomplete data and cases with complete data, the trend of missingness can be predicted from the other variables contained in the database rather than depending on the specific variable with the missing data. MAR implies that the variable has a missing value but depends on the other X variable in the data set, although not dependent on the Y variable of interest [11]. Hence, the probability of missing data on any variable depends not on its particular value. In Table 2-1, the missingness of the missing value in x_4 is considered to be MAR if it depends on $\boldsymbol{x_1}, \boldsymbol{x_2}, x_3$, and $\boldsymbol{x_5}$ but not on x_4.

2.2.3 Non-Ignorable Case

This case arises in situations where the missing data probability X is related to the value of X itself even when the other variables are controlled during the analysis[13]. This implies that the missing data are not random but dependent on the missing values. For instance, if households that earn high income are more unwilling to file their income even after other variables have been adjusted, then, the missing income probability is considered non-ignorable. The trend of data missingness in this case is non-random and cannot be predicted from the other variables in the database. This form of missing data is the most difficult to model and approximate[14]. In Table 2-1, the missingness of the missing value in x_4 is considered non-ignorable if the missing value in x_4 is dependent on variable $\boldsymbol{x_3}$ itself.

2.3 MISSING DATA IMPUTATION

In the currently existing statistical packages, there are various data imputation methods based on the data missing mechanism. These methods include the simple ones like list-wise or case-wise data deletion and methods that employ complex and sophisticated AI techniques.

Some of the mechanisms commonly used to handle missing data are discussed here, starting with the simple methods to the complicated and efficient methods.

2.3.1 List-Wise or Case-Wise Data Deletion

With this approach, an entire observation or case can be eliminated by most statistical procedures if the variables contain any missing data. Hence, this is called list-wise or case-wise data deletion; it is encountered when there are missing data in a record for one or more identified variables. It is a simple and easy way of handling data but remains the worst choice of handling missing data. This method omits any record that contain missing data for any variable.

This method can only be used to treat missing data if the missing data are small relative to the complete available data, else, this method can lead to biased estimate from the database when used in dataset with relatively larger missing data compared to the complete data. For instance, in Table 2-1, case-wise deletion method will omit records number 2, 3, and 5 and proceed with the analysis using the remaining data records.

2.3.2 Pairwise Data Deletion

This method works by relying on the available pairwise data to perform the required analysis, meaning that a record with missing data on one variable can only be used in calculations where that variable is not involved. The sample size in this manner is often larger compared to when using complete case analysis. As per [13], pairwise deletion results in biased estimates and is not recommended except if the data are MCAR. Considering Table 2-1, pairwise deletion method will only use record number 1 whenever there is analysis that do not involve x_4.

2.3.3 Mean Substitution

This method calculates the mean value of the variable from the available cases and use the calculated mean value as the imputed value for the missing cases. Similar to the pairwise deletion method, this method has a high chance of producing biased estimates; hence, it is not recommended. Considering Table 2.1, mean substitution method will substitute the values of all the missing values in variables x_4 by averaging the available values in that variable. Here, the value will be:

$$\frac{\sum_{i=1}^{N} x_i}{N} = \frac{145.1 + 161.1}{2} = 153.1$$

2.3.4 Hot Deck Imputation

Here, the most similar case to the case with a missing value is identified and substituted with the most similar case *x* value for the missing case. In Table 2-1, this method will substitute the missing value in the first record by finding the most similar record to the record number and substituting the most similar record's $\boldsymbol{x_4}$ value to record one x_3 variable.

Once the complete data case that is most similar to the record with incomplete data has been identified, the most similar complete case value for the missing variable will be substituted into the data matrix. Among the advantages of hot deck are its conceptual simplicity, proper maintenance of the variable's measurement level, and complete data set availability at the end of the imputation process. However, its major disadvantage is that it is difficult to define similarity as there are several ways of defining similarity in this context. Therefore, hot deck is not an out of the box approach of incomplete data handling. Sophisticated hot deck framework can identify more than one similar record before randomly selecting one from the available donor records for missing values imputation or using an average value when it is appropriate.

2.3.5 Regression Method

This method depends on the complete case data for a given variable to develop a regression equation; it treats missing variables as dependent variables while the other relevant variables in the database are considered predictors. For any record with a missing value, we approximate its value using the regression equation that was developed in terms of other variables. It should be noted that, in this method, a regression model is developed for each variable that has missing values while the other variables are considered dependents. The process is sequentially repeated for all the variables with MVs, meaning that for a variable x_j with MVs, a model is fitted based on the observations with the observed values for the other available variables. If the regression method is applied in Table 2-1, for the approximation of the MV in record 1, a regression equation will be developed in terms of variables x_1, x_2, x_3, and x_5 and can be formulated as:

$$x_4 = b_1 x_1 + b_2 x_2 + b_3 x_3 + b_5 x_5 + \varepsilon \tag{2.1}$$

This fitted model will contain the regression parameter estimates b_i and an error ε. Then, Eq 2.1 can be employed for the MV estimation by plugging the values of x_1, x_2, x_3, and x_5.

2.3.6 Expectation Maximization

This is an iterative approach that works in two steps; the first step is the expectation (E) step where the expected value of the complete data log likelihood is computed based on the complete data cases and the algorithm's best guess in terms of the sufficient statistical functions are for the MVs; it considers the specified model and the existing data points. Maximization (M) step is the next step wherein the expected values are substituted for the missing data derived from the E step before maximizing the likelihood function as if there are no missing data just to obtain new parameter estimates. Then, the new parameter estimates are resubstituted into the E step before performing a new M step. The procedure is repeated through the E and M steps until a negligible change of the parameter estimates is achieved iteration-wise (meaning convergence). Thus, the EM has the following main steps:

- Missing values replacement by the estimated values.
- Estimate parameters.
- Missing values re-estimation by assuming the correctness of the new parameter estimates.
- Parameters re-estimation and iterating until convergence.

The EM approach is advantageous because it has well-known statistical properties and performs better than the other methods as it assumes random existence of missing values in incomplete cases rather than completely missing at random. However, its major disadvantage is the addition of no uncertainty component to the estimated data, meaning that despite the reliability of EM-based parameter estimation, the standard errors and related test statistics are not reliable and this has led to the development of the raw maximum likelihood approach (full information maximum likelihood) and multiple imputation which are two novel likelihood-based approaches for handling missing data.

2.3.7 Raw Maximum Likelihood Method

Under the MAR assumption, the raw maximum likelihood, also called Full Information Maximum Likelihood (FIML) method constructs the best possible first & second order moment estimates using all the data points available in a database. This simply means that to meet the

MAR assumption, the maximum likelihood-based approaches ought to generate a vector of means, as well as a covariance matrix form all the variables in a database which are superior to the generated vector of means & covariance matrix by the commonly employed approaches of handling missing data.

The vector of means and covariance matrix are calculated using all available data in a manner that is superior to the other methods. The EM and raw maximum likelihood return similar parameter estimate values under an unrestricted mean & covariance structure. However, raw maximum likelihood is applicable in SEMs & regression models unlike EM. It also produces parameter estimates and standard errors by assuming that the fitted model is not false, thereby making the standard errors and parameter estimates model-dependent. This means that their values are dependent on the selected model by the operator. The advantage of the raw maximum likelihood is its ease of use and well-known statistical characteristics. It also permits direct computation of the appropriate test statistics and standard errors. However, its drawback is that it assumes joint multivariate normality of the used variables during the analysis and its analysis does not produce a raw data matrix.

The raw maximum likelihood approaches are also model-based, meaning that their implementation is done as a part of a fitted statistical model. The researcher may wish to introduce relevant variables thought to be capable of improving the parameter estimates accuracy but may not include such variables as the predictors or outcomes of the statistical model. Even though this can be done easily, it is not usually convenient especially when dealing with complex models.

Lastly, the raw maximum likelihood approach assumes the all incomplete data cells are MAR. It can perform better than the list-wise and pair-wise deletion approaches even when faced with the non-ignorable data.

2.3.8 Multiple Imputation

This approach is similar to the raw maximum likelihood approach just that it creates 5 to 10 data sets in the raw data to fill the missing data[15]. Then, the data from the imputed data set are pooled before estimating the parameters. Multiple imputation boasts the advantages of the EM and raw maximum likelihood approaches as it can produce the dataset for the analysis via hot deck imputation. Just like EM, multiple imputation generates a maximum likelihood from the vector of means and the covariance matrix. The multiple imputation method goes a step further by introducing a degree of statistical uncertainty into the model. With this uncertainty, it can mimic the natural variability among the cases existing in a complete database[14]–[16].

The multiple imputation method performs actual data values imputation to fill in the missing data points in the data matrix. The multiple imputation method mainly differs from hot deck imputation from the procedural perspective by requiring the data analyst to generate 5 to 10 databases with imputed values. The, each database is analysed by the data analyst before collecting the analysis results and summarizing them into a summary set of findings. For example, if a researcher decides to execute a multiple regression analysis on a database that has incomplete data, first, the researcher will be required to run multiple imputation to generate 10 imputed databases before running the multiple regression analysis on the 10 generated databases. Then, the results from the 10 regression analyses will be combined to get the result summary.

The advantages of multiple imputation include its ease of understanding and resistance against non-normality of the variables used in the analysis. Its analysis also produces complete raw data matrices just like hot deck imputation. It performs better than list-wise, pairwise, and mean substitution methods of missing data handling. However, its low points include the prolonged time imputing of 5 to 10 databases, time for separate testing of each database, and time for getting the summary of the model results[17], [18]. The Multiple Imputation (MI) method is the commonly used approach for general purpose missing data handling in multivariate analysis. As per Little and Rubin, the basic idea of The MI is as follows:

1. MVs imputation using a suitable model that includes random variation.
2. Repeat this for M times (say 3-5 times) to produce M complete datasets.
3. Perform the desired analysis on each data set based on the standard complete data methods.
4. Average the parameter estimates values across the M samples to get a single point estimate.
5. Estimate the standard errors by (a) taking the average of the squared standard errors of the M estimates, (b) estimate the variance of the M parameter estimates for each sample, (c) use a simple formula to combine the two quantities.

The raw maximum likelihood and MI methods currently seems to be the preferred techniques of handling missing data due to lack of obvious advantages of the other methods, such as regression, hot deck imputation, and expectation maximization over MI and raw maximum likelihood. MI methods are more suitable for a range of linear & nonlinear models; even when faced with non-ignorable missing data, raw maximum likelihood has been found to perform better than pair-wise deletion & complete case analysis approaches.

2.4 MACHINE LEARNING CLASSIFICATION MODELS

Machine Learning (ML) models can be classified based on the learning technique and the existence of the targeting labels. Therefore, there are two main types of machine learning models:

1- Supervised Learning Models.

This type of machine learning models requires the targeting – or class – labels in the training dataset. If the class labels are categorical, the case study is called “Classification”, otherwise if the class labels are numerical, the case study is called “Regression”. There are tens of classification models used in the literature. In this study, three common classifiers are used for calculating the fitness function, and for evaluating the results. These models are: K-nearest neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayes Classifier (NBC)[19], [20].

2- Unsupervised Learning Models.

In this type of machine learning models, the targeting – or class – labels are unknown. The case studies in this type are called “Clustering”. K-Means is one of the most common clustering models used in the literature.

In the next subsections, the classification models used in this thesis are explained in detail.

2.4.1 K-Nearest Neighbors

This is one of the most traditional algorithms for performing both classification and regression tasks. Considering the scope of this study, the focus is only on the classification potential of KNN. As a powerful algorithm, KNN is used to solve real world problems in many fields; for instance, it can be used for visual pattern recognition to scan and detect hidden packages in the bottom of a shopping cart at check-out; it can also be used to predict the incidence of some diseases by collecting the medical data of patients. With a proper application of machine learning algorithms, there are many benefits associated with them.

As earlier described, the output values in a dataset are classes that represent the target of the case study. For every new-added sample, the KNN algorithm calculates the distance between this sample and all other samples in the dataset through the feature space to find its k-nearest neighbours. Then, this sample will be assigned to the class with the most samples among these neighbours[21]–[23].

As KNN relies on calculation of the distances between new sample and each of the training samples to decide the final classification output, the major problem becomes how to calculate the distance between two samples. To address this issue, the simple solution is to imagine that for every new sample with N features, the values of features are the coordinates in N-dimensional space and are used to calculate the distance according to distance formula. Consider Figure 2-1, the new sample (the blue circle) will be classified as positive if small squares are taken as positive samples and as negative if small circles are taken as negative samples. This is because, among its 5 nearest neighbours, the number of positive samples is larger than the number of negative samples based on the voting mechanism. Many functions exist for calculating the distance; for instance, the Euclidean distance function is used widely and fits our dataset. It is calculated as:

$$Dist(A,B) = \sqrt{\frac{\sum_{i=1}^{m}(x_i - y_i)^2}{m}} \tag{2.2}$$

where $A = (x_1, x_2, \dots, x_m)$, $B = (y_1, y_2, \dots, y_m)$ and m is the dimensionality of the space. However, there is a need to try all the functions to determine their levels of performance.

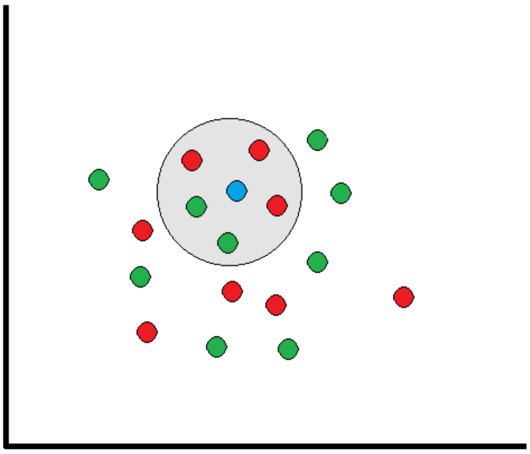


Figure 2-1 A graphical illustration for KNN Classifier

Factor K greatly impacts the performance of KNN, therefore, the choice of K really affects the results. At first, the choice of the parameter k is a crucial and somehow the most important step in this algorithm; the choice relies mainly on the kind of data available. When planning to implement KNN with different k values on a two-class dataset, different boundaries will have to be set to separate the two classes; the boundary will become more and more gentle if K is increased gradually.

Secondly, many methods are available for finding the optimal k; one of such methods is K-Fold Cross Validation as will be discussed later. The best k value can be found by plotting the cross-validation accuracy with different k. As depicted in Figure 2-2, the value of k was calculated using 15 repeats of 10-fold cross-validation. Looking at the plot, the k value with the highest accuracy lies between 15 and 20.

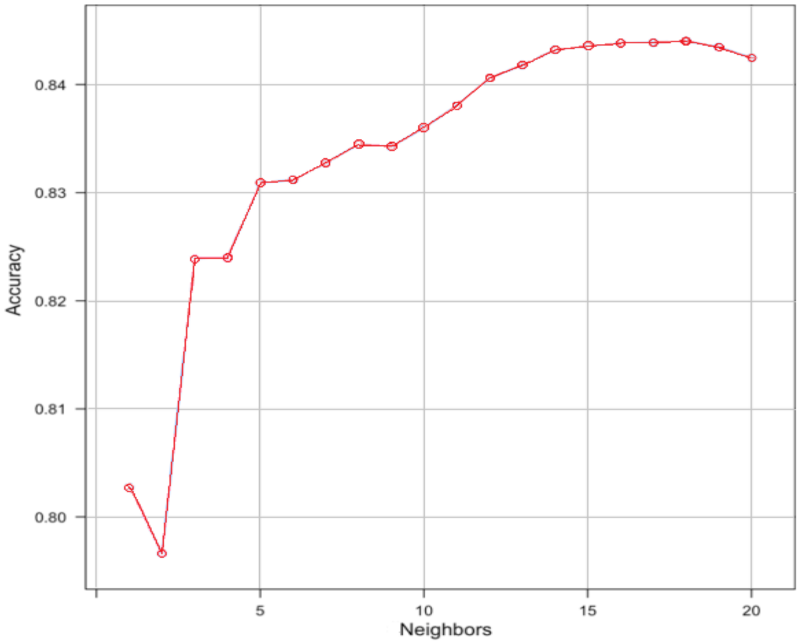


Figure 2-2 The accuracy obtained by cross validation with different values of K

2.4.2 Support Vector Machine

SVM is another popular machine learning algorithm for both classification and regression tasks and has found application in many fields such as in solving various real-world problems, text categorization and image identification, and handwriting recognition[24].

For classification tasks, SVM is based on the principle that the features of the samples are considered coordinates that should be mapped into N-dimensional space (N is decided by the number of features). Based on the data and the kernel function in the algorithm, SVM will train a model and classify samples into different classes with the help of a margin and its boundaries. Then, the classification of a new sample will rely on where it falls in. Just like KNN, the optimal SVM parameters (margin and boundaries) also needs to be found. There are two forms of classifications in SVM, these are linear classification and non-linear classification [2], [24]–[26].

A. Linear classification

This form of classification indicates that two classes can be divided by a margin hyperplane. It involves two situations which are linear separable and non-linear separable.

❖ Linear separable: A situation where positive samples and negative samples can be separated completely by a straight line or a hyperplane is termed linear separation (See Figure 2-3). Consider Figure 2-4; here, multiple separating lines or hyperplanes can be drawn between the samples. All the straight lines in this figure are meaningful as they can clearly separate the classes.

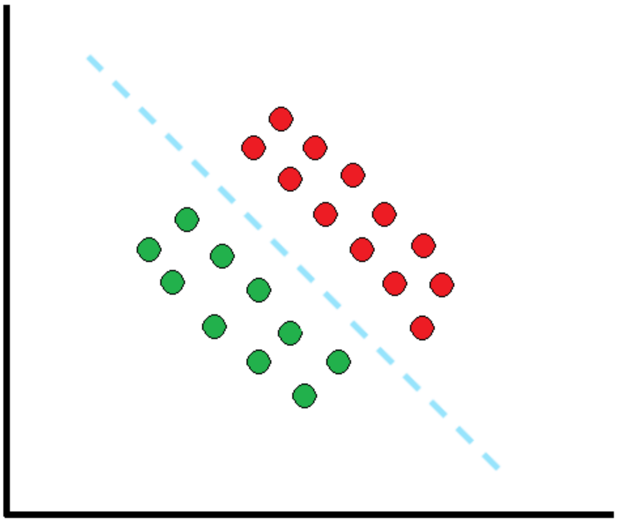


Figure 2-3 Single Line Separation

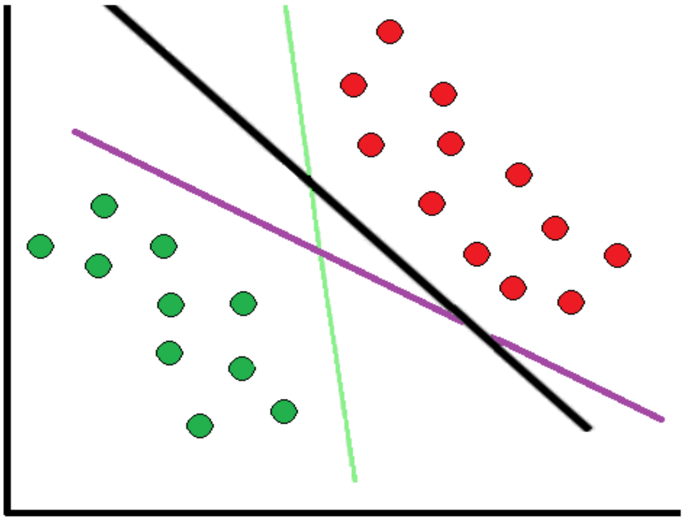


Figure 2-4 Multiple separation lines

- ❖ <u>Non-linear separable:</u> Another situation is non-linear separation where some points can be misclassified due to data complexity.

B. Non-linear classification

In this case, the separation boundaries produced by the previous process in linear classification do not always work because of the complexity in real life data (Figure 2-5). As earlier stated, SVM maps sample features into a N-dimensional space and use the number of features as the dimension of space. For the non-linear separable data points, the strategy of the kernel function is to increase the space dimension to reduce the problem complexity:

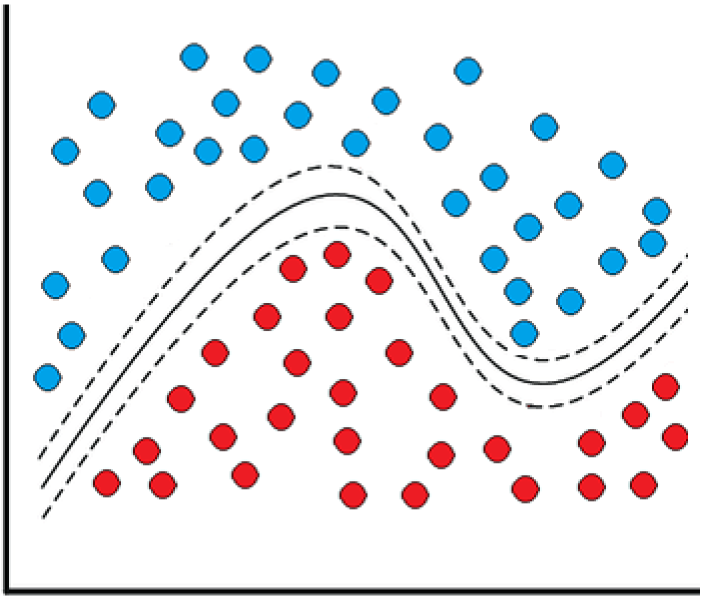


Figure 2-5 Non-Linearly Separable data

Linear
$$K(X_i, X_j) = X_i^T X_j \tag{2.3}$$

Sigmoid
$$K(X_i, X_j) = \tanh\,(\gamma X_i^T X_j + r) \tag{2.4}$$

Radial Basis Function (RBF)
$$K(X_i, X_j) = \exp\left(-\gamma\left|\left|X_i - X_j\right|\right|^2\right), \gamma > 0 \tag{2.5}$$

Polynomial
$$K(X_i, X_j) = \left(\gamma X_i^T X_j + r\right)^d \quad , \ \gamma > 0 \tag{2.6}$$

Linear separation employs the first kernel function one while non-linear separation uses the last three kernel functions. Furthermore, Radial Basis Function (RBF) has been proven as the most effective kernel among these functions and has been applied most frequently in practice.

Generally, SVM relies on two main control parameters as follows:

- **<u>Parameter γ:</u>** In the RBF (equation 2.5), parameter γ controls the separation boundaries; it decides the distance between the samples and the boundaries. Low γ results in long distance and high γ can lead to short distance. If the γ is too high, the kernel function will shrink and will not recapitulate the data totally, thereby raising the risk of underfitting. On the other hand, if the γ is too low, the kernel function will extend and more data points will be included in the margin, thereby working like a linear separation, and will be prone to overfitting.

- **Parameter C:** This is another important factor in kernel functions; it is the number of misclassified data points and is considered the penalty function for an error. When parameter C is increasing, it indicates that the margin is getting bigger; so, more points will be penalized. On the other hand, a decreasing value of parameter C implies that the margin is getting smaller and fewer points will be penalized. The aim is not to penalize too many points so that more data points will be available to train and get an accurate model; however, it is aimed that there will be enough margin to generalize the dataset as much as possible. So, an ideal point will be to have a trade-off between having more data points to train and generalizing the dataset as much as possible.

Both parameters λ and C are important in SVM, hence, they need to be optimized for the SVM to perform optimally.

2.4.3 Naïve Bayesian Classifier (NBC)

The NBA captures the probabilistic relationship between the attribute set, citation's frequency and citation's position with the class variable; it relies on the Bayes theorem of probability to predict the class of unknown dataset with an assumption, where feature's properties independently contribute to the probability. The Bayes theorem was introduced for solving mainly classification tasks. This section discussed the Bayes theorem and its implementation in NBC.

The Bayes Theorem is a statistical principle that governs the combination of prior knowledge of a class with new information gathered from the data; this process employs the formula in equation 2.7. Consider X and Y as a pair of random variables; the Bayes theorem will facilitate the expression of the posterior probability in terms of the prior probability P(Y), the conditional class probability $P(X|Y)$, and the evidence P(X) [1], [2].

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2.7}$$

Where the prior probability of the target Y is represented by $P(Y)$, and prior probability of the samples X is represented by $P(X)$. While the posterior probability of Y is represented by $P(Y|X)$, and the posterior probability of X is represented by $P(X|Y)$.

The Bayes theorem is also applicable in classification and prediction problems; this study employed the Bayes theorem for classification. Let X represent the class attribute set and Y as the class variable. Then, $P(Y)$ represents the prior probability estimable from the training set using the data fraction that belong to each class; then, the class-conditional probability $P(X|Y)$ applied will be specified, followed by learning the posterior probabilities $P(Y|X)$ for each X and Y combination based on gathered information from the training data. The classification of a test record can be done by finding the class that maximizes the posterior probability.

The NBC is a statistical principle that governs the combination of prior class knowledge with new information gathered from data. The Naive Bayes classifier is a simple classifier that can perform better than most of the existing complex classification methods. It achieves high speed and accuracy when applied to a large database. In this section, the application of the NBC will be discussed based on our training citation dataset. The NBC assumes the attributes are conditionally independent for the class-conditional probability and to classify a test record, it computes the posterior probability for each class, Y. The general concept of the process is as depicted in Figure 2-6.

$$P(Y|X) = \frac{P(Y)\prod_{i=1}^{D} P(X_i|Y)}{P(X)} \tag{2.8}$$

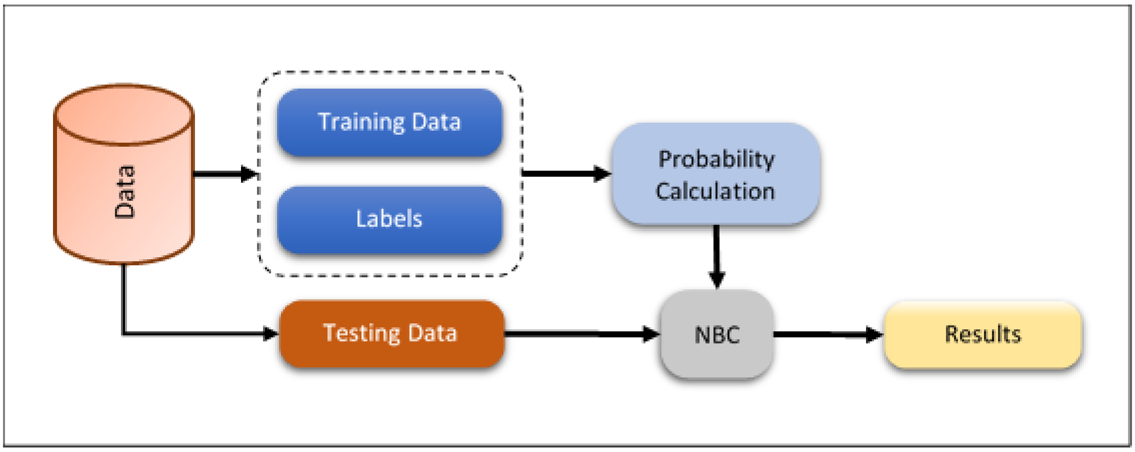


Figure 2-6 The process of NBC

The conditional probabilities for continuous attributes can be estimated by assuming a Gaussian probability distribution for the continuous variable and using the training data to estimate the parameters of the distribution as represented in the formula below. Two parameters characterized this distribution; they are the mean and the variance.

2.5 DIABETES DISEASE BASED MACHINE LEARNING

Diabetes disease, also described as diabetes mellitus, is one of the common health problems. It is a group of metabolic disorders that manifests in hyperglycaemia either due to insulin intolerance, insufficient insulin production, or by both aetiologies. Early detection has been approved as the best way of diagnosing DM. As per the WHO report of Nov 14, 2016, there are about 422 million people living with diabetes globally while about 1.6 million people have died due to DM. Hence, the severity of diabetes can be easily predicted based on this report.

The onset of DM is associated with various levels of organ damages, such as damage to the eyes, kidney, nerves, and heart. As per "*Williams's Textbook of Endocrinology*", in 2013, there were more than 382 million diabetic patients globally and many of them had died due to DM-related issues in both poor and rich countries. According to the "Centres for Disease

Control and Prevention (CDCP)", the US witnessed a 23% increase in type II DM cases over a period of 9 years (2001 to 2009). This means that DM has become a global health problem in terms of its prevalence and prevention.

The two major types of DM are Type I and Type II diabetes. Type I diabetes is also called insulin-dependent diabetes as the human body cannot produce enough insulin upon its onset. It accounts for about 10 % of all DM cases. Regarding Type II diabetes, the Canadian Diabetes Association predicted an increase from 2.5 million cases to 3.7 million between 2010 and 2020. Diabetes is, therefore, a health condition that demands early detection and prevention to reduce its associated life-threatening consequences.

Considering the global impact of DM, several methods of its prediction have been developed. For instance, Abdar et al. (2017) developed a machine learning-based approach for both prediction and classification of diabetes. Furthermore, Aljumah et al. (2013) presented data mining techniques for useful information recovery from large health datasets. Data mining has become a useful tool in diabetes studies owing to the advancements in Information Technology; it has led to better health care delivery and improved decision-making support for better disease supervision.

Bashir et al. (2016) suggested that no single technique currently exist which offers the highest disease prediction accuracy since the good performance of a given classifier can be better in one disease dataset but perform badly in another. Hence, this study presents the hybridization of different classifiers for DM prediction and classification. The aim of this hybridization approach is to overcome the issues associated with the component classifiers.

The study by Komi et al. (2017) discussed different classification frameworks based on different parameters, such as skin thickness, insulin, glucose, blood pressure, BMI, age, and diabetes pedigree while effectively excluding pregnancy as a parameter during DM prediction.

This study only depended on small sample data to predict diabetes using 5 different algorithms (GMM, ANN, SVM, EM, and LR). The outcome of the study showed that ANN performed best DM prediction. Kavakiotis et al. (2017) used ML algorithms for the prediction of different medical data sets, including DD dataset and found them reliable. The study evaluated the capability of SVM, LR, and NB for medical datasets prediction based on 10-fold cross-validation. The performance and accuracy of the employed algorithms were compared and SVM was found to offer the best accuracy compared to the other algorithms.

A study by Nilashi et al. (2017) employed CART for fuzzy rule generation. The study also used PCA and EM as clustering algorithms for data pre-processing before rules application. Different medical datasets (MD), such as heart, breast cancer, and diabetes were considered for the development of a decision support system for various diseases, including diabetes. The results showed that CART provided better and effective disease prediction with preprocessed data compared to the non-processed data.

As per Mercaldo et al. (2017), feature selection is an important step towards increased accuracy. The study relied on various algorithms for the prediction task while different feature selection algorithms were employed for the feature selection task.

Kandhasamy and Balamurali (2015) used different datasets, including DD to construct models that can be applied to different medical datasets. However, the proposed classification algorithm was not validated via cross-validation. Among the algorithms used in the study (ANN, KNN, NB, J48, ZeroR, etc), NB achieved the best accuracy on DD while KNN and ANN performed well on the other datasets.

Perveen et al. (2016) focused on the use of CPCSSN “Canadian primary care sentinel surveillance Network” dataset and 3 ML methods for early prediction of DD. The prediction was performed using Bagging, Adaboost, and J48; the performance of the employed frameworks was compared and Adaboost method was adjudged the most effective and accurate method in Weka data mining tools compared to the other methods.

The study by Kamadi et al. (2016) focused on the identification of classification problems, focusing more on data reduction as a major problem in classification tasks due to its influence on prediction accuracy. The study noted the data needs to be reduced to get better and accurate performances. The study used PCA for data pre-processing while the modified DT and Fuzzy were employed for the prediction task. From the results, the system performed better with the reduced dataset, thereby highlighting the need for data reduction.

The performance of ML techniques on pre-processed and non-processed datasets has been compared by Pradeep and Naveen (2016) in terms of their accuracy. The study indicated the impact of data preprocessing during disease prediction on the process accuracy. The results showed that DT performed the best DD prediction in terms of accuracy on the non-processed dataset while RF and SVM performed better on the pre-processed dataset.

Santhanam and Padmavathi (2015) used GA and K-means to improve the process of data dimension reduction while SVM was used for the prediction task. The study relied on 10 cross-validation approach for the evaluation purpose and from the results, the system performed better on the reduced data set compared to the large dataset.

Different data mining methods have been used by Meng et al. (2013) for DD prediction on real-world data sets; the study relied on structured questionnaire for data collection while machine learning and statistical tools (i.e., “WEKA” and “SPSS”) were used for the data analysis & prediction phases. The study compared ANN, LR, and j48 and found j48 ML technique as the best in terms of accuracy and efficiency.

Aljumah et al. (2013) used Oracle Data miner and Oracle Database 10 g for data analysis and storage. In this study, the identification of the target variable was based on their percentage. The study also considered the patients’ treatment stage as the patients were grouped into old and young categories based on their age before predicting their treatment. The outcome of the study showed high predictive percentages for both the young and old control groups as predicted using SVM.

CHAPTER THREE

PROPOSED ALGORITHM

3.1 INTRODUCTION

As stated in the first chapter, the main contribution of this study is to design an imputation algorithm based on (FA). In this chapter, the proposed algorithm is explained in detail. First, an overview on the firefly algorithm is given, then the mathematical model of the proposed algorithm including the stages is presented. The data normalization in data preprocessing stage is explained with an example. The Firefly algorithm for estimating the missing values is explained in details in a specific subsection. The last section in this chapter describes and analyzes the PIDD dataset used in this study.

3.2 OVERVIEW ON FIREFLY ALGORITHM

The ability of nature-inspired metaheuristics to provide solutions to modern optimization problems has attracted much research interest, especially their performance on NP-hard optimization problems, such as the traveling salesman problem. One of the nature-inspired metaheuristics commonly used in solving difficult optimizations tasks is the Particle Swarm Optimization (PSO) which was first developed in 1995 by Kennedy & Eberhard [40]. The PSO was inspired by the swarm behaviour of natural species, such as the flocking of birds and the schooling of fish. The PSO has found application in different optimization field where it has performed excellently. The Firefly Algorithm (FA) is another metaheuristic that has demonstrated good performance in may applications; it was developed by Yang, (2009). In these multiagent frameworks, the search mechanisms are governed by efficient local search, randomization, and optimal solution selection. However, the randomization normally uses uniform or Gaussian distribution.

Fireflies flashes distinctive light patterns that can best be appreciated in the temperature during the summer. Numerous species of firefly exist, with most of them producing distinctive short and rhythmic lights. A bioluminescence process is responsible for the flashing of light in fireflies even though the actual functions of such signalling systems are yet to be understood. It is believed that the flashing serves two basic functions which are to attract potential mating partners and to attract potential prey. Another role of the flashing could be as a protective warning mechanism. The attraction between both sexes is determined by the rhythmic flash, the flashing rate, and the timing of the flashing. Naturally, the female fireflies of a species respond to the flashing of the males from the same species; however, the female fireflies of some species, such as Photuris, can copy the light pattern of another species just to attract the male species and eat them.

The light flashing pattern can be formulated in a manner that will associate it with the intended objective function to be optimized; hence, it can be formulated into a new optimization algorithm. Hence, the FA will be discussed in terms of its basic formulation and implementation.

To develop the firefly-inspired algorithms, it is important to idealize some of the flashing patterns of the fireflies. The following idealized rules are used to describe the FA [42]–[44]:

1) There are no sex differences between all fireflies (they are unisex); hence, they can be easily attracted to each other irrespective of the sex.
2) The level of attraction of any firefly is determined by the intensity of the light it emits; this means that fireflies that it lights of lower intensity will be attracted to those that emit lights of higher intensity as attractiveness is a function of the brightness of the emitted light and both are indirectly related to distance. In the absence of a brighter firefly within a surrounding, the swarm will be moving randomly.
3) The landscape of the objective function determines the brightness of any firefly. The brightness, for any maximization problem, is directly proportional to the value of the objective function. The description of other forms of brightness is similar to that of the fitness function in the bacterial foraging or genetic algorithms

Two important issues are highlighted in the FA; they are the changes in light intensity and the attractiveness formulation. We can simply assume that the brightness of any firefly determines its level of attractiveness and this is related to the encoded objective function.

For the maximum optimization problems, the brightness I of a firefly at a given location x can be denoted as I(x) ∝ f(x); however, the attractiveness β is relative and should be determined by the other fireflies. Therefore, it is dependent on the distance r_{ij} between firefly i and firefly j. Furthermore, the intensity of light reduces as the distance from source increases due to light absorption in the medium; hence, attractiveness should be allowed to vary based on the extent of light absorption. Simply, changes in light intensity $I(r)$ follows the inverse square law $I(r) = \frac{I_s}{r^2}$ where I_s represents the light intensity at source. However, light intensity I varies with the distance r for any medium with a constant light absorption coefficient γ; that is:

$$I = I_0 e^{-yr^2} \tag{3.1}$$

where I_0 represents the initial light intensity.

The level of attractiveness of a firefly is a function of the observed light intensity by the neighboring fireflies; so, the attractiveness β of a firefly can be defined as:

$$\beta = \beta_0 e^{-y r^2} \tag{3.2}$$

With the availability of more reports on the FA, it may be necessary to ask: Why is FA so efficient? The reason is not far-fetched. The analysis of the main features of the standard FA will arrive at the following points[41], [45]:

- In the FA, the population can be automatically partitioned into subgroups as a result of the stronger local attraction compared to long-distance attraction. Consequently, FA can efficiently handle highly nonlinear multi-modal optimization tasks.
- FA has no record of individual best and has no explicit global best. With this, it cannot be tempted to premature convergence. Furthermore, FA relies not on velocities and cannot be prone to velocity-related issues like PSO.
- FA can control its modality to fit into a problem domain by controlling its scaling parameter, such as γ. Simply, FA is a generalization of SA, PSO, & DE.

FA can also solve different problems; for multi-objective problem, FA can convert them into single-objective problems by linearly combining different objectives as a weighted sum. For PSO, a penalty factor h will be introduced for the same purpose. FA relies on a population of solutions to find multiple optimal solutions easily (say in one run) compared to the PSO where each agent is a potential solution to the problem. Lastly, FA converges within an acceptable time frame.

3.3 MATHEMATICAL MODEL OF THE PROPOSED ALGORITHM

In this section, the proposed imputation algorithm based on FA is presented. The section is divided into two sub-sections, In the first subsection, the proposed imputation algorithm in general, while the second subsection explains the FA algorithm used in this study in details.

3.3.1 The Proposed Imputation Algorithm

In the process of imputing or estimating the missing values in the targeted case study, the imputation algorithm based is designed for this purpose. The proposed algorithm consists of several stages as follows (See Figure 3-1):

Stage 1: Dataset Preparation

The first stage of the proposed algorithm represents the preparation of the dataset. It means reading and preprocessing the dataset using three simple steps, as follows:

- Step 1. Read the Dataset
- Step 2. Convert the Dataset from it is original format (i.e., excel format '.xlsx') into a "comma separated value '.csv', which can be easily read by almost any modern programming language.
- Step 3. Normalize the dataset in a fixed range [0,1] using MinMax method. Subsection 3.3.2 explains this method in details.

Stage 2: The Inputs

In this stage, the algorithmic parameters such as the size of the swarm, the maximum number of iterations, and other FA controlling variables are entered.

Stage 3: Determine the Positions of the Missing Values

In order to fill the missing values, the positions of these values should be determined. In addition, the number of these missing values is determined as well. Based on the previous two information, the solution representation for each firefly in the swarm is structured.

Stage 4: FA Implementation

In this stage, the firefly algorithm is executed in order to search for the best values, which replace the missing values in the dataset. The main steps of FA are given in the subsection (3.3.3).

Stage 5: Evaluation

In this stage, the best solution obtained using FA is evaluated in terms of classification accuracy, error rate, sensitivity and specificity. This stage is explained in details in section 3.4, while the obtained results are presented in the next chapter.

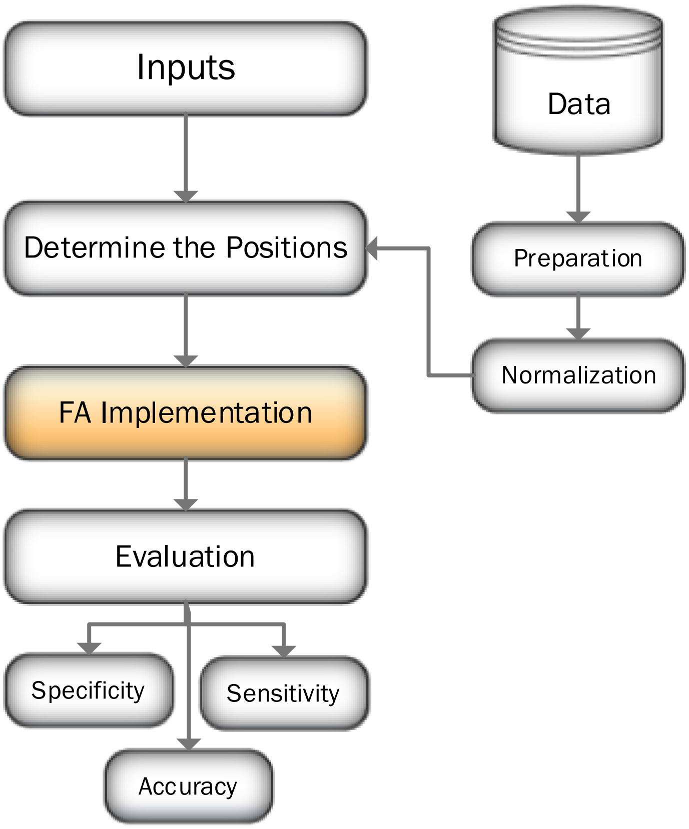


Figure 3-1 The block diagram of the proposed algorithm

3.3.2 Dataset Preparation (Data Normalization)

In most case studies, the features in each dataset have distinct ranges, which may reduce the efficiency of the classification performance. Therefore, there is a need for unifying the ranges of the features in one upper and lower boundary. This process is called “Data Normalization”. One of the most common method is *MinMax* Normalization, which converts the ranges of each attributes into a specific range, when the maximum value is converted to 1, while the minimum value is converted to 0. The rest values are converted in between 0 and 1. *MinMax* method is implemented based on the following equation:

$$N_v = \frac{X_v - Min}{Max - Min} \tag{3.3}$$

Where N_v represents the normalized value, while X_v represents the original value. Min and Max denote the maximum and minimum values of a specific feature respectively.

In order to have a clear understanding, a simple example is given below:

Ex: Consider the following data for a single feature { 7, 13, 5, 20}. The maximum value is 20, and the minimum is 5. The normalized values are:

- For the value (7) :

$$\frac{7-5}{20-5} = 0.133$$

- Fore the value (13)

$$\frac{13-5}{20-5} = 0.533$$

- For the value (5)

$$\frac{5-5}{20-5} = 0$$

- Fore the value (20)

$$\frac{20-5}{20-5} = 1$$

It can be seen from the above mentioned example, that the normalized values for 5 and 20, are 0 and 1 which are the minimum and maximum values. In addition, the normalized values for 7 and 13 are 0.133 and 0.533, which in range of [0,1]. The method of *MinMax* is implemented for all features in the dataset.

3.3.3 FA for missing values estimation

FA algorithm is organized in a way that it requires the following steps to be set up properly. Not all these steps are a necessary requirement but helps in implementing the algorithm more efficiently. Figure 3-2 illustrates the flowchart of the main steps of FA.

1. Set initial parameters in the parameter vector $[S.S\ MaxItr\ \ \alpha\ \beta\ \gamma]$. The search space is limited by Upper bound (UB) and Lower bound (LB) values. The values of UB and LB are initialized based on the case study, while the Swarm Size (S.S) and the maximum number of iterations are initialized based on different scenarios.

2. Initialization:

Generate a random position for each firefly in the swarm, via the uniform distribution method as follows:

$$F_i = (UB - LB) \times Rand(0{,}1) + LB \qquad (3.3)$$

Where $Rand$ is a randomization method, which generates a random value in range [0,1].

3. Fitness Function:

In order to evaluate each generated – or estimated – solution via the classification accuracy (A). The accuracy is generated via K-Fold Cross Validation where K is equal to 5. Three different classification models are used in this study, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Naïve Bayesian Classifier (NBC).

The intensity of each firefly in the swarm is calculated via the following equation:

$$F_i.I = \frac{1}{(1 - F_i.A)^2 + 1} \qquad (3.4)$$

4. Position Updating

Move the Firefly F_i towards another Firefly F_j with higher intensity $(I(F_i) < I(F_j)$ via the following equation:

$$F_i.Position = F_i.Position + \beta\big(F_j.Position - F_i.Position\big) + a \times (Rand(0,1) - 0.5) \tag{3.5}$$

Where

$$\beta = \beta_0 e^{-yr^2} \tag{3.6}$$

And r represents the distance between these two fireflies, calculated as follows:

$$r_{ij} = \big\|F_i.Position - F_j.Position\big\| = \sqrt{\sum_{k=1}^{D} \big(F_i.Position_k - F_j.Position_k\big)^2} \tag{3.7}$$

The parameter a in equation 3.5 represents the step size, which is linearly decreased via the following equation:

$$a = a \times \delta \tag{3.8}$$

Where δ is in range [0.90, 0.98].

5. Check the Boundaries Limits

Check whether the values obtained in the new position of the firefly is within the search space or not, as follows:

$$F_i.Position = \begin{cases} LB & If\ F_i.Position < LB \\ UB & If\ F_i.Position > UB \end{cases} \tag{3.9}$$

Then, F_i s evaluated using the fitness function explained in Step 3.

6. Sorting and Ranking

After updating the positions of all fireflies, the swarm is sorted and ranked based on the fitness value. Obtain F_{Best} value from the Swarm (which will always be the topmost value after sorting). Compare every value of the F with itself (comparing F_i and F_j).

7. Stop Condition

The first and second steps are executed only one time, while the rest steps (3-6) are iterated for t times. Meaning that the algorithm checks t if it is still less than $MaxItr$ – which has been identified in the first step – then go to step 4. Otherwise, exit the loop and return the last F_{Best}.

The pseudo-code of the proposed IFA is summarized in the algorithm below.

Imputation Firefly Algorithm (IFA)

```
Set Initial values for all parameters ($N, MaxItr, a, \beta, \gamma, \delta$ )
Determine the positions of Missing Values (MVs)
Determine the classifier (1. KNN, 2. SVM, 3. NBC)
Initialize all fireflies in the swarm via eq. 3.3
While ($Itr \leq MaxItr$)
        For i = 1 To N
            For j = 1 To N
                IF ( $Intensity(F_i) < Intensity(F_j)$)
                    Update the position of $F_i$ vie eq. 3.5
                    Check the boundaries via eq. 3.9
                    Full the dataset and update the Fitness value of $F_i$
                End IF
            Next j
        For i
        Rank the swarm and determine $F_{Best}$
Loop ( $Itr + 1$)
Return $F_{Best}$
```

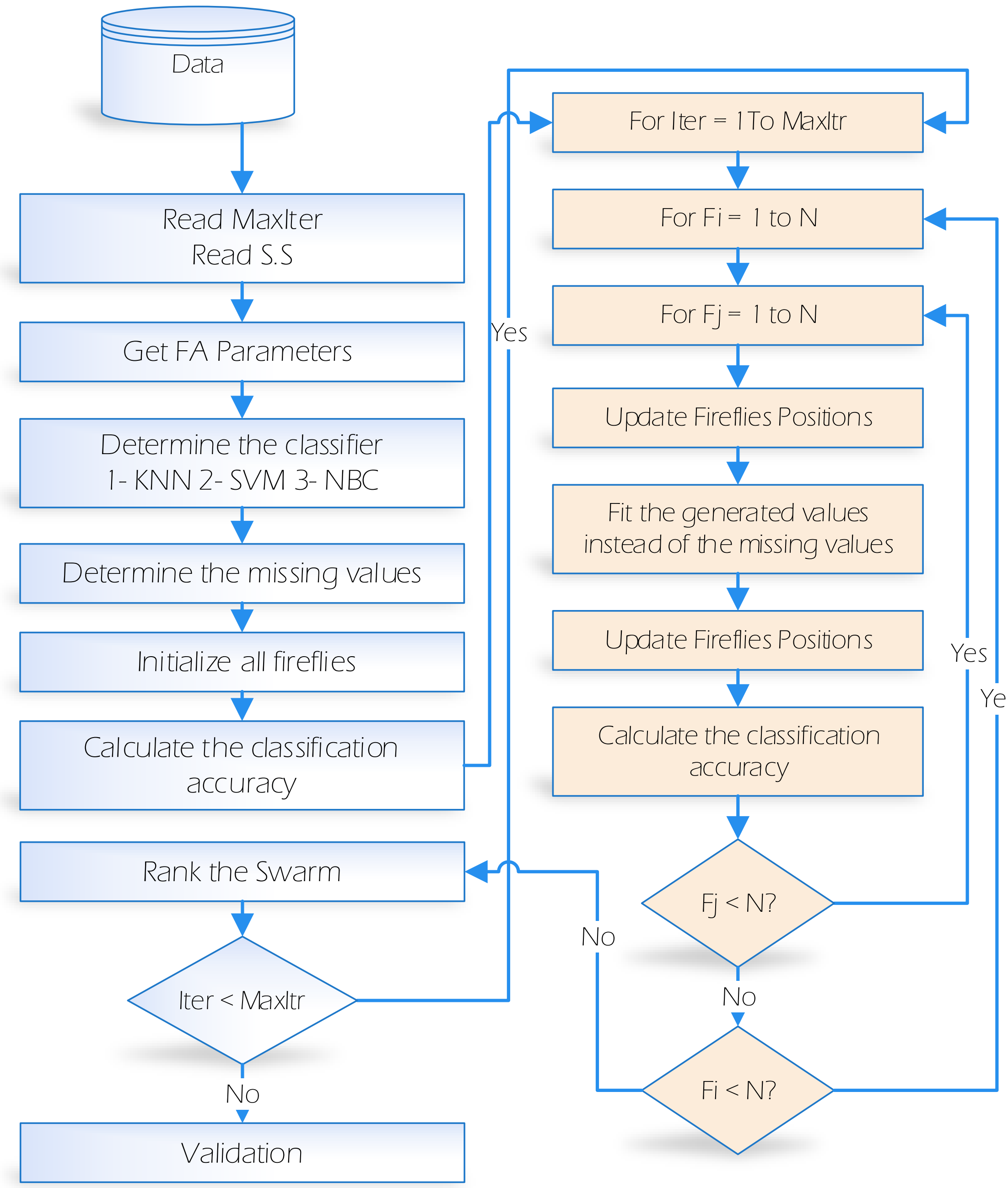


Figure 3-2 FA algorithm for predicting the missing values

CHAPTER FOUR

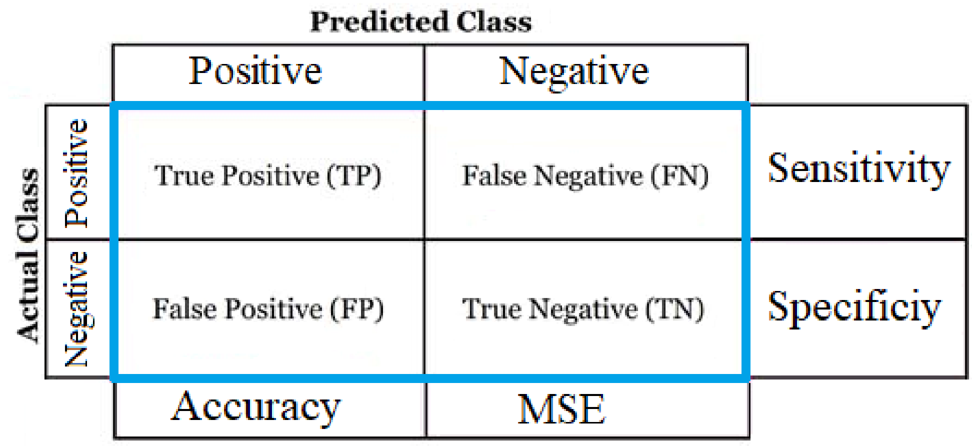
RESULTS ANALYSIS

4.1 INTRODUCTION

In this chapter, the proposed imputation algorithm based on (FA) is evaluated. In addition, the dataset used in this study which is PIDD is described and analyzed statistically in this chapter. In the first section, the evaluation metrics are presented, while the second section, the dataset is described. Final section presents the obtained results of the proposed imputation algorithm.

4.2 PERFORMANCE EVALUATION

In any optimization algorithm integrated with a machine learning model – or a classifier – must be evaluated based on several evaluation metrics. In this study, the proposed algorithm is evaluated based on several evaluation metrices. These metrices are calculated based on four evaluation parameters, they are called : True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These parameters are derived from the confusion matrix. Confusion matrix used widely in the evaluation process of the binary classification problems. It is illustrated in the following figure.



		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN)	Sensitivity
	Negative	False Positive (FP)	True Negative (TN)	Specificiy
		Accuracy	MSE	

Figure 4-1 Confusion Matrix

Various performance measures like accuracy, sensitivity, and specificity are calculated using the matrix shown in Table 5 [7] such that:

A. Accuracy

For any measurement system, the accuracy is the number of instances correctly classified in the dataset (be it positive or negative). It is calculated as shown in Eq. 4.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (4.1)$$

B. Sensitivity

Sensitivity is the ability of a system to correctly pick out the number of patients that are down with a specific disease. Sensitivity is basically used to determine the classification system of any disease; hence, it is test of the number of true positives relative to the total number of sick persons in a population; it is expressed mathematically as follows:

$$Sensitivity = \frac{TP}{TP + FN} \qquad (3.11)$$

C. Specificity

Specificity is the correct selection of the patients devoid of any condition. Specificity is a measure of the number of true negatives relative to the total number of persons in the dataset; it is computed thus:

$$Specificity = \frac{TN}{TN + FP} \tag{3.12}$$

In any classification test, a positive result with a high degree of specificity is considered effective for eventual decisions regarding the disease type.

D. Mean Square Error (MSE)

This measure indicates the average error of the prediction or classification model, as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2 \tag{3.12}$$

Where y is the actual value, $\hat{y}$ is the predicted value, and n is the number of instances or samples in the testing dataset.

4.3 DATASET DESCRIPTION

The dataset was initially put together by the “National Institute of Diabetes and Digestive and Kidney Diseases”. The recommendations of the World Health Organization (WHO) were followed during the investigations. The subjects in this study were women who were 21 years old or more and of PIMA Indian heritage. Various researchers have previously used this dataset to develop classification systems and this was the reason for selecting this data in this study to facilitate the benchmarking process with other previous studies on the problem of PID diagnosis. This dataset consists of 768 instances and each instance is associated with 8 features. Table 4-1 showed all the features in this dataset and their numerical values, while Table 4-2 presents statistical information for all attributes in the datasets, including the ranges, mean, median, and standard deviation.

Table 4-1 The features set in the dataset

F	Name	Type
1	No. of times pregnant	Numeric
2	Plasma Glucose Concentration	Numeric
3	Diastolic Blood Pressure	Numeric (mmH_g)
4	Triceps skin fold thickness	Numeric (mm)
5	2 Hours Serum insulin	Numeric ($\mu U/ml$)
6	Body mass index	Numeric (kg/m^2)
7	Diabetes pedigree function	Numeric
8	Age	Numeric (years)

Table 4-2 Statistical Information

F	**Min**	**Max**	**Mean**	**Median**	**Std.dev**
1	0	17	3.845	3	3.370
2	0	199	120.673	117	32.282
3	0	122	69.105	72	19.356
4	0	99	2.536	23	15.952
5	0	846	79.788	30.5	115.236
6	0	82.7	32.058	32.1	8.100
7	0.078	2.42	0.474	0.375	0.332
8	21	81	33.241	29	11.760

The last value, a binary, was used for the classification task; it was partitioned into 2 classes which are “Class Zero (Non-diabetic) and Class One (Diabetic)”. The first 8 features in the dataset served as the input while the last value served as the ground truth. There are a total number of 268 Diabetic cases (34.90%) in the dataset while non-diabetic cases accounted for 65.10% (500 cases).

The missing data in most of medical case studies is a standard issue, for two main reasons. First, some of the medical tests are above the budget of the patients so they can not afford them. Second, sometimes the values were not recorded correctly due to the time constraints. These missing values may effect on the classification performance. PIMA dataset is also associated with a large percentage of missing data as depicted in Table 4-3. All the features contain missing values, except the first feature where there are no missing values in it.

Table 4-3 Information about missing values in the dataset

F	Name	Missing values
1	No. of times pregnant	-
2	Plasma Glucose Concentration	5
3	Diastolic Blood Pressure	35
4	Triceps skin fold thickness	227
5	2 Hours Serum insulin	374
6	Body mass index	11
7	Diabetes pedigree function	1
8	Age	63

The histogram which illustrates the distribution of each feature is given in Figure 4-2. It can be seen from the histogram that each attribute has different distribution, meaning that the existed and missed values are have great effect on the classification performance. In addition to the histogram, Figure 4-3 shows the distribution of the density of the features.

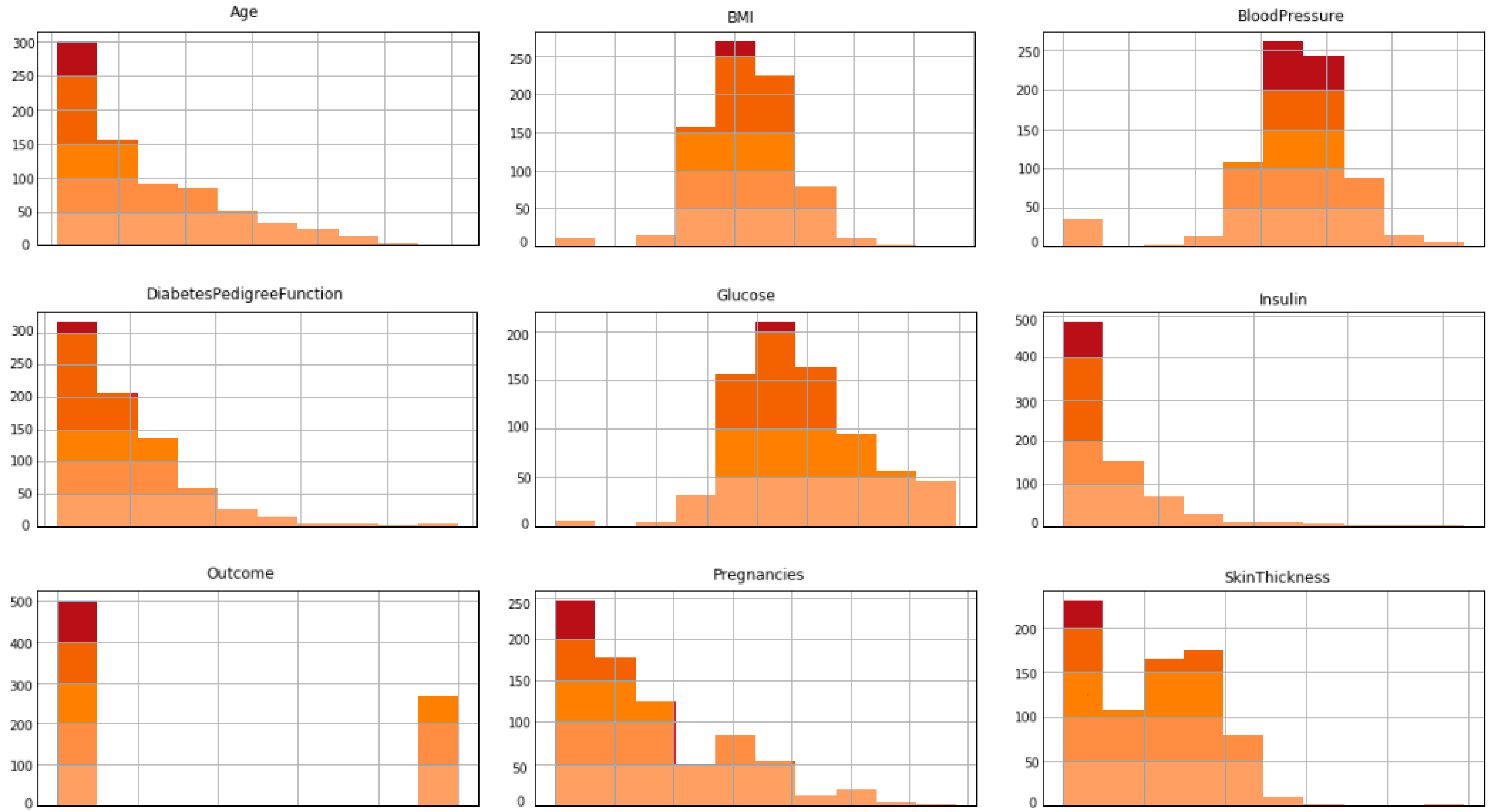


Figure 4-2 The histogram of each attribute where the X-Axis represents the values of each feature, while the Y-Axis represents the frequency of each value in the dataset.

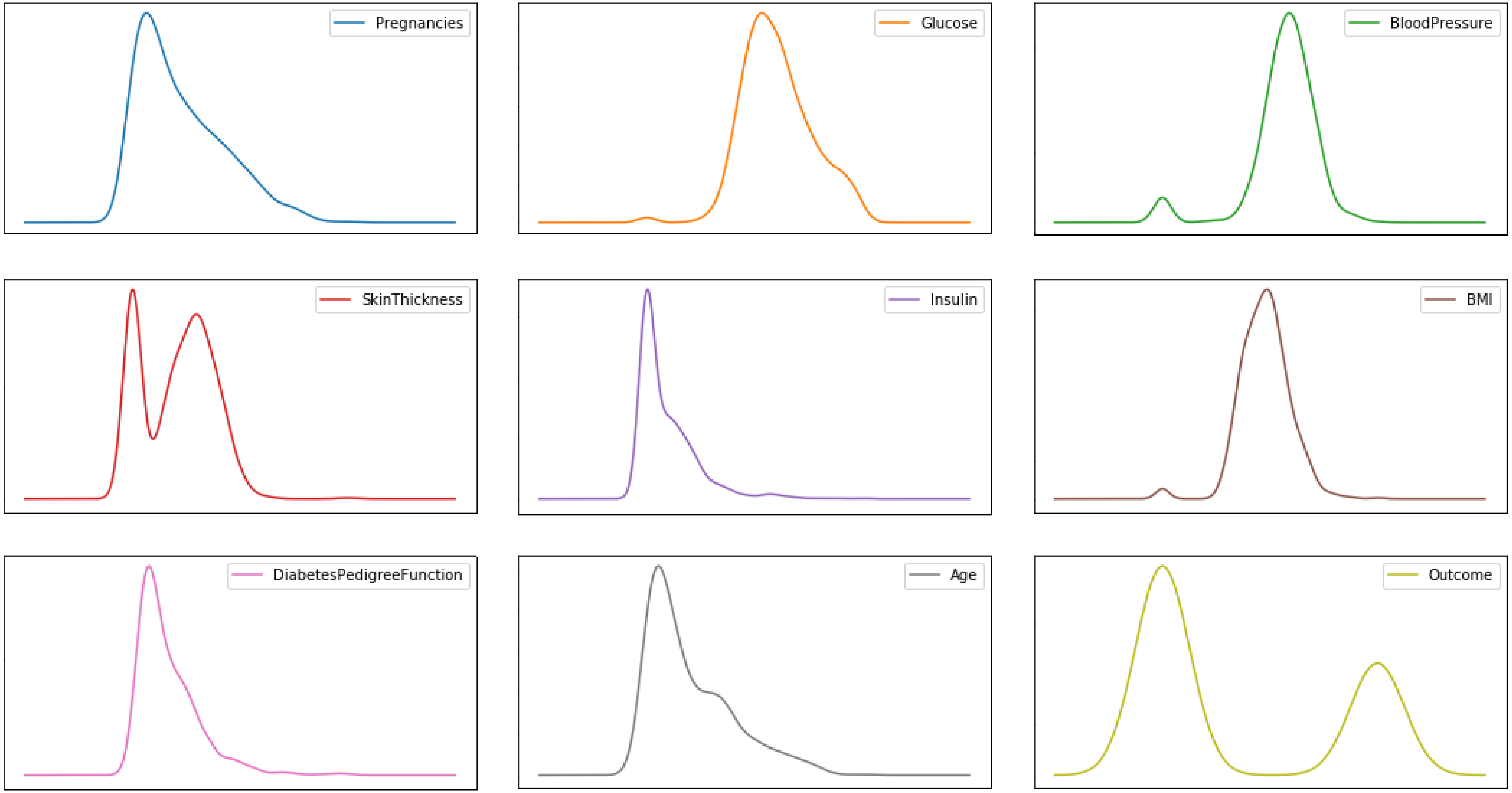


Figure 4-3 The density of each feature where the X-Axis represents the values of each feature, while the Y-Axis represents the frequency of each value in the dataset.

In order to statistically evaluate the relationship among the features, Person Correlation Coefficient is used. It is the best method of measuring the dependencies among the features. Figure 4-4 shows the correlations among all variables in the dataset.

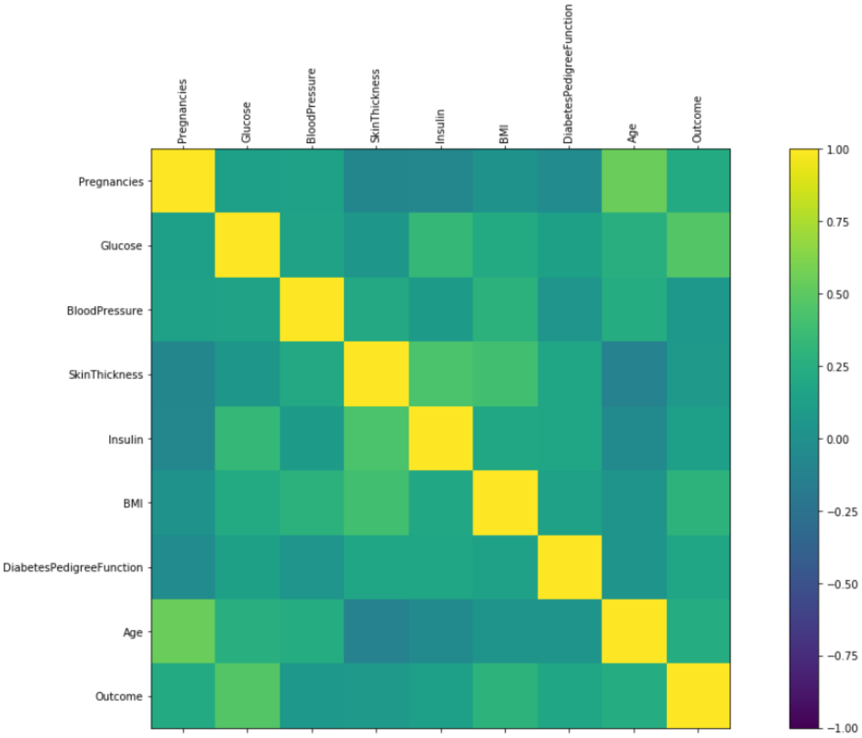


Figure 4-4 Correlation Coefficient for each feature

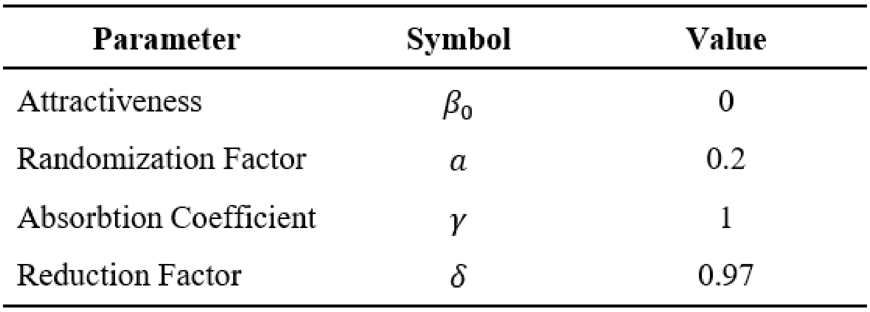
4.4 RESULTS AND DISCUSSION

In order to evaluate the performance of proposed imputation algorithm, a set of experiments should be implemented. The evaluation process consists of several experiments, each experiment consists different test settings. The imputation algorithm has been written and executed using MATLAB, version 2018b, and implemented in the environment of Windows 10 with CPU 2.6GH-64bit, and RAM 8GB.

4.4.1 Testing and Experimenting Settings

As stated in the previous chapter, the imputation algorithm including FA require several controlling parameters. Table 4-4 below presents all the required parameters for FA.

Table 4-4 Parameters of Firefly Algorithm



Parameter	Symbol	Value
Attractiveness	β_0	0
Randomization Factor	a	0.2
Absorbtion Coefficient	γ	1
Reduction Factor	δ	0.97

On the other hand, the settings of the experiments depend mainly on the structural parameters, which are : number of iterations (ITR), and the number of solutions in the swarm (N). In order to validate the effect of these two parameters on the performance of the algorithm, several values of each one are implemented, as follows:

- Case 1: Based on N: Changing the number of solutions has an impact on the performance of any nature optimization algorithm, sometimes, the large size of N enhances the performance, however this may effects on the speed of the algorithm. Therefore, in order to determine the best N as much as possible, several tests are performed, $N = \{10,15,20,30\}$.
- Case 2: Based on ITR: The number of iterations has another impact on the performance of the optimization algorithms. In order to determine the best possible ITR, several test are performed where $ITR = \{25, 50, 100, 200\}$.
- Case 3: Based on Classifier: As explained in the previous chapter, the fitness function of proposed imputation algorithm depends on three different classifiers. In other words, there three different versions of the proposed imputation algorithm, Imputation Firefly Algorithm with K-Nearest Neighbors (IFA-KNN), Imputation Firefly Algorithm with Support Vector Machine (IFA-SVM), and Imputation Firefly Algorithm with Naïve Bayesian Algorithm (IFA-NBC).

The settings of the tests can be summarized in Table 4-5, each test was executed 10 run times. The obtained results of each test are:

- Beginning Accuracy ($B.Acc$): represents the obtained accuracy based on the original dataset with missing values.
- K-Fold Cross validation ($CV.Acc$): represents the obtained accuracy using the proposed imputation algorithm.
- Original Holdout Accuracy ($OR_c.Acc$): represents the obtained accuracy based on different classifiers and the original dataset, when the dataset is divided into training set (65%) and testing set (35%).
- Optimized Holdout Accuracy ($OP_c.Acc$): represents the obtained accuracy based on different classifier and the enhanced dataset, when the enhanced dataset is divided into training set (65%) and testing set (35%).

Table 4-5 Tests Settings

Test	$\boldsymbol{N}$	$\boldsymbol{ITR}$
$\boldsymbol{T_1}$	10	25
$\boldsymbol{T_2}$	10	50
$\boldsymbol{T_3}$	10	100
$\boldsymbol{T_4}$	10	200
$\boldsymbol{T_5}$	15	25
$\boldsymbol{T_6}$	15	50
$\boldsymbol{T_7}$	15	100
$\boldsymbol{T_8}$	15	200
$\boldsymbol{T_9}$	20	25
$\boldsymbol{T_{10}}$	20	50
$\boldsymbol{T_{11}}$	20	100
$\boldsymbol{T_{12}}$	20	200
$\boldsymbol{T_{13}}$	30	25
$\boldsymbol{T_{14}}$	30	50
$\boldsymbol{T_{15}}$	30	100
$\boldsymbol{T_{16}}$	30	200

4.4.2 Results

In this subsection, the results obtained by the proposed imputation algorithm of all sixteen tests mentioned in the previous subsections are presented. Each test is illustrated in tables and figures. The results are divided into three main parts, a) KNN, b) SVM, and c) NBC.

A) Results obtained using KNN as a Fitness Function

In this part, KNN classification model is used for measuring the fitness of each solution or firefly in the swarm. The results of this experiments were obtained based on all $[T_1 - T_{16}]$ mentioned in Table 4-5 are presented in Appendix A, where each test has been implemented ten times. The average results of each test are summarized in two Figures, first figure illustrates the results obtained using cross validation of the original and the optimized dataset. While the second figure illustrates the comparison results obtained using holdout results of three classifier.

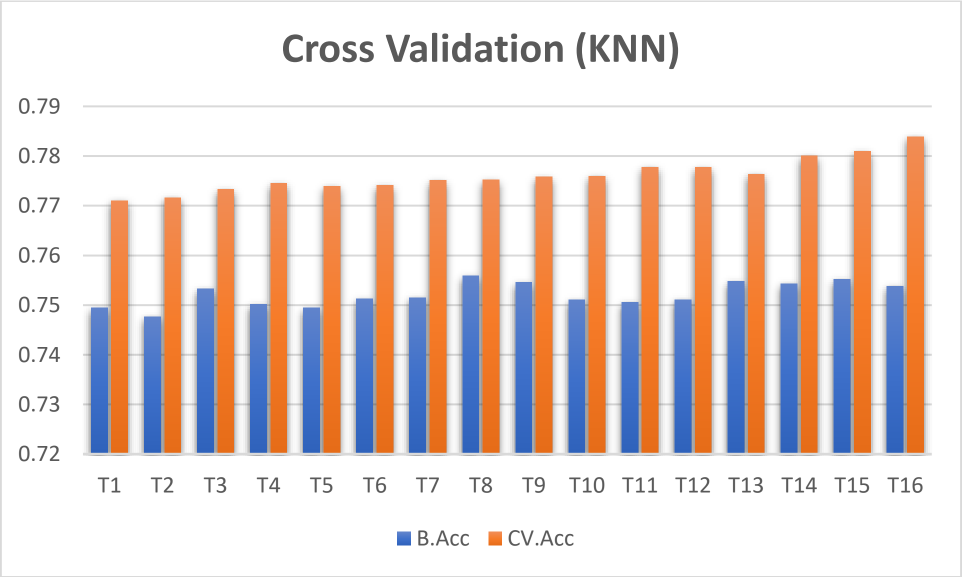


Figure 4-5 Comparison between average results of the obtained accuracies

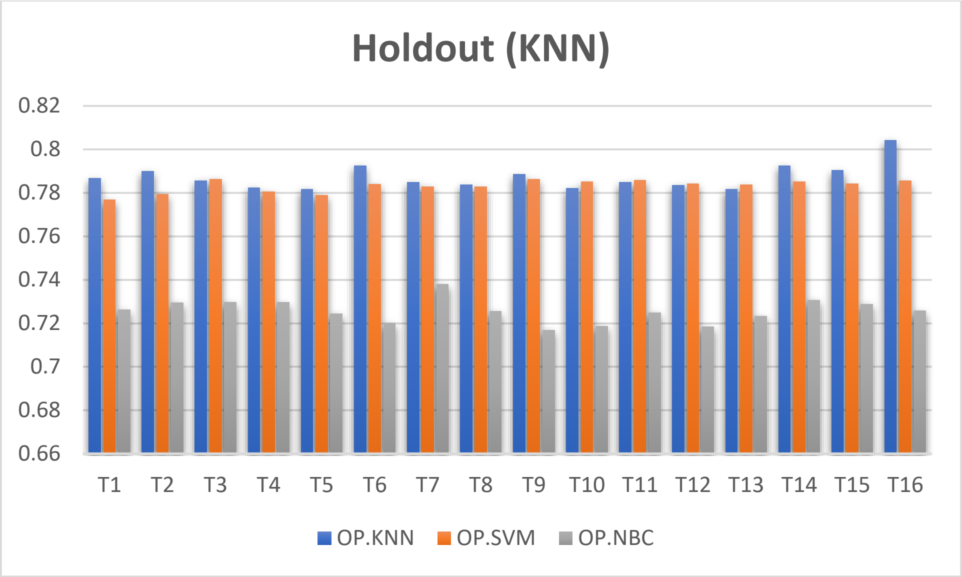


Figure 4-6 Comparison between the average accurizes using holdout

It can be seen from the above two figures that the proposed imputation algorithm based on KNN model as a fitness function has enhanced the results. In other words, the proposed algorithm estimated and filled the missing values in PIDD dataset with values better for the prediction and classification process. In addition, it can be seen in the second figure that KNN model has the best performance when it was used for the validation of the generated dataset, as compared to the others two classifiers. However, SVM has a very close performance to KNN, while the performance of NBC was the worst.

B) Results obtained using SVM as a Fitness Function

In this experiment, SVM classification model is used for evaluating the solutions in the swarm. The experiments have been validated based on the test mentioned in Table 4-5. Ten run times have been implemented, the average of these runs are presented in the two figures below. All results are presented in Appendix B.

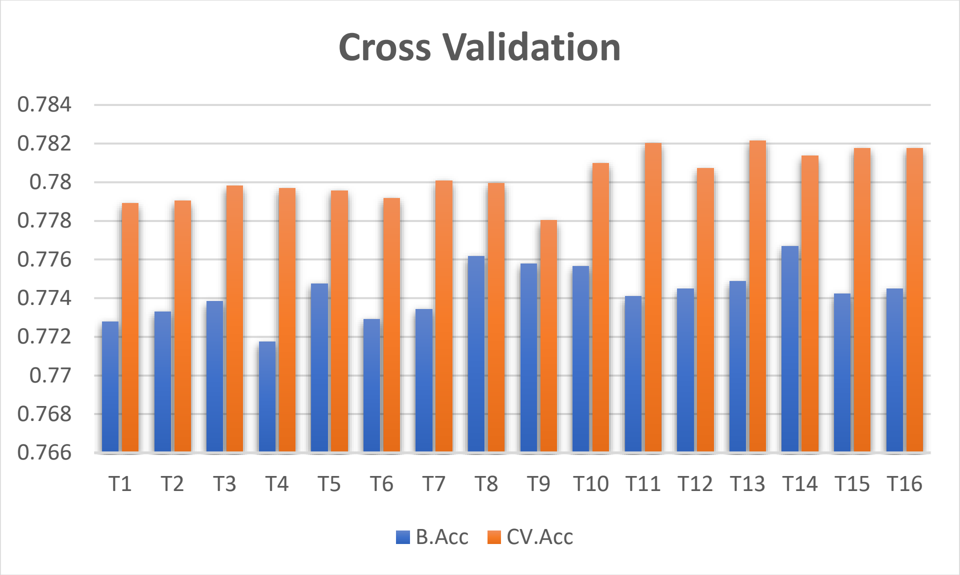


Figure 4-7 Comparison between average results of the obtained accuracies

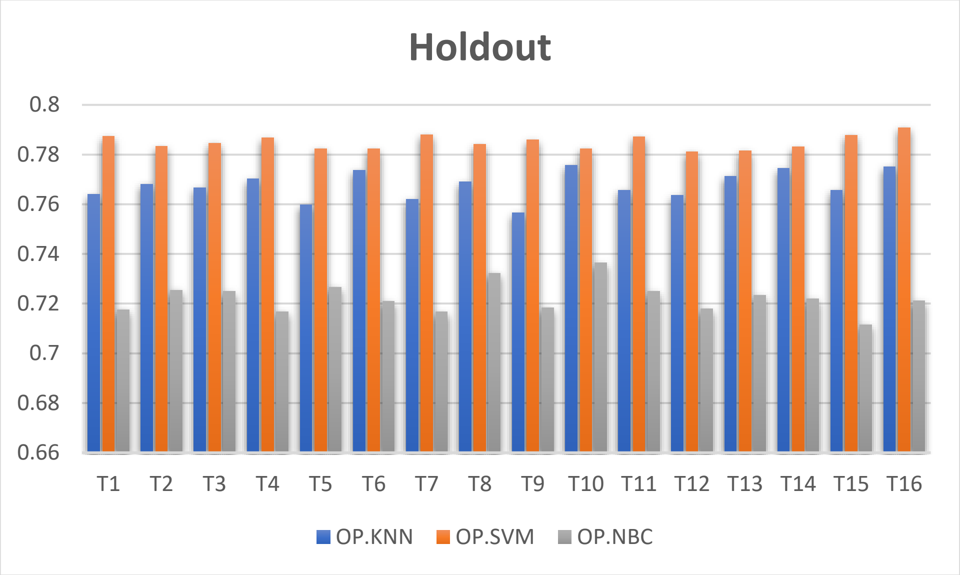


Figure 4-8 Comparison between the average accurizes using holdout

The figures above showed different results as compared to the previous experiment, as the SVM in Figure 4-8 showed a superior performance. SVM was ranked first, while NBC ranked third and attained the worst performance just like the previous experiment. On the other hand, the comparison between the obtained results in this experiment were much better than the results obtained using the original dataset with missing values (See Figure 4-7).

C) Results obtained using NBC as a Fitness Function

In the final experiment, NBC classifier is used for evaluating the generated datasets or the solutions in the swarm. The algorithm has been implemented ten run times based on the tests mentioned in Table 4-5. All the results are presented Appendix C, while the average of these runs are presented in the figures below.

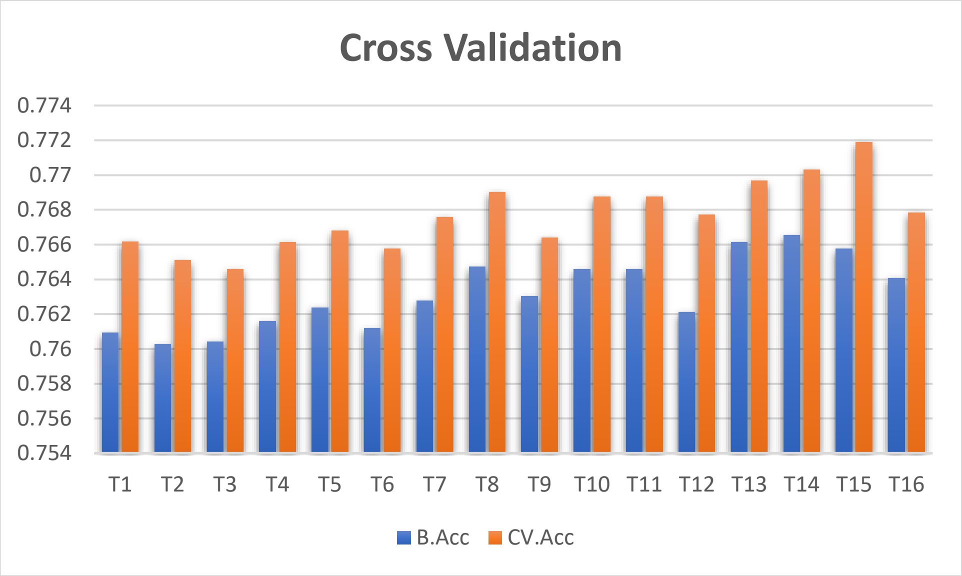


Figure 4-9 Comparison between average results of the obtained accuracies

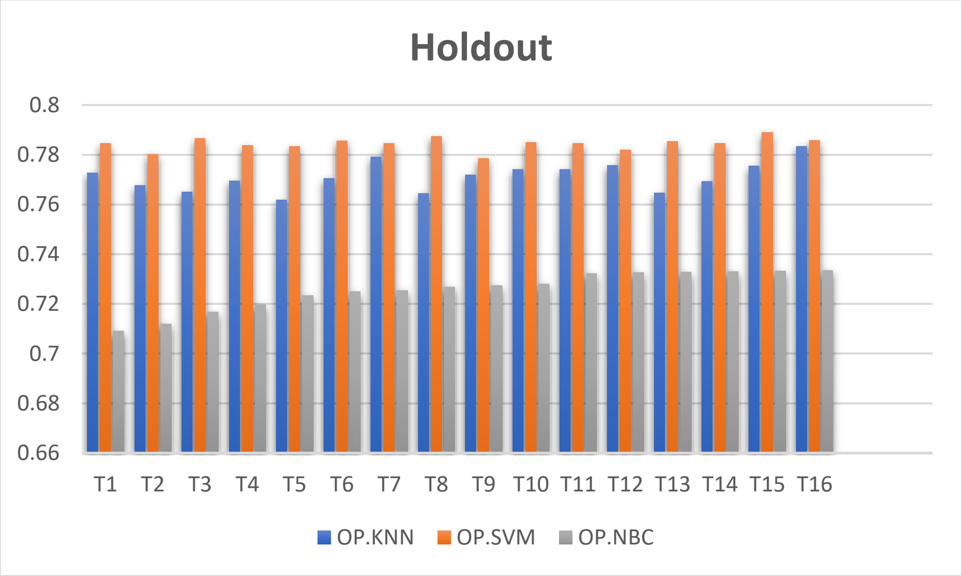


Figure 4-10 Comparison between the average accurizes using holdout

The figures above showed that NBC has the worst performance as compared to the other three classifiers. Moreover, the comparison between the accuracy obtained based on the dataset filled using the proposed imputation algorithm were better than the original dataset in all tests. Therefore, NBC enhances the performance of the proposed algorithm in general, but with worse results as compared to the other classifiers.

4.4.3 Discussion

In the previous subsection, it was clear that the proposed FA imputation algorithm based on all classifiers was able to handle the problem of the missing values in the PIDD dataset. Even the worst performance of NBC classifier was better than the best performance of all test based on the original dataset. Moreover, there are three observations can be summarized as follows:

1- When KNN used as a fitness function, the holdout validation experiments showed that KNN classifier based on the 35% testing set was better than the other classifiers. However, KNN ranked the second position when SVM or NBC used as fitness functions. In general, SVM showed the best performance due to the Sequential Minimum Optimization (SMO) algorithm for tuning the C and γ in the RBF kernel function.

2- All of the results obtained using SVM and KNN were more than 77%, while the results obtained using NBC were in range [70%, 75%].

3- It can be seen from cross-validation experiments, that the results were better when the number of the solutions – or the swarm size – are increased (i.e., Tests $T_{10} - T_{16}$). Meaning that the number of solutions has an obvious impact on the searching performance of FA. On the other hand, the number of iterations (ITR) has a less impact on FA.

The evaluation measurements other than the classification accuracy (explained in Section 4.2) are presented in the following table:

Table 4-6 Evaluation Measurements

Algorithm	Sensitivity	Specificity	MSE
IFA-KNN	0.520583	0.862484	0.159723
IFA-SVM	0.532583	0.873494	0.154563
IFA-NBC	0.489166	0.762887	0.161783

4.5 RESULTS COMPARISON

In the previous subsections, the proposed FA imputation algorithm based on different classifiers was evaluated. The evaluation process depended mainly on sixteen tests, and two validation methods: Cross Validation and Holdout. In this section, the proposed imputation algorithm is benchmarked and compared against four well-known imputation approaches on PIDD dataset. These approaches are:

a. A_1 : Removing the entire row with the missing values or attributes. This approach leads to decrease the amount of training data which may effect on the classification process.

b. A_2 : Replacing the missing values with zeros. In some cases, this could be a good solution, however, the value of zero may also effect on the classification process when the classification model is trained based on modified data.

c. A_3 : Replacing the missing values by the average or mean of the other values of the attribute. In most cases, this approach is better than the previous approaches because the generated values depend mainly on the other values of the same attribute.

d. A_4 : Replacing the missing values by random values in the range [0,1]. However, this method may generate values effects on the classification models. In other words, the values may have some noise, or change the distribution of the samples.

The approaches above have been integrated with three classifiers used in this study, and executed ten run times. Then, the best, the mean, the standard deviation were recorded. Table 4-7 below presents the comparison of the four approaches against IFA-KNN, IFA-SVM, and IFA-NBC. In addition, the mentioned approach, the classification accuracy of the dataset without implemented any imputation approach is also presented.

Table 4-7 Comparison against other imputation approaches

Classifier	**Approach**	**Best**	**Mean**	**Std. Dev**
KNN	$Original$	0.75008	0.75008	0
	A_1	0.73641	0.73122	0.24782
	A_2	0.75421	0.75231	0.21412
	A_3	0.76822	0.76741	0.19321
	A_4	0.76025	0.75942	0.20411
	IFA	**0.794153**	**0.78421**	**0.18695**
SVM	$Original$	0.77935	0.77935	0
	A_1	0.75982	0.75611	0.22782
	A_2	0.76724	0.76514	0.21842
	A_3	0.77942	0.77862	0.20142
	A_4	0.77834	0.77285	0.19782
	IFA	**0.790758**	**0.78793**	**0.002744**
NBC	$Original$	0.70414	0.70414	0
	A_1	0.69842	0.69215	0.25413
	A_2	0.69624	0.69342	0.24821
	A_3	0.70128	0.70101	0.20421
	A_4	0.70431	0.70321	0.20142
	IFA	**0.73348**	**0.72569**	**0.00754**

It is obvious that the proposed imputation algorithm obtained the highest results as compared to the other approaches. A_1 with all classifiers attained the worst position, because in this approach the many samples were deleted from the dataset, which decreases the training set. The second approach A_2 had almost the same performance with slightly better results due to using zero as the value for all missing data. On the other hand, the third and fourth approaches A_3 and A_4 were better than the previous approaches because of filling the missing data with mean or random values. The generated values are better than using zero, or removing the sample with missing data, because at least these approaches filled them.

Moreover, the best attained results were obtained using IFA-KNN, however, IFA-SVM has better average results. The standard deviation proofed that both of IFA-SVM and IFA-NBC are more stable than IFA-KNN.

CHAPTER FIVE

CONCLUSION AND FUTURE WORKS

5.1 RESEARCH SUMMARY

The missing data or missing values is an issue with most of the medical datasets. It occurred for two main reasons, a) the expense of the medical tests, and b) the fault of recording all the features for time constraints or human faults. Therefore, there is a need for a specific process for reparation these missing data, this process is called “Imputation”. The main contribution of the research is the development of an imputation algorithm for filling the missing data in the medical datasets, which is Pima Indian Diabetes Disease (PIDD).

In the literature, there are several imputation techniques have been proposed for handling the problem of missing data. Such as, removing the samples with missing values, or replacing the missing values with zero, mean of the feature values, or replacing the missing values with random values. Recent studies proposed a new type of imputation techniques which are based on optimization algorithms, where the algorithms try to find the near best values for replacing the missing data other than replacing them with zero or random values.

Firefly Algorithm (FA) is a well-known nature-inspired optimization algorithm, where each firefly in the swarm represents a potential solution. In this research, Firefly Algorithm (FA) is used as an imputation method. Three different classifiers are used for evaluating the generated missing values, these classifiers are: K Nearest Neighbors (KNN), Support Vector Machine (SVM), and Naïve Bayesian Classifier (NBC).

5.2 RECOMMENDATIONS FOR FUTURE WORKS

The proposed imputation algorithm has been evaluated based two main experiments. First, using cross validation with 5 folds, while in the second experiment, the algorithm has been evaluated using holdout validation method, where the generated dataset was divided into training set (65%) and testing set (35%). In each experiment, a set of sixteen tests have been evaluated. The tests evaluated the performance of the algorithm based on different number of iterations and swarm sizes.

For future studies, we suggest the following recommendations:

1- Implementing the proposed imputation algorithm on different medical datasets, such as Heart, Parkinson, or Leaver datasets.

2- Enhancing the global search performance of the FA by using an initialization method such as chaotic map other than the uniform distribution method.

3- Using shallow and deep artificial neural networks (ANNs) instead of KNN, SVM, and NBC. The good performance of ANNs have been proved in the literature when utilized for early prediction of medical case studies.

REFERENCES

[1] F. Gorunescu, “Introduction to Data Mining,” in *Intelligent Systems Reference Library*, 2011, pp. 1–43.

[2] C. M. C. C. M. Bishop, “Pattern recognition and machine learning,” *Pattern Recognit.*, vol. 4, no. 4, p. 738, 2006.

[3] Sowmya R and Suneetha K R, “Data Mining with Big Data,” in *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, 2017, pp. 246–250.

[4] A. R. T. Donders, G. J. M. G. van der Heijden, T. Stijnen, and K. G. M. Moons, “Review: A gentle introduction to imputation of missing values,” *J. Clin. Epidemiol.*, vol. 59, no. 10, pp. 1087–1091, Oct. 2006.

[5] S. J. Choudhury and N. R. Pal, “Imputation of missing data with neural networks for classification,” *Knowledge-Based Syst.*, vol. 182, p. 104838, Oct. 2019.

[6] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, “Pattern classification with missing data: a review,” *Neural Comput. Appl.*, vol. 19, no. 2, pp. 263–282, Mar. 2010.

[7] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi, “Metaheuristic research: a comprehensive survey,” *Artif. Intell. Rev.*, vol. 52, no. 4, pp. 2191–2233, Dec. 2019.

[8] X. S. Yang, S. Deb, S. Fong, X. He, and Y. X. Zhao, “From swarm intelligence to metaheuristics: nature-inspired optimization algorithms,” *Computer (Long. Beach. Calif).*, vol. 49, no. 9, pp. 52–59, 2016.

[9] H. de Silva and A. S. Perera, “Missing data imputation using Evolutionary k- Nearest neighbor algorithm for gene expression data,” in *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2016, pp. 141–146.

[10] A. Alhroob, W. Alzyadat, I. Almukahel, and H. Altarawneh, “Missing Data Prediction using Correlation Genetic Algorithm and SVM,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 2, pp. 703–709, 2020.

[11] T. D. Pigott, “Handling Missing Data,” in *Using Propensity Scores in Quasi-Experimental Designs*, 1 Oliver’s Yard, 55 City Road London EC1Y 1SP: SAGE Publications, Ltd, 2009, pp. 245–271.

[12] D. C. Howell, “The Treatment of Missing Data,” in *The SAGE Handbook of Social Science Methodology*, 1 Oliver’s Yard, 55 City Road, London England EC1Y 1SP United Kingdom: SAGE Publications Ltd, 2011, pp. 212–226.

[13] P. D. Allison, “Handling Missing Data by Maximum Likelihood,” *SAS Glob. Forum*

2012 Stat. Data Anal., 2012.

[14] S. Nakagawa and R. P. Freckleton, “Missing inaction: the dangers of ignoring missing data,” *Trends Ecol. Evol.*, vol. 23, no. 11, pp. 592–596, Nov. 2008.

[15] J. Schefier, “Dealing with missing data,” *Res. Lett. Inf. Math. Sci.*, vol. 3, pp. 153–160, 2000.

[16] S. R. Seaman and I. R. White, “Review of inverse probability weighting for dealing with missing data,” *Stat. Methods Med. Res.*, vol. 22, no. 3, pp. 278–295, Jun. 2013.

[17] Y. C. Yuan, “Multiple Imputation for Missing Data: Concepts and New Development (Version 9.0),” *SAS Inst. Inc, Rockville, MD*, vol. 49, no. 12, pp. 1–11, 2010.

[18] A. Pedersen, E. Mikkelsen, D. Cronin-Fenton, N. Kristensen, T. M. Pham, L. Pedersen, and I. Petersen, “Missing data and multiple imputation in clinical epidemiological research,” *Clin. Epidemiol.*, vol. Volume 9, pp. 157–166, Mar. 2017.

[19] Y. Tu, “Machine Learning,” in *EEG Signal Processing and Feature Extraction*, Singapore: Springer Singapore, 2019, pp. 301–323.

[20] S. Badillo, B. Banfai, F. Birzele, I. I. Davydov, L. Hutchinson, T. Kam-Thong, J. Siebourg-Polster, B. Steiert, and J. D. Zhang, “An Introduction to Machine Learning,” *Clin. Pharmacol. Ther.*, vol. 107, no. 4, pp. 871–885, Apr. 2020.

[21] J. Gou, H. Ma, W. Ou, S. Zeng, Y. Rao, and H. Yang, “A generalized mean distance-based k-nearest neighbor classifier,” *Expert Syst. Appl.*, vol. 115, pp. 356–372, Jan. 2019.

[22] F. Bulut and M. F. Amasyali, “Locally adaptive k parameter selection for nearest neighbor classifier: one nearest cluster,” *Pattern Anal. Appl.*, vol. 20, no. 2, pp. 415–425, May 2017.

[23] N. Labuda, J. Seeliger, T. Gedrande, and K. Kozak, “Selecting Adaptive Number of Nearest Neighbors in k-Nearest Neighbor Classifier Apply Diabetes Data,” *J. Math. Stat. Sci.*, vol. 17, no. 1, pp. 1–13, 2017.

[24] R. Herbrich, “Learning kernel classifiers: theory and algorithms,” *Machine Learning*. 2002.

[25] C.-F. Chao and M.-H. Horng, “The Construction of Support Vector Machine Classifier Using the Firefly Algorithm,” *Comput. Intell. Neurosci.*, vol. 2015, pp. 1–8, 2015.

[26] A. A. Al-Musawi, A. A. H. Alwanas, S. Q. Salih, Z. H. Ali, M. T. Tran, and Z. M. Yaseen, “Shear strength of SFRCB without stirrups simulation: implementation of hybrid artificial intelligence model,” *Eng. Comput.*, vol. 36, no. 1, pp. 1–11, Jan. 2020.

[27] M. Abdar, M. Zomorodi-Moghadam, R. Das, and I. H. Ting, “Performance analysis of

classification algorithms on early detection of liver disease.," *Expert Syst. Appl.*, vol. 67, pp. 239–251, 2017.

[28] A. A. Aljumah, M. G. Ahamad, and M. K. Siddiqui, "Application of data mining: Diabetes health care in young and old patients. Journal of King Saud University," *Comput. Inf. Sci.*, vol. 25, no. 2, pp. 127–136, 2013.

[29] S. Bashir, U. Qamar, F. H. Khan, and L. Naseem, "HMV: a medical decision support framework using multi-layer classifiers for disease prediction," *J. Comput. Sci.*, vol. 13, p. 10–25., 2016.

[30] M. Komi, J. Li, Y. Zhai, and X. Zhang, "Application of data mining methods in diabetes prediction. In Image, Vision and Computing (ICIVC), 2017," in *2nd International Conference on (pp. 1006-1010). IEEE.*, 2017.

[31] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine learning and data mining methods in diabetes research.," *Comput. Struct. Biotechnol. J.*, vol. 15, pp. 104–116, 2017.

[32] M. Nilashi, O. bin Ibrahim, H. Ahmadi, and L. Shahmoradi, "An analytical method for diseases prediction using machine learning techniques," *Comput. Chem. Eng.*, vol. 106, pp. 212–223, 2017.

[33] F. Mercaldo, V. Nardone, and A. Santone, "Diabetes Mellitus Affected Patients Classification and Diagnosis through Machine Learning Techniques.," *Procedia Comput. Sci.*, vol. 112, no. C, p. 2519–2528., 2017.

[34] J. P. Kandhasamy and S. Balamurali, "Performance analysis of classifier models to predict diabetes mellitus.," *Procedia Comput. Sci.*, vol. 47, pp. 45–51, 2015.

[35] S. Perveen, M. Shahbaz, A. Guergachi, and K. Keshavjee, "Performance analysis of data mining classification techniques to predict diabetes.," *Procedia Comput. Sci.*, vol. 82, pp. 115–121, 2016.

[36] V. V. Kamadi, A. R. Allam, and S. M. Thummala, "A computational intelligence technique for the effective diagnosis of diabetic patients using principal component analysis (PCA) and modified fuzzy SLIQ decision tree approach.," *Appl. Soft Comput.*, vol. 49, pp. 137–145, 2016.

[37] K. R. Pradeep and N. C. Naveen, "Predictive analysis of diabetes using J48 algorithm of classification techniques. In Contemporary Computing and Informatics (IC3I)," in *2016 2nd International Conference on IEEE.*, 2016, pp. 347–352.

[38] T. Santhanam and M. S. Padmavathi, "Application of K-means and genetic algorithms for dimension reduction by integrating SVM for diabetes diagnosis.," *Procedia Comput.*

Sci., vol. 47, pp. 76–83, 2015.

[39] X. H. Meng, Y. X. Huang, D. P. Rao, Q. Zhang, and Q. Liu, “Comparison of three data mining models for predicting diabetes or prediabetes by risk factors.,” *Kaohsiung J. Med. Sci.*, vol. 29, no. 2, pp. 93–99, 2013.

[40] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in *Micro Machine and Human Science, 1995. MHS’95., Proceedings of the Sixth International Symposium on*, 1995, pp. 39–43.

[41] X. S. Yang, “Firefly algorithms for multimodal optimization,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics),* vol. 5792 LNCS, pp. 169–178, 2009.

[42] X. S. Yang, *Firefly algorithm for multimodal optimization*. Luniver Press, 2008.

[43] L. Zhang, L. Shan, and J. Wang, “Optimal feature selection using distance-based discrete firefly algorithm with mutual information criterion,” *Neural Comput. Appl.*, pp. 1–14, 2016.

[44] K. Kaur, R. Salgotra, and U. Singh, “An improved firefly algorithm for numerical optimization,” *Proc. 2017 Int. Conf. Innov. Information, Embed. Commun. Syst. ICIIECS 2017,* vol. 2018–Janua, pp. 1–5, 2018.

[45] H. A. Ahmed, M. F. Zolkipli, and M. Ahmad, “A novel efficient substitution-box design based on firefly algorithm and discrete chaotic map,” *Neural Computing and Applications*, 2018.

[redacted]APPENDIX A: RESULTS OF KNN

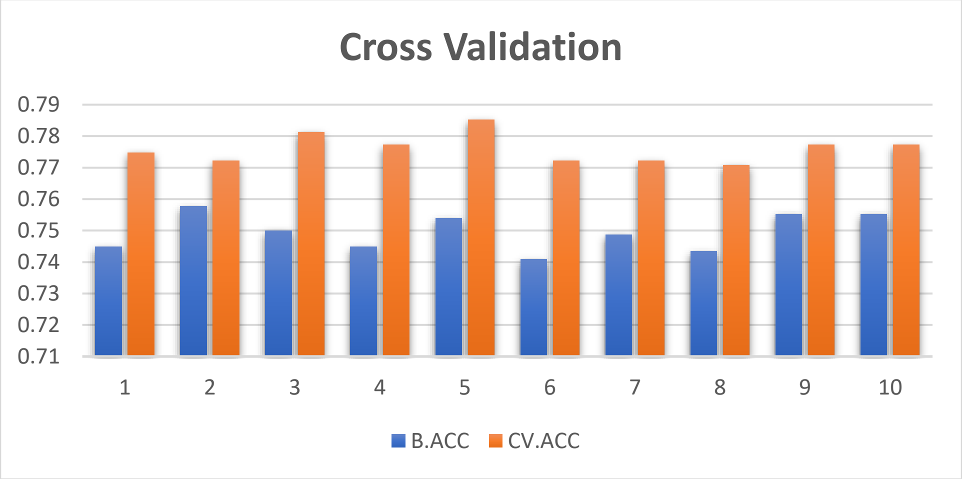


Figure A-11 Comparison between original and obtained accuracy (T_1)

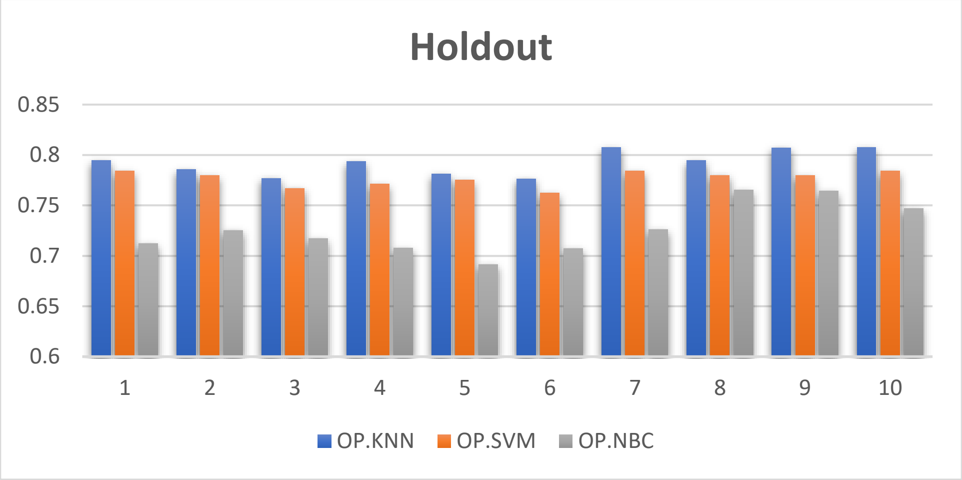


Figure A-12 Holdout results obtained using all classifiers (T_1)

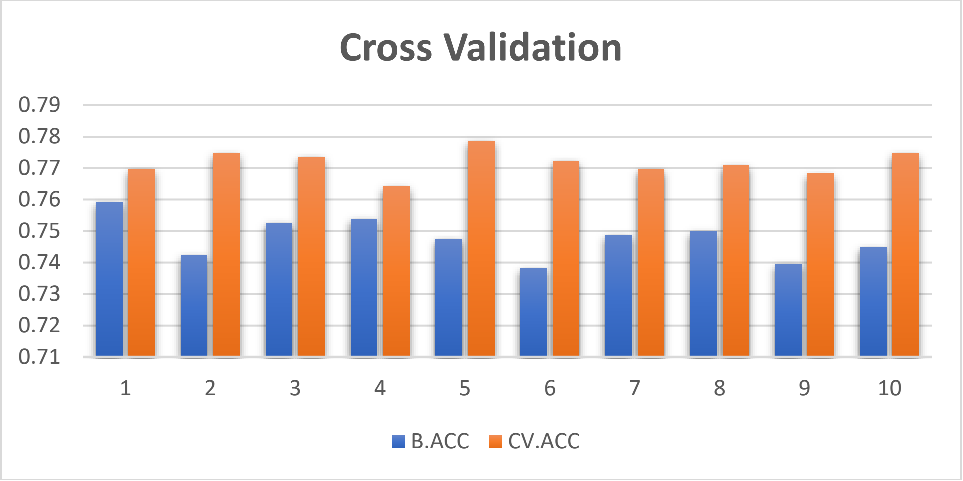


Figure A-13 Comparison between original and obtained accuracy (T_2)

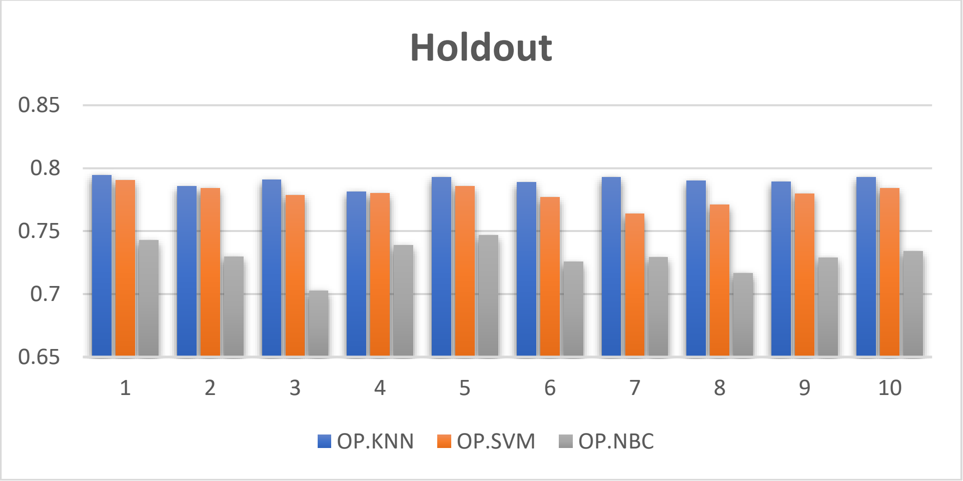


Figure A-14 Holdout results obtained using all classifiers (T_2)

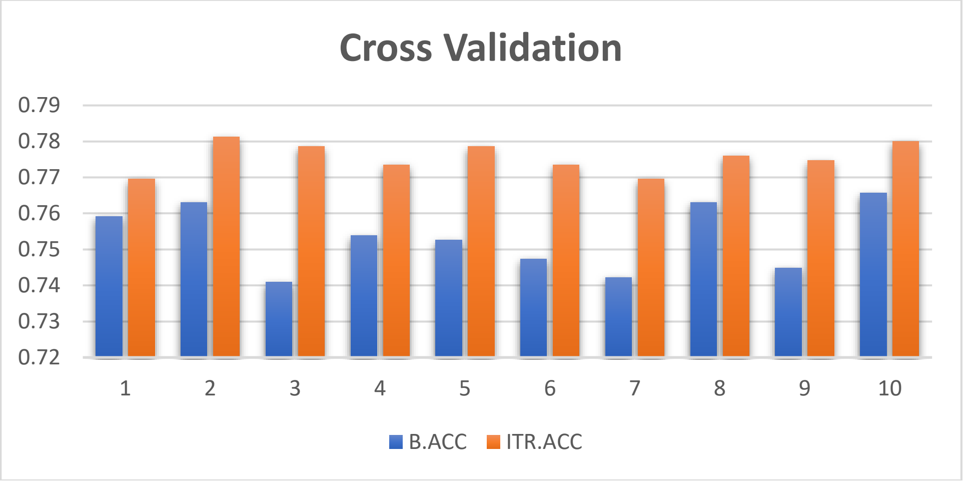


Figure A-15 Comparison between original and obtained accuracy (T_3)

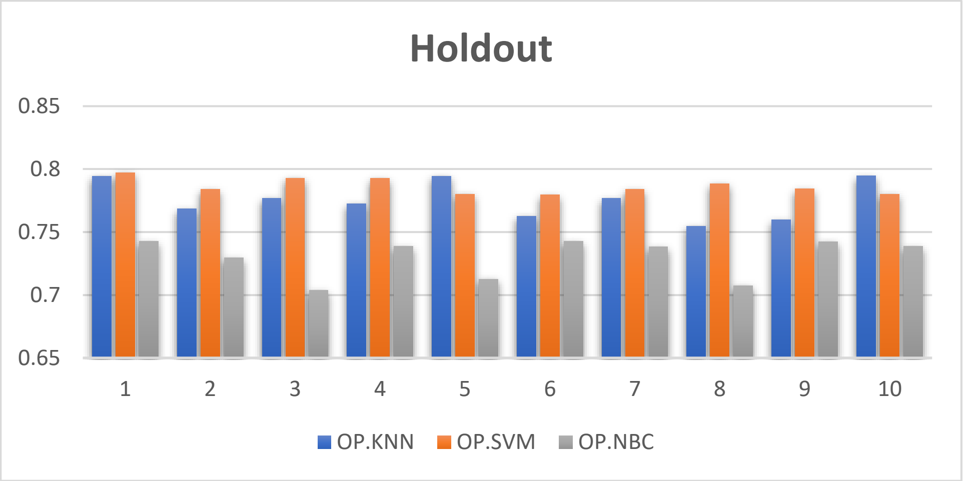


Figure A-16 Holdout results obtained using all classifiers (T_3)

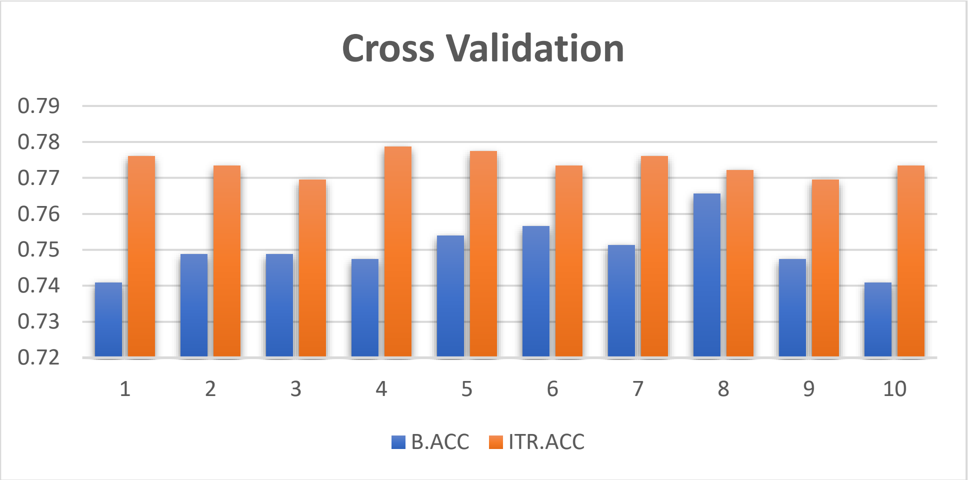


Figure A-17 Comparison between original and obtained accuracy (T_4)

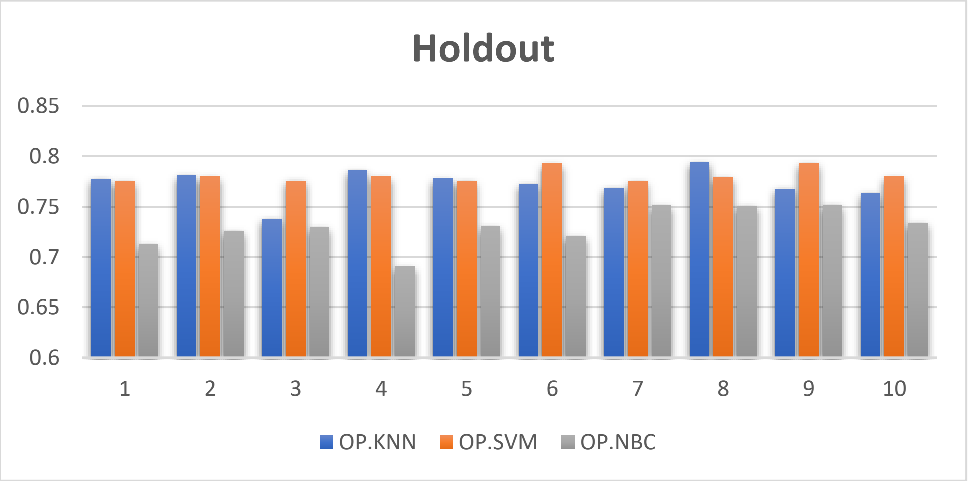


Figure A-18 Holdout results obtained using all classifiers (T_4)

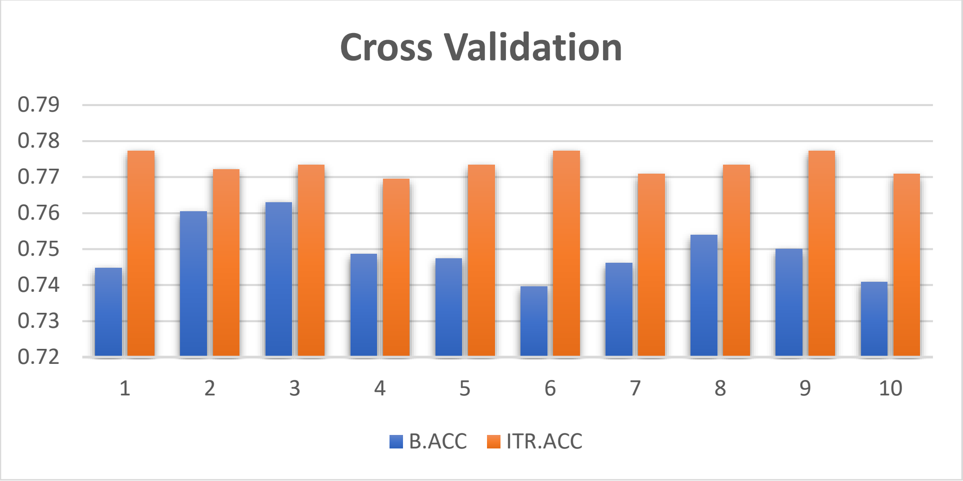


Figure A-19 Comparison between original and obtained accuracy (T_5)

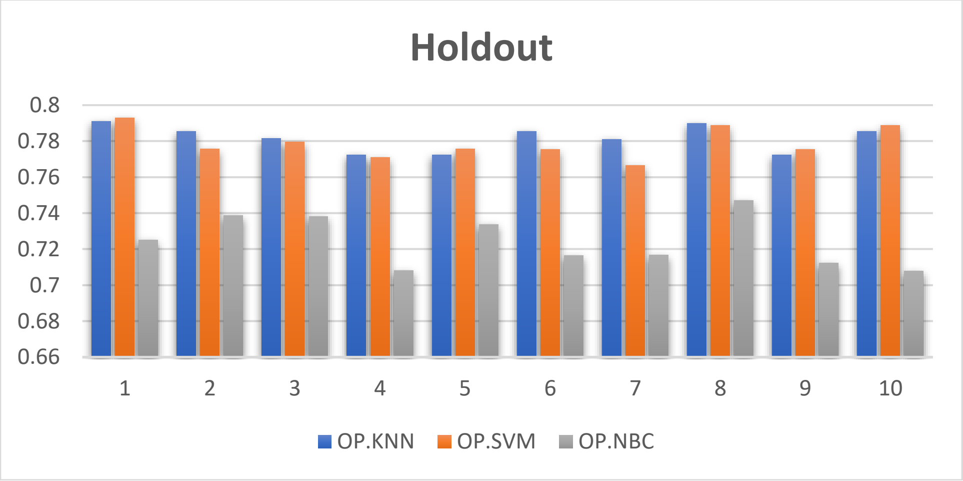


Figure A-20 Holdout results obtained using all classifiers (T_5)

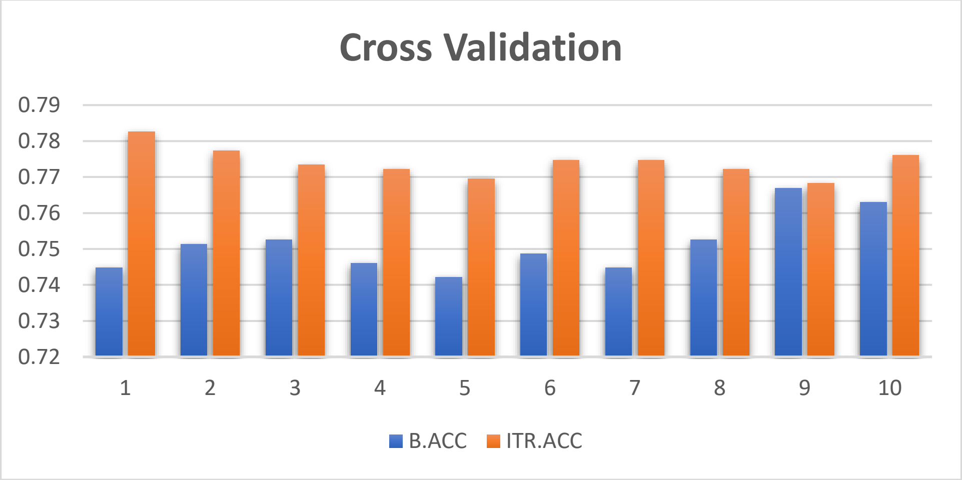


Figure A-21 Comparison between original and obtained accuracy (T_6)

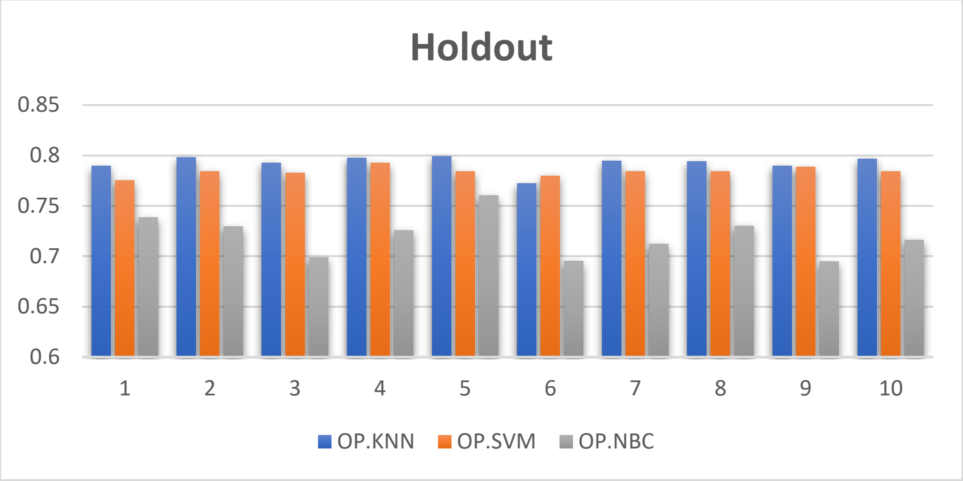


Figure A-22 Holdout results obtained using all classifiers (T_6)

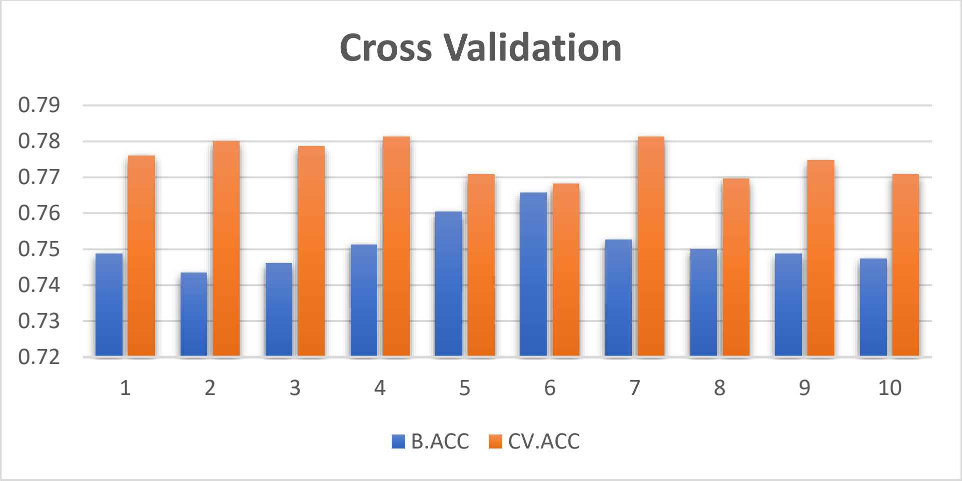


Figure A-23 Comparison between original and obtained accuracy (T_7)

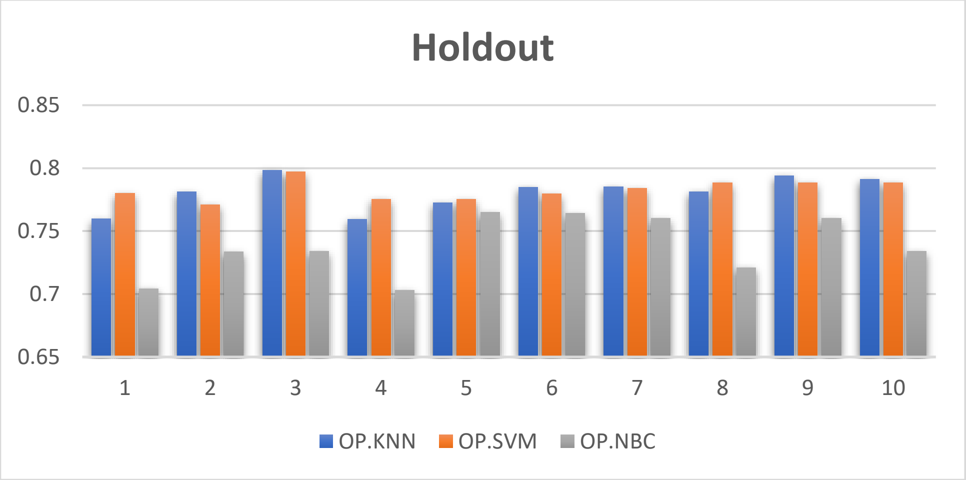


Figure A-24 Holdout results obtained using all classifiers (T_7)

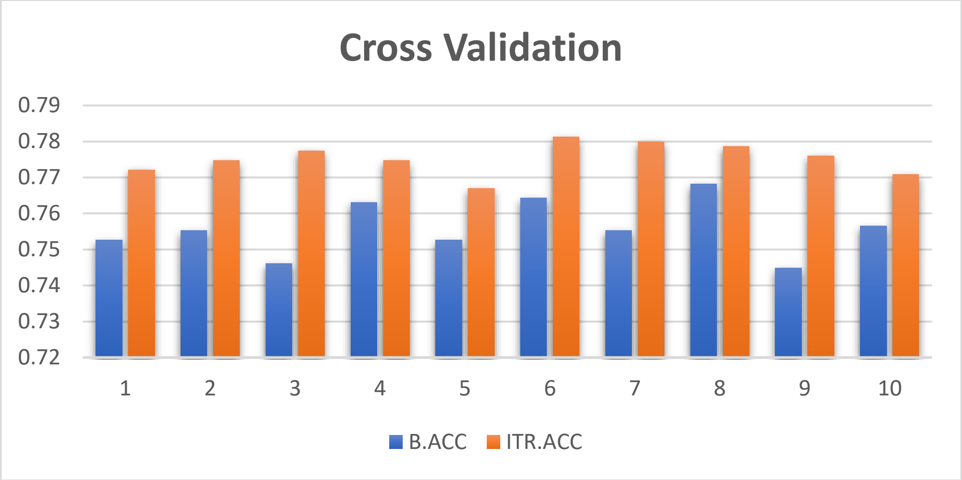


Figure A-25 Comparison between original and obtained accuracy (T_8)

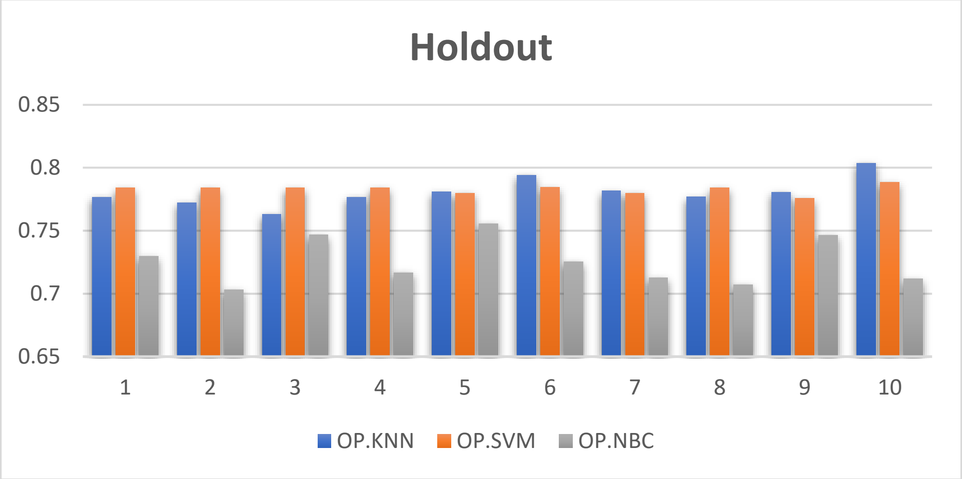


Figure A-26 Holdout results obtained using all classifiers (T_8)

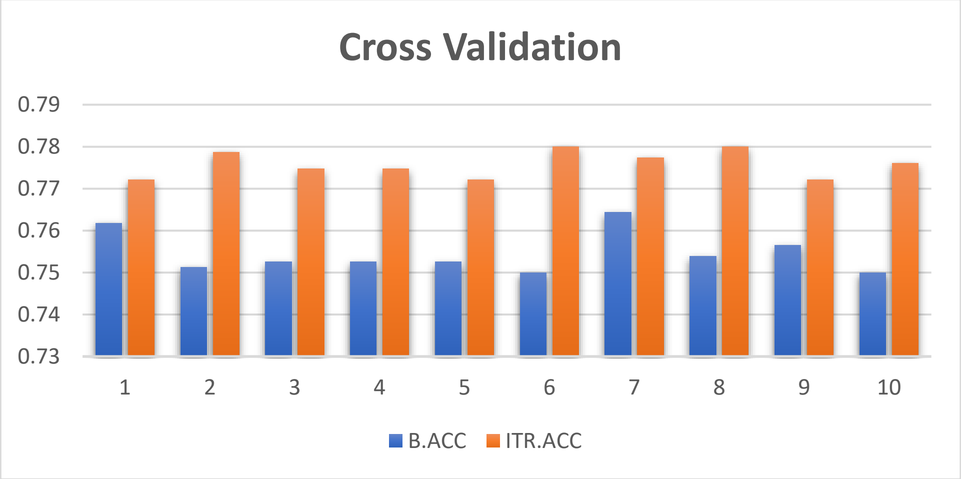


Figure A-27 Comparison between original and obtained accuracy (T_9)

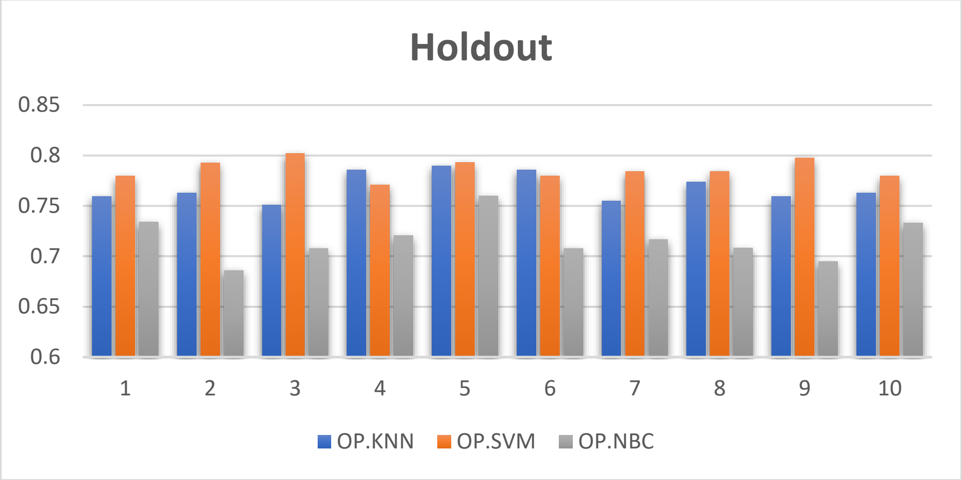


Figure A-28 Holdout results obtained using all classifiers (T_9)

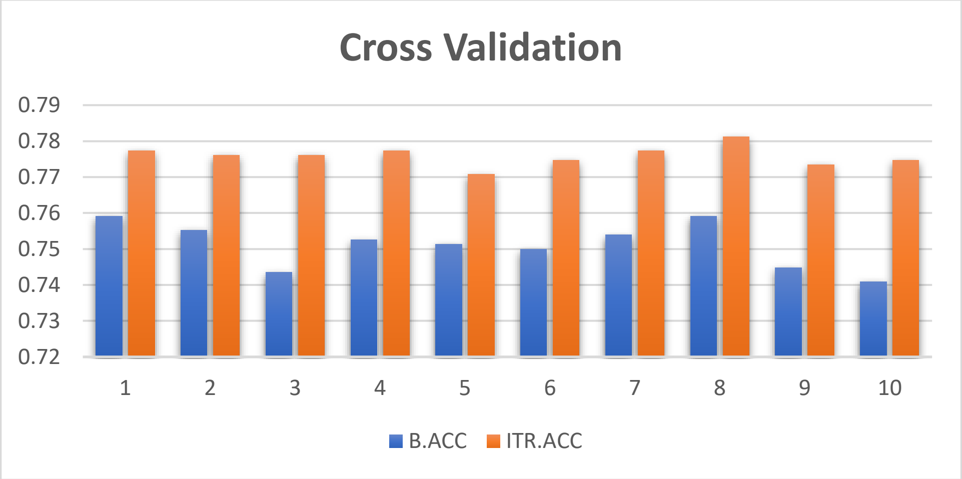


Figure A-29 Comparison between original and obtained accuracy (T_{10})

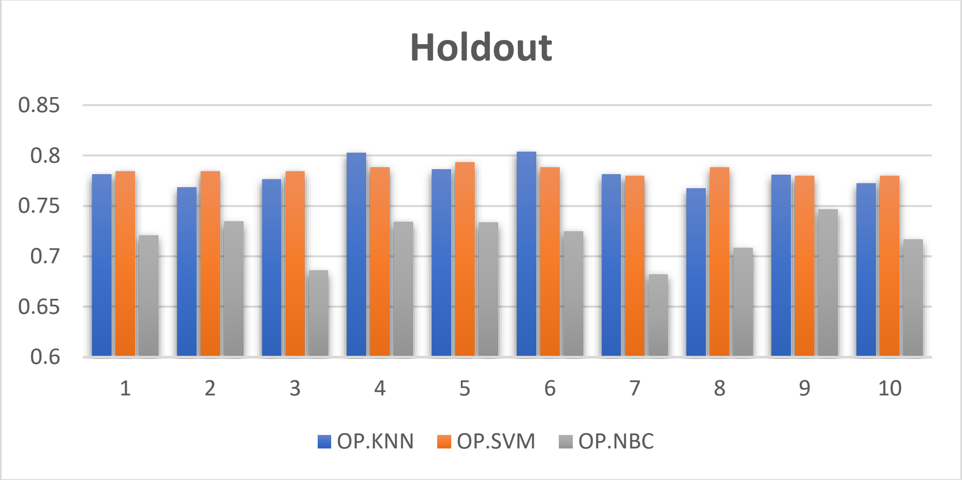


Figure A-30 Holdout results obtained using all classifiers (T_{10})

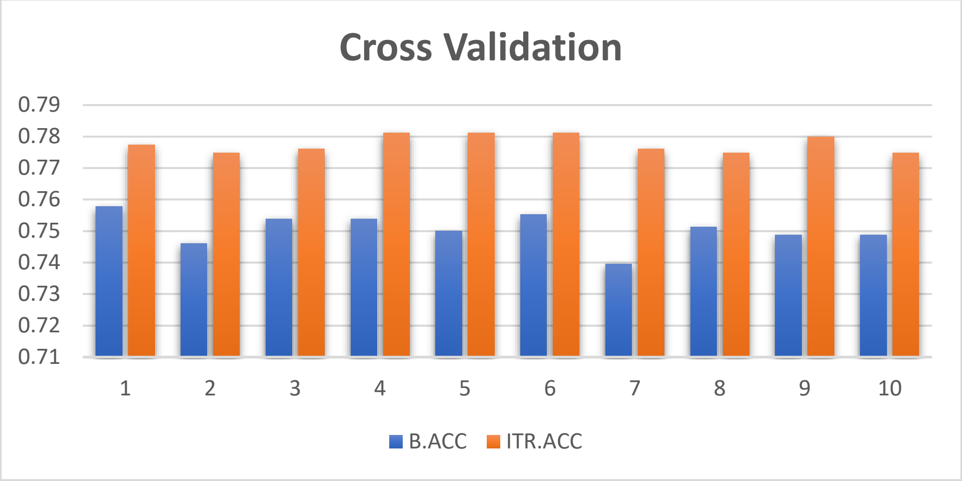


Figure A-31 Comparison between original and obtained accuracy (T_{11})

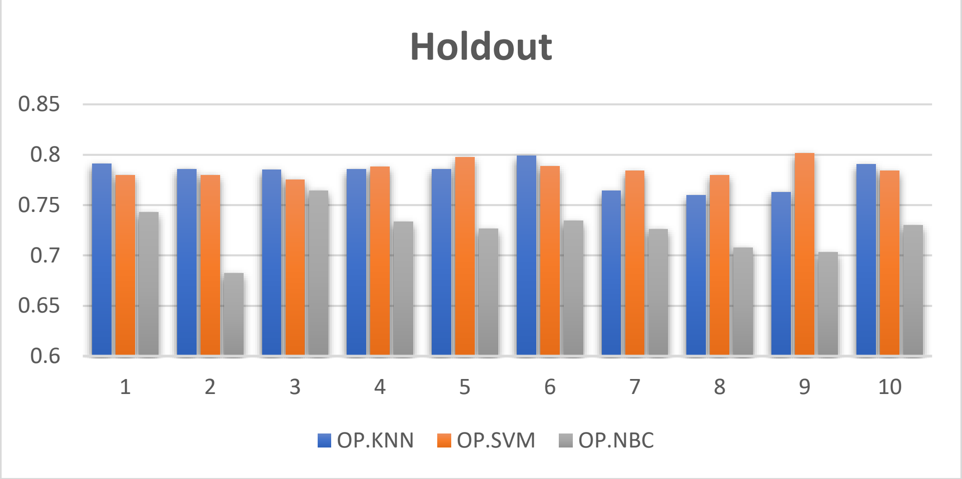


Figure A-32 Holdout results obtained using all classifiers (T_{11})

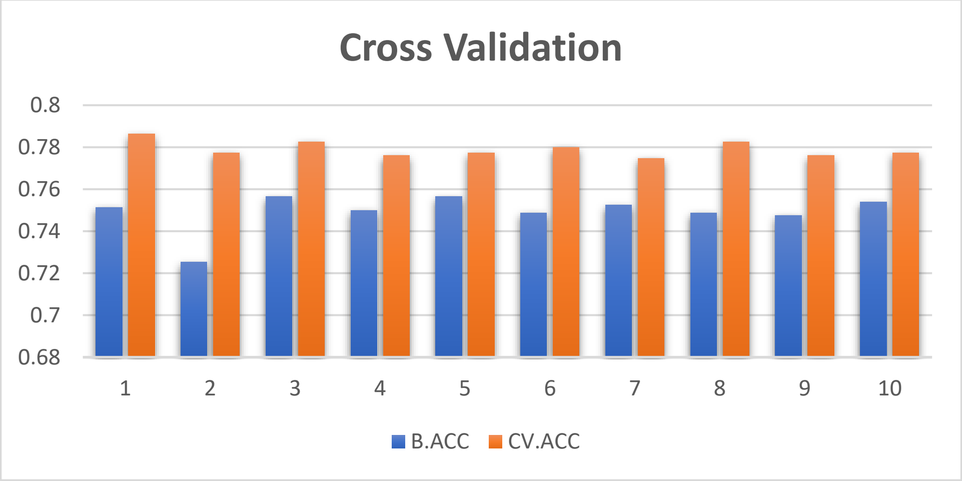


Figure A-33 Comparison between original and obtained accuracy (T_{12})

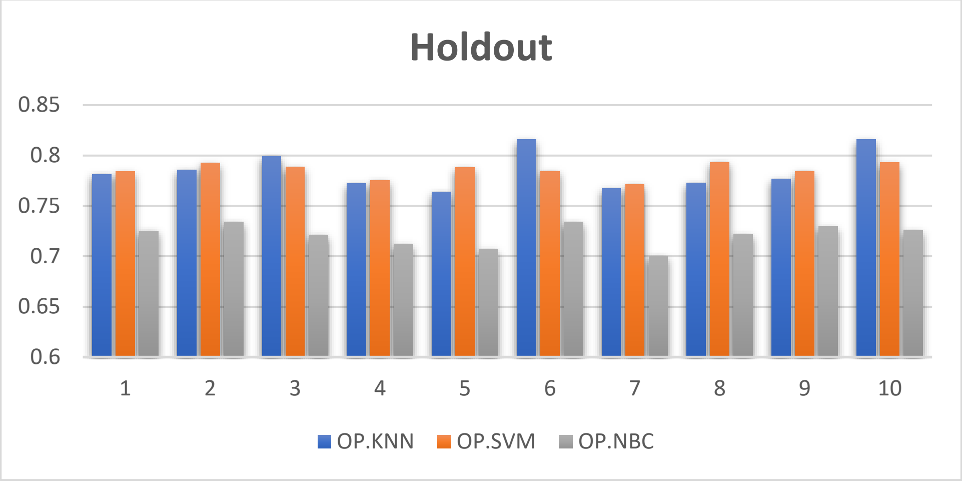


Figure A-34 Holdout results obtained using all classifiers (T_{12})

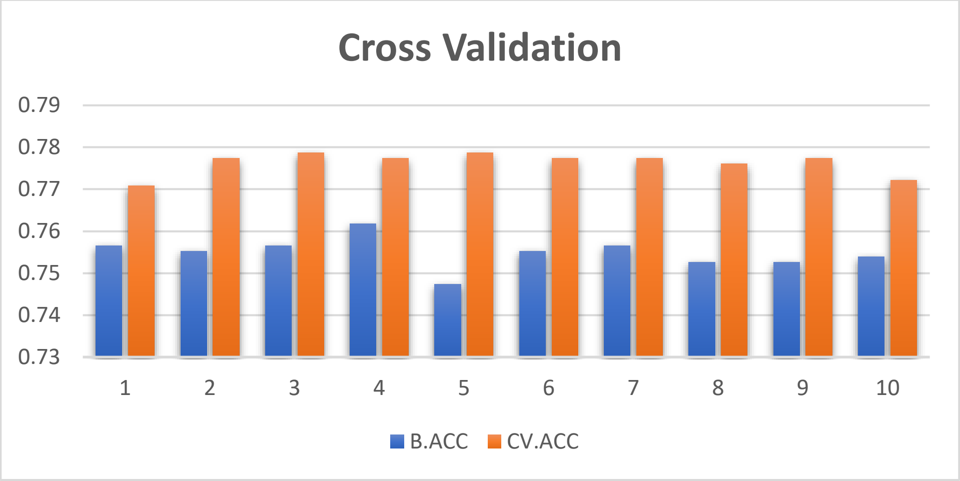


Figure A-35 Comparison between original and obtained accuracy (T_{13})

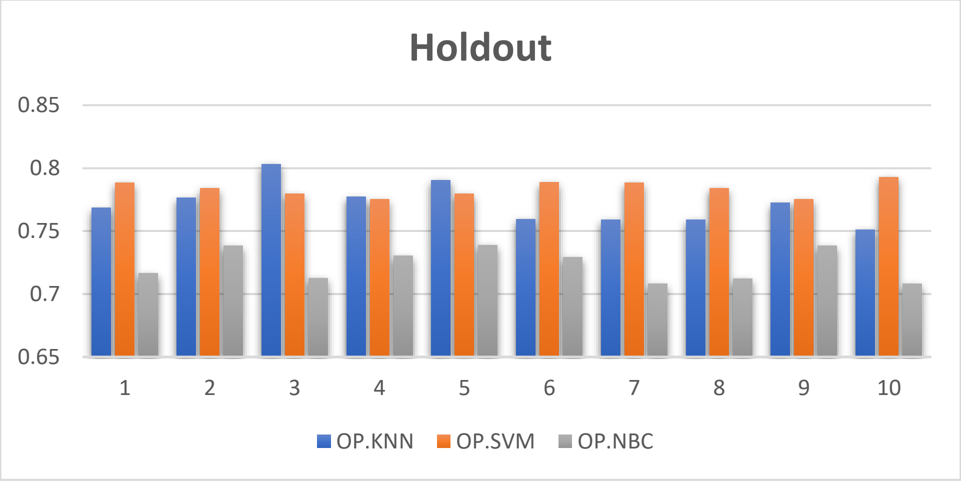


Figure A-36 Holdout results obtained using all classifiers (T_{13})

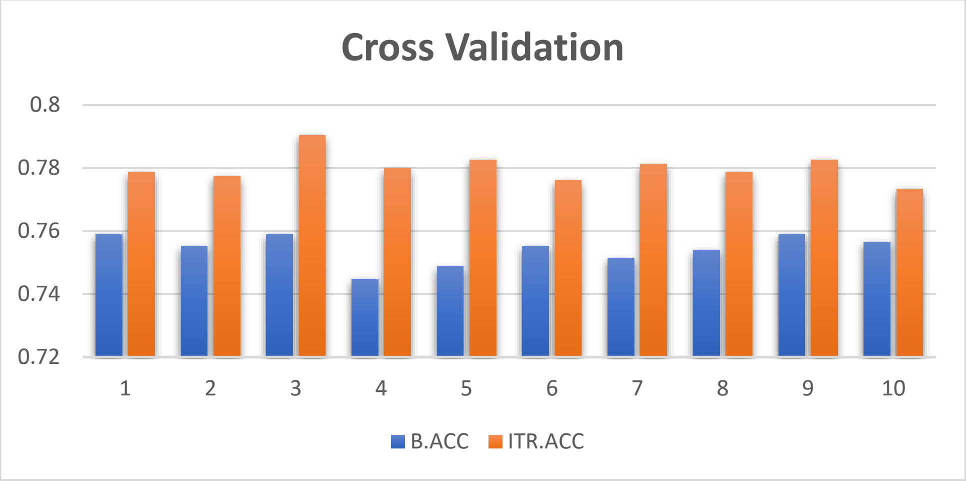


Figure A-37 Comparison between original and obtained accuracy (T_{14})

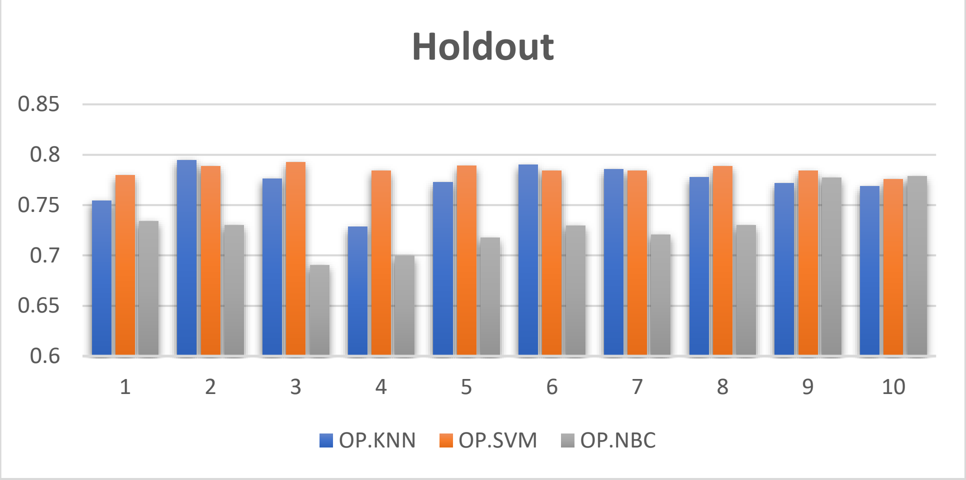


Figure A-38 Holdout results obtained using all classifiers (T_{14})

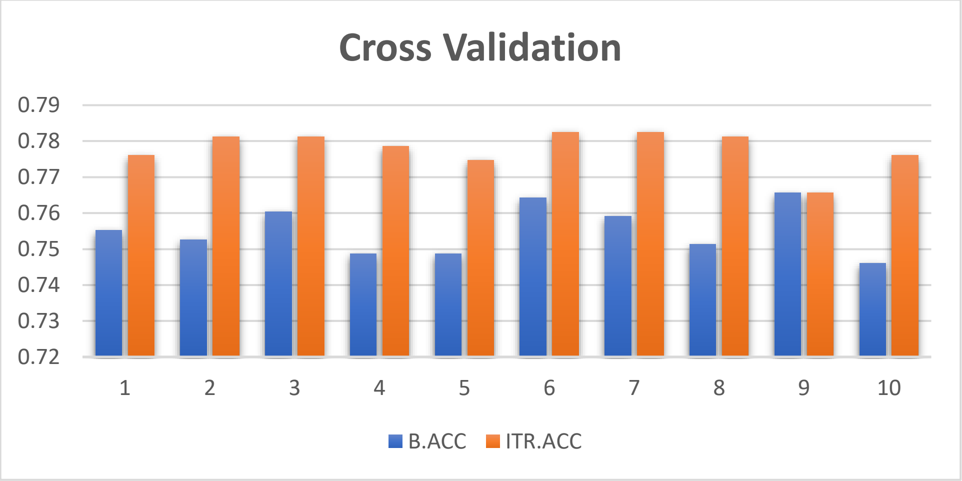


Figure A-39 Comparison between original and obtained accuracy (T_{15})

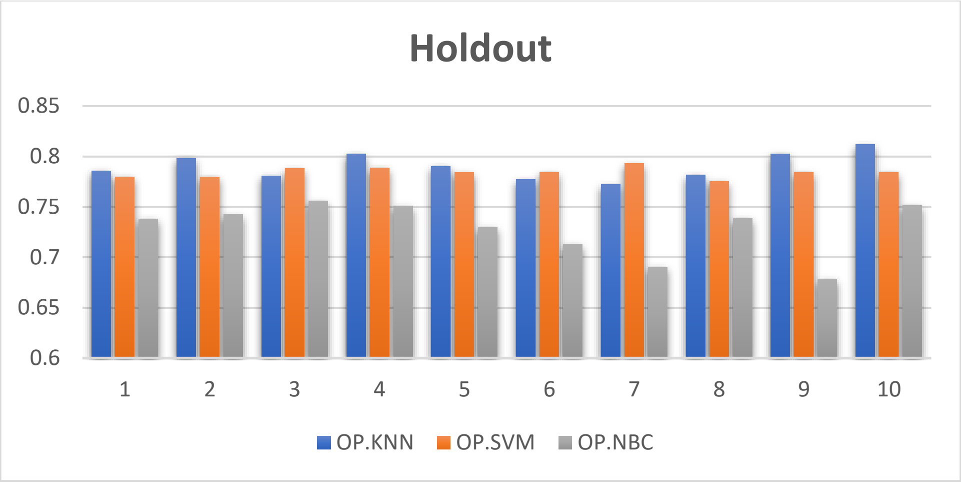


Figure A-40 Holdout results obtained using all classifiers (T_{15})

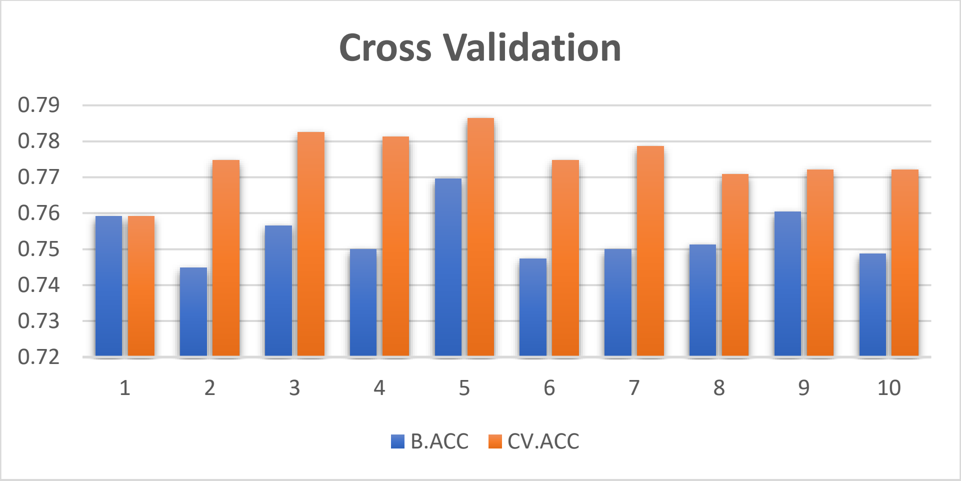


Figure A-41 Comparison between original and obtained accuracy (T_{16})

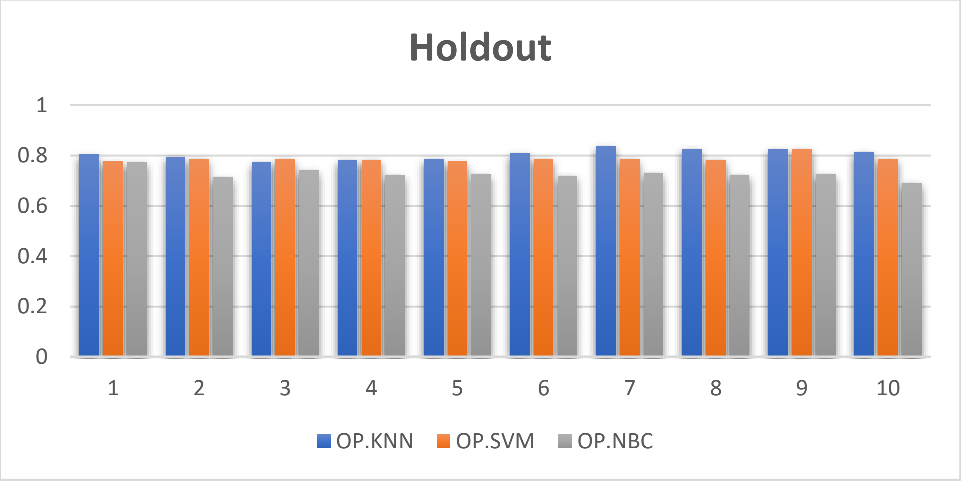


Figure A-42 Holdout results obtained using all classifiers (T_{16})

[redacted]APPENDIX B: A RESULTS OF SVM

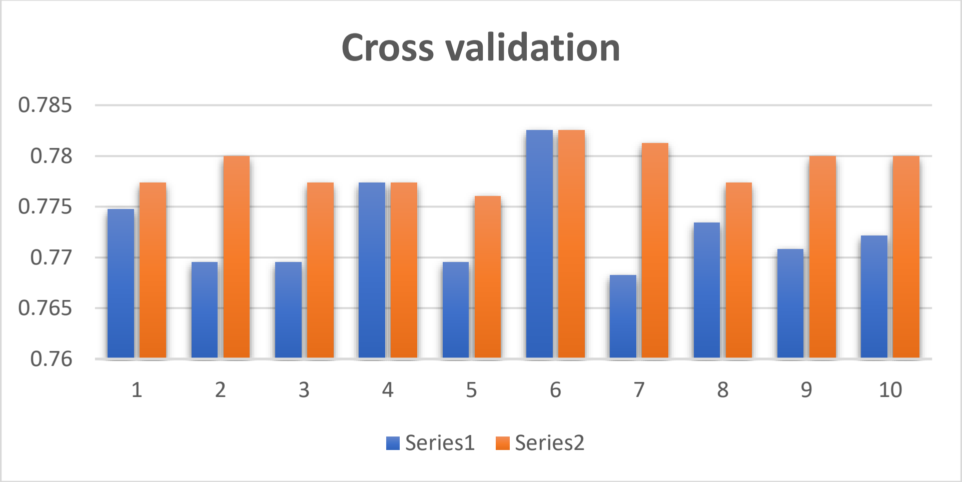


Figure B-1 Comparison between original and obtained accuracy (T_1)

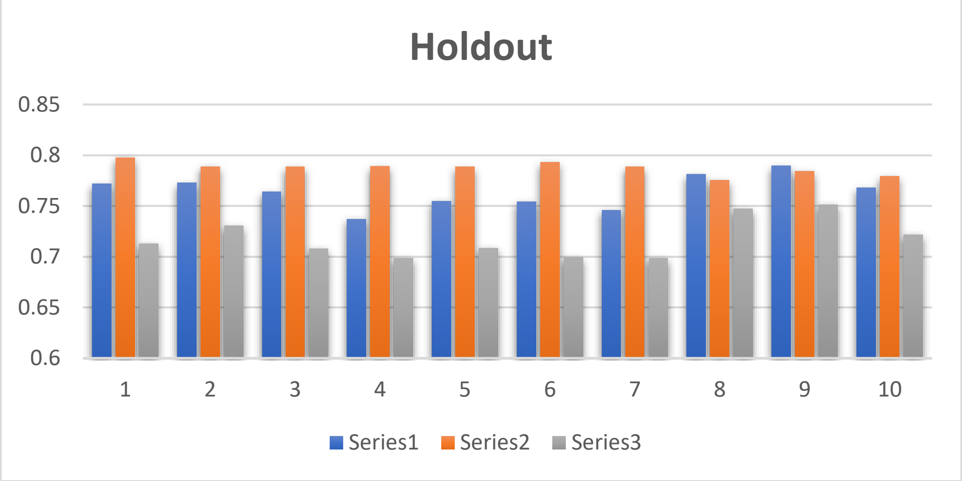


Figure B-2 Holdout results obtained using all classifiers (T_1)

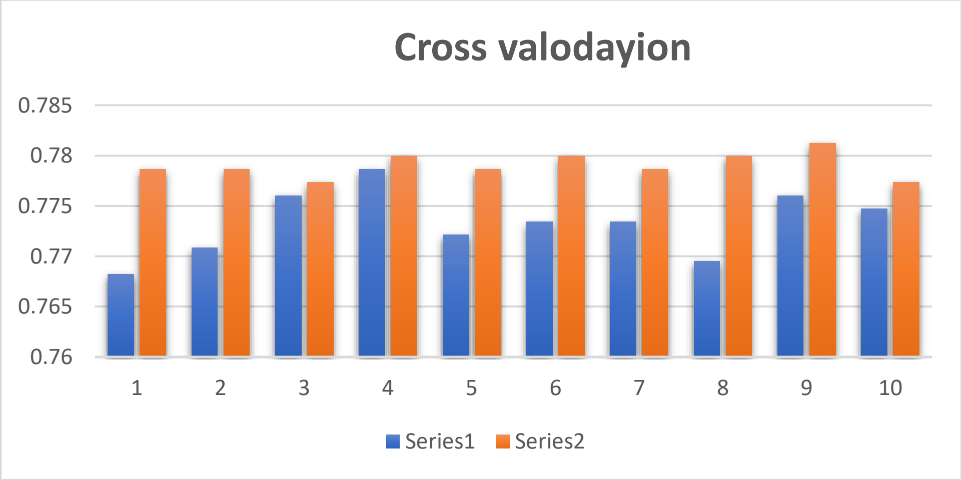


Figure B-3 Comparison between original and obtained accuracy (T_2)

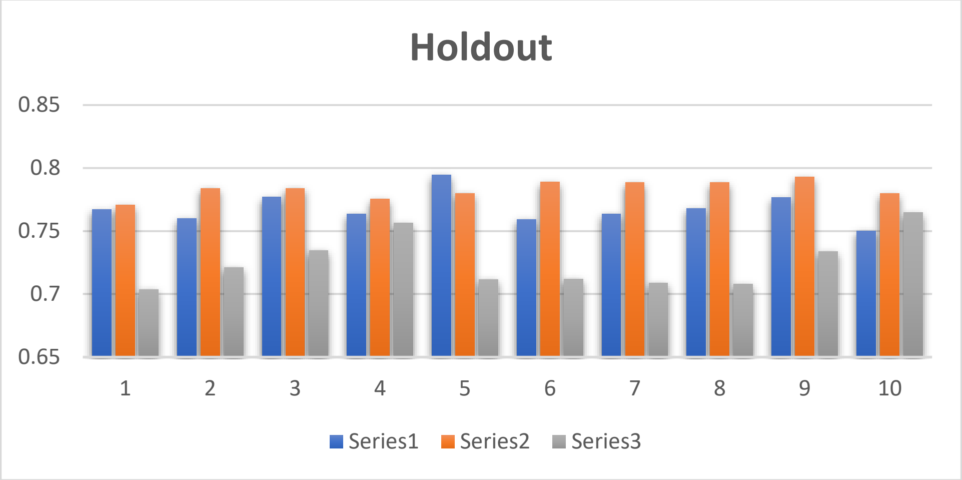


Figure B-43 Holdout results obtained using all classifiers (T_2)

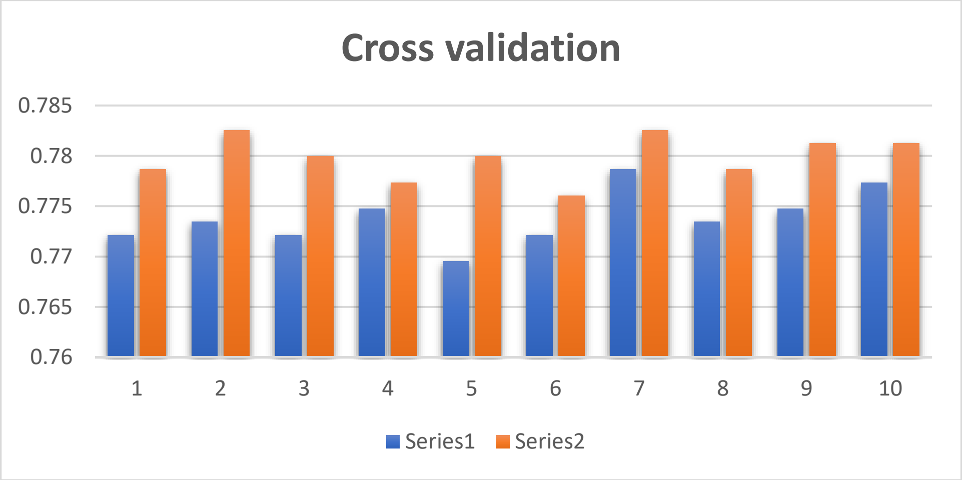


Figure B-5 Comparison between original and obtained accuracy (T_3)

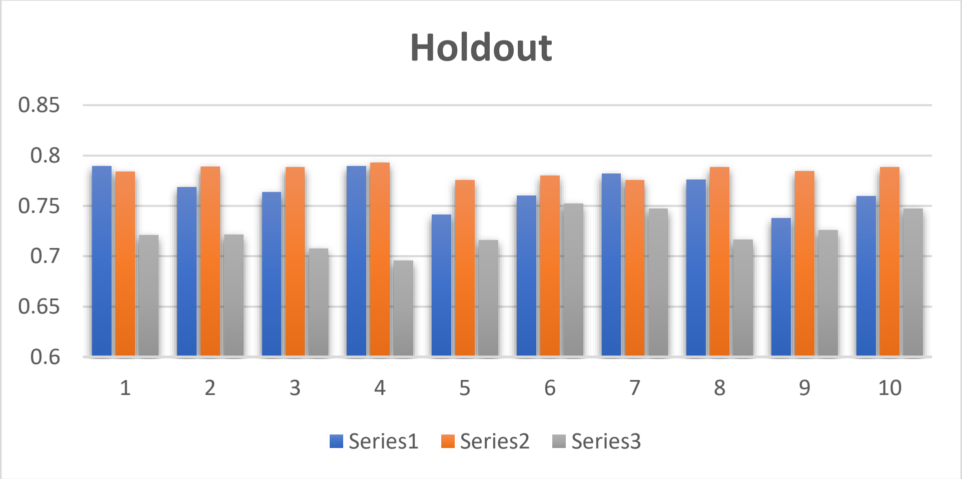


Figure B-6 Holdout results obtained using all classifiers (T_3)

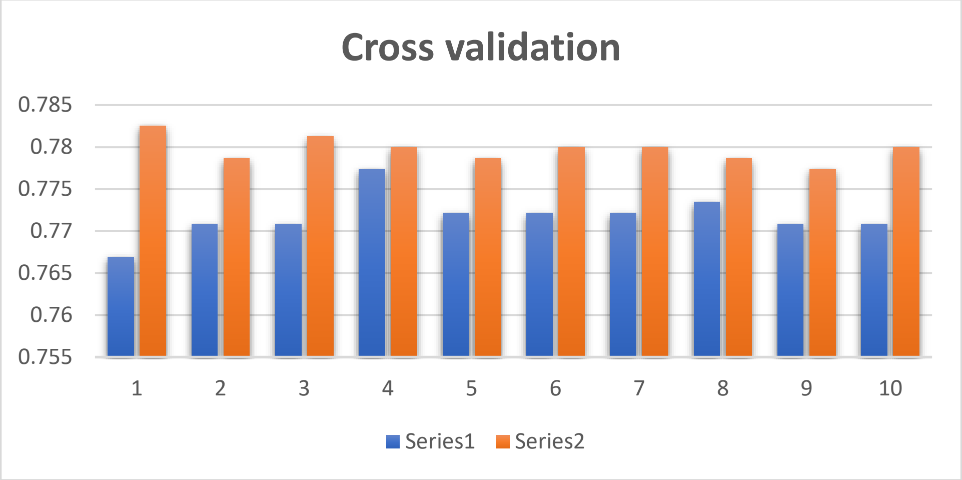


Figure B-7 Comparison between original and obtained accuracy (T_4)

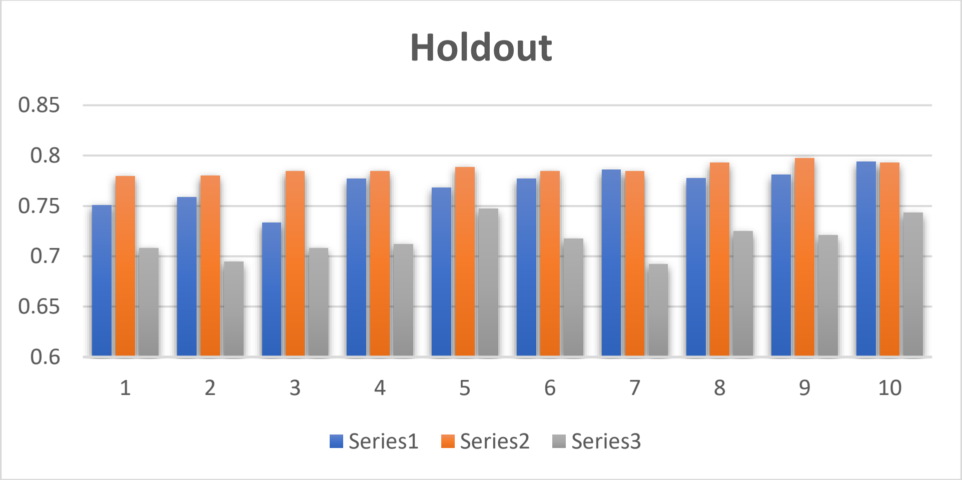


Figure B-8 Holdout results obtained using all classifiers (T_4)

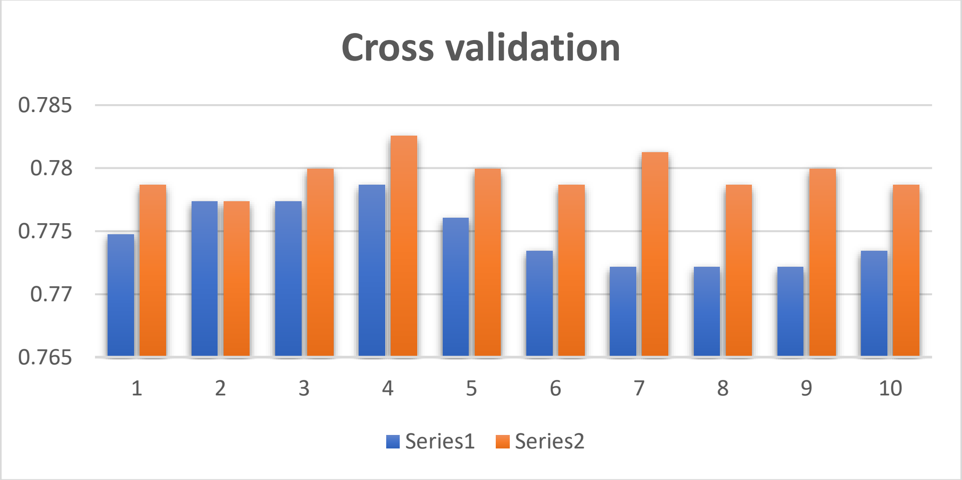


Figure B-9 Comparison between original and obtained accuracy (T_5)

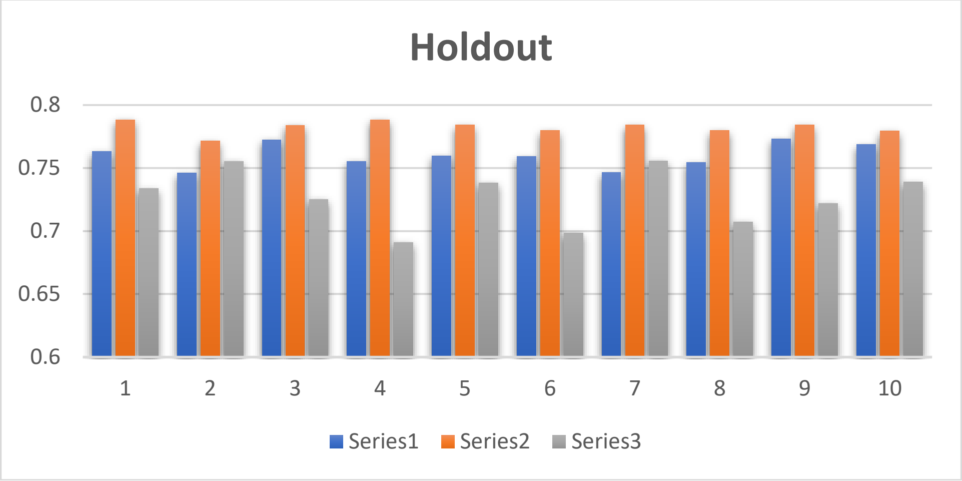


Figure B-10 Holdout results obtained using all classifiers (T_5)

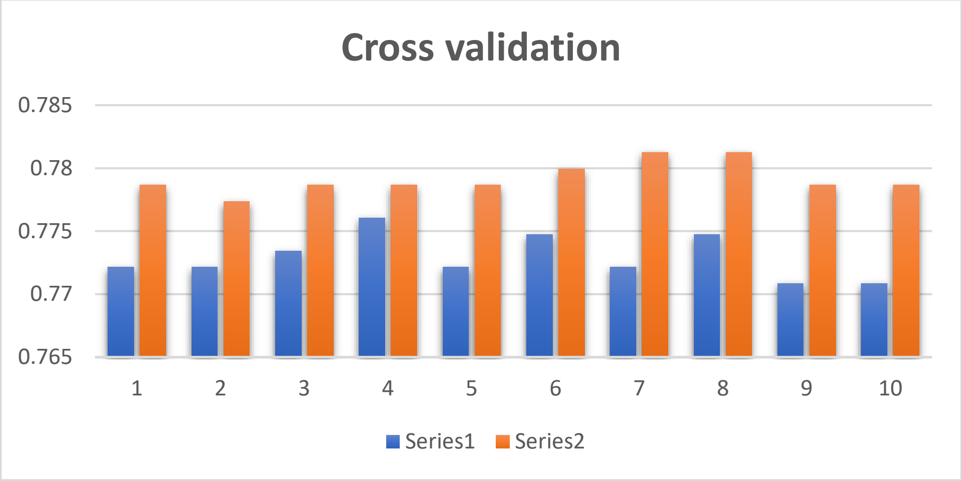


Figure B-11 Comparison between original and obtained accuracy (T_6)

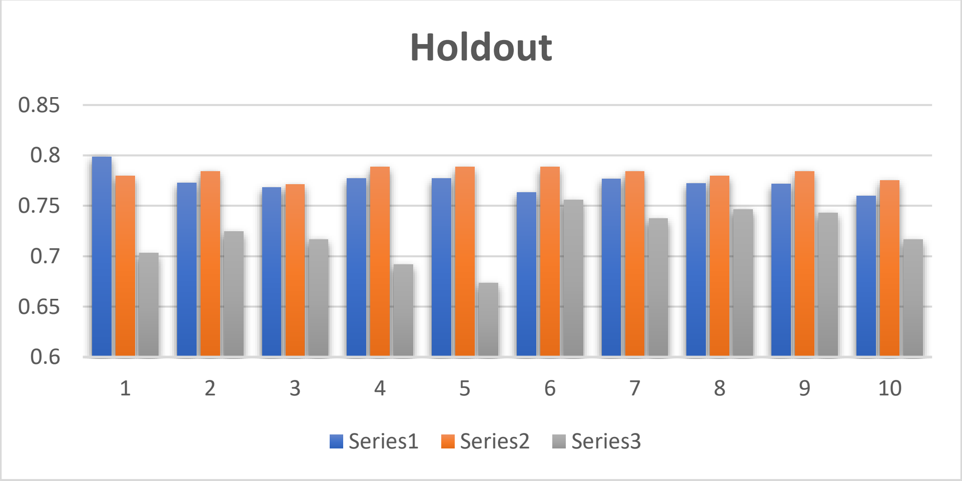


Figure B-12 Holdout results obtained using all classifiers (T_6)

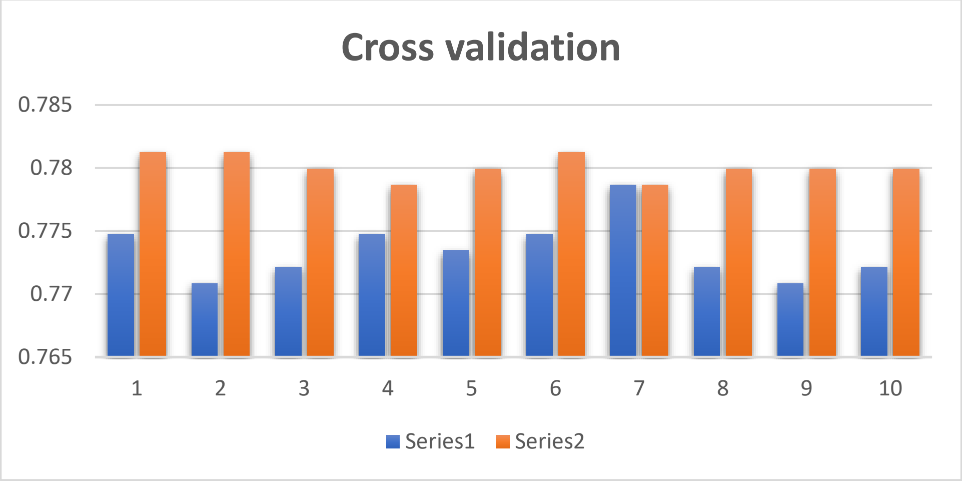


Figure MaxItr Comparison between original and obtained accuracy (T_7)

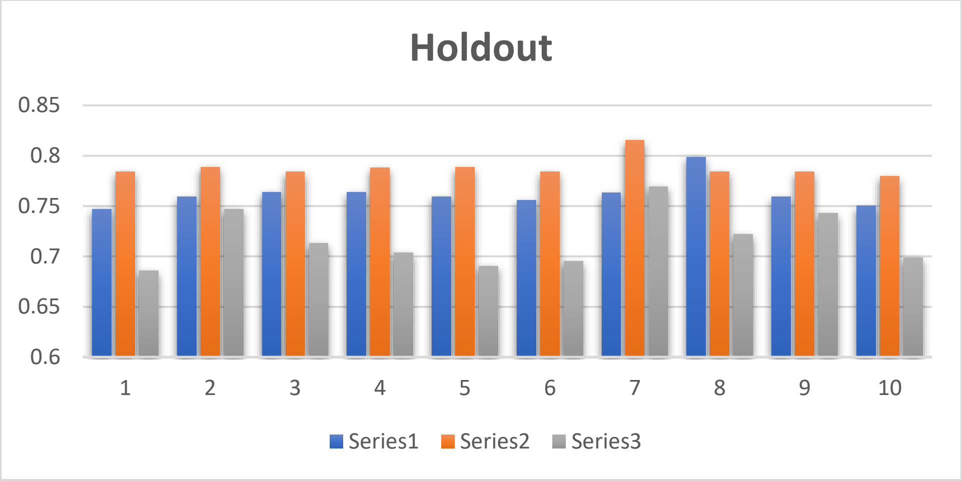


Figure B-14 Holdout results obtained using all classifiers (T_7)

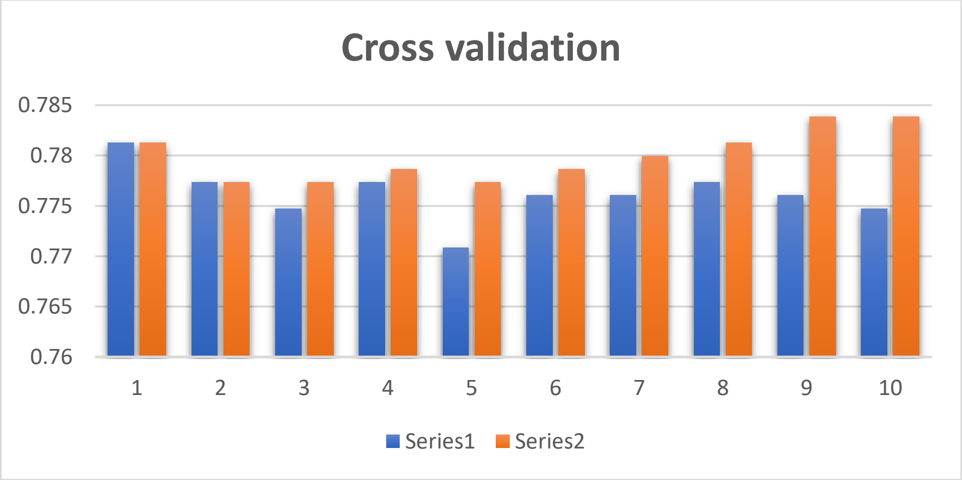


Figure B-15 Comparison between original and obtained accuracy (T_8)

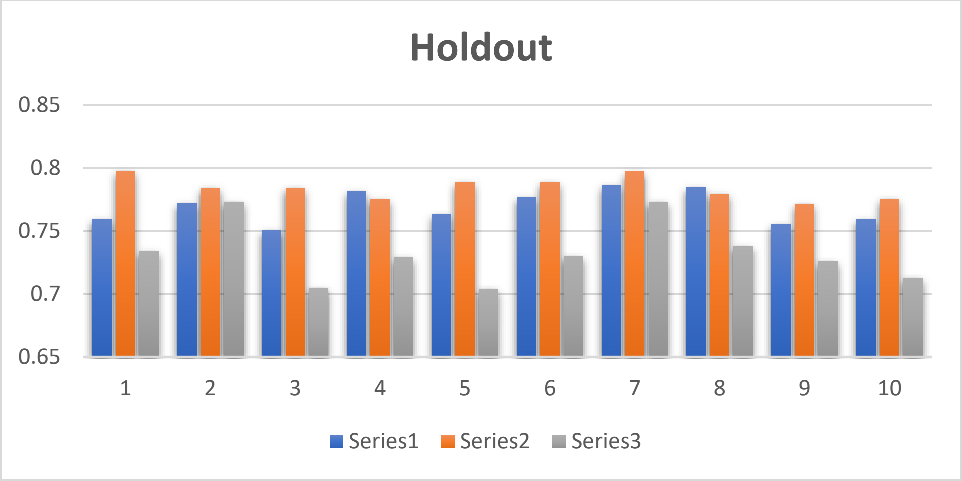


Figure B-16 Holdout results obtained using all classifiers (T_8)

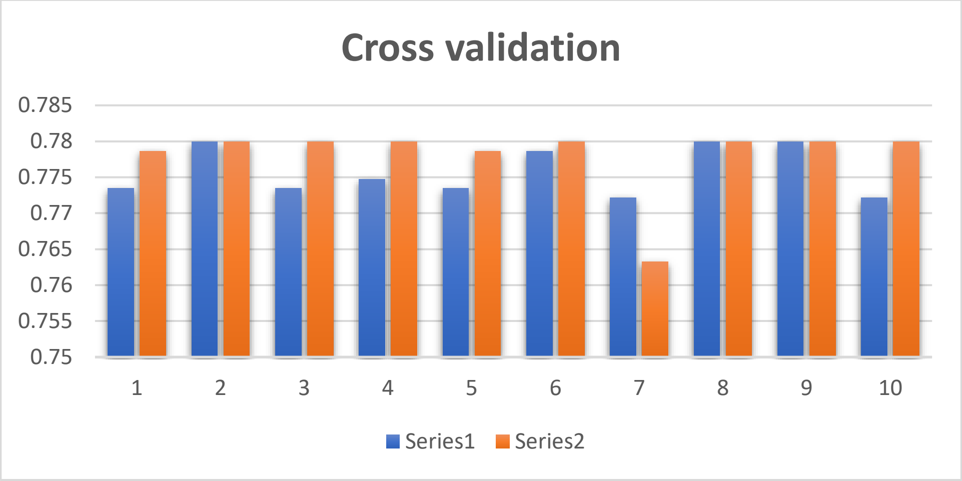


Figure B-17 Comparison between original and obtained accuracy (T_9)

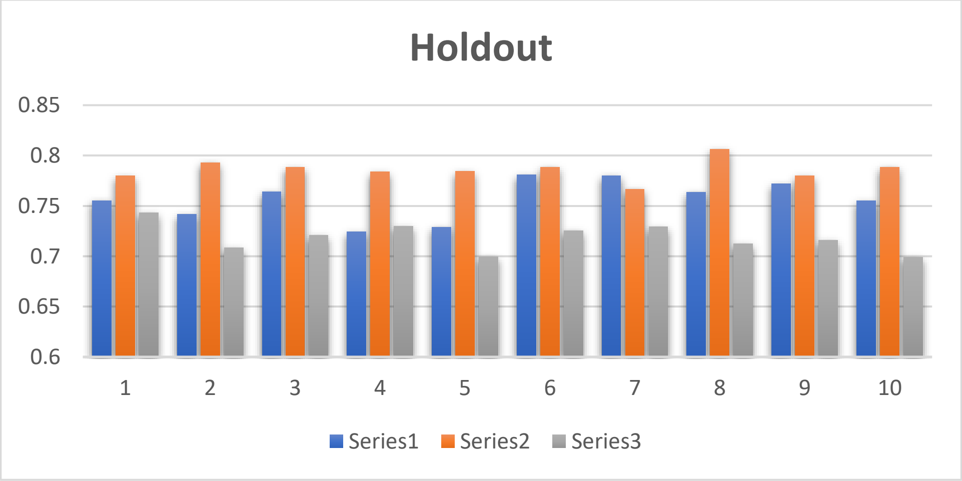


Figure B-18 Holdout results obtained using all classifiers (T_9)

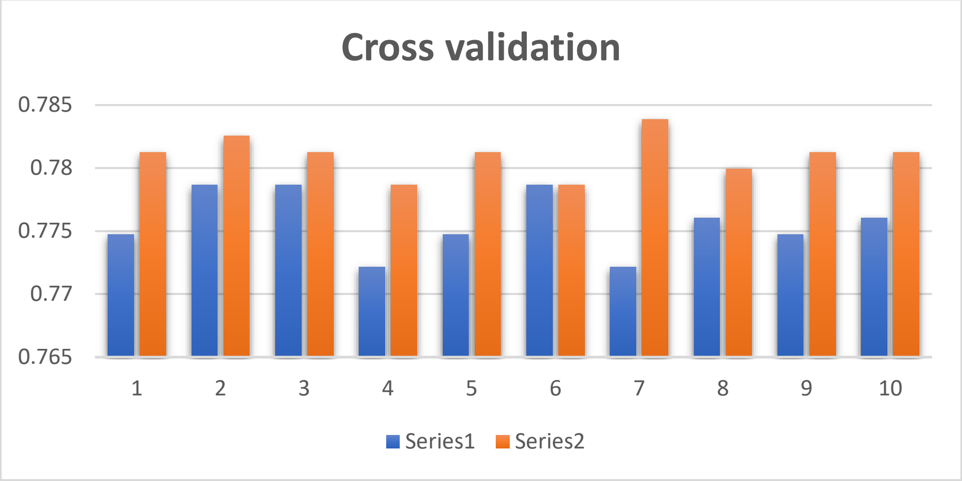


Figure B-19 Comparison between original and obtained accuracy (T_{10})

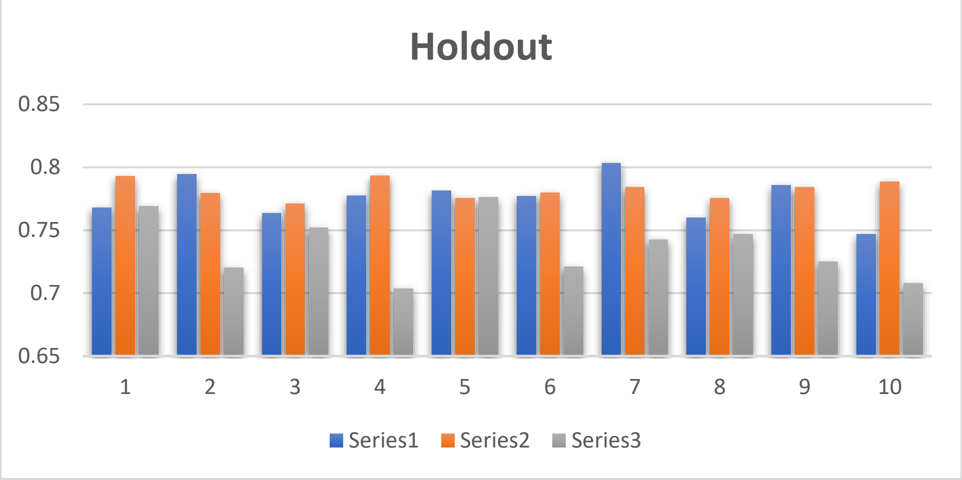


Figure B-20 Holdout results obtained using all classifiers (T_{10})

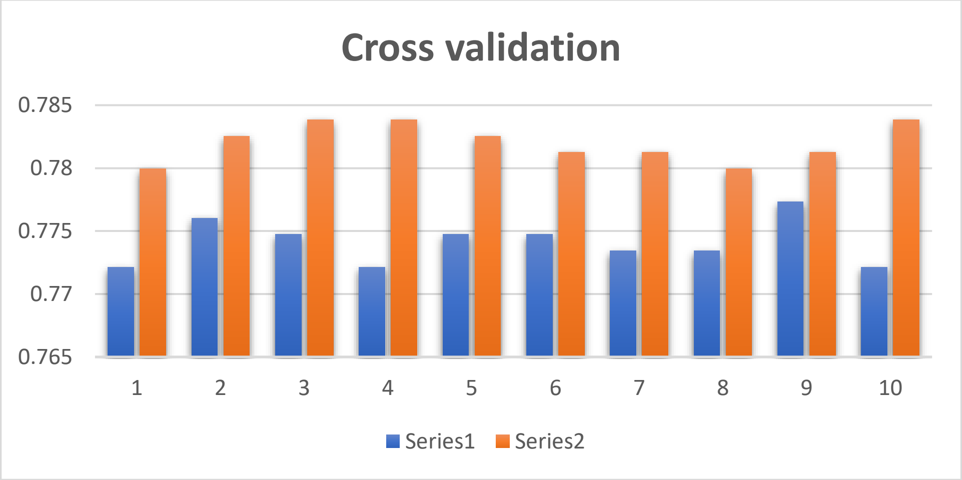


Figure B-21 Comparison between original and obtained accuracy (T_{11})

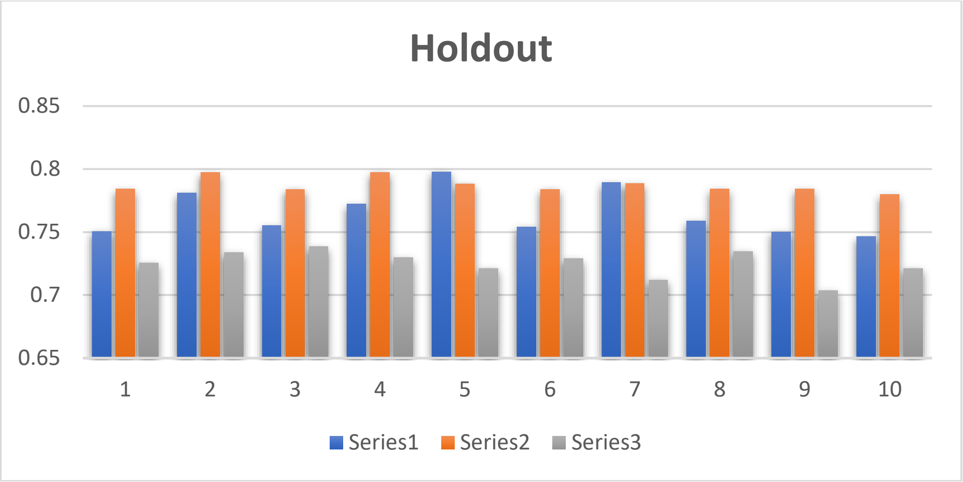


Figure B-22 Holdout results obtained using all classifiers (T_{11})

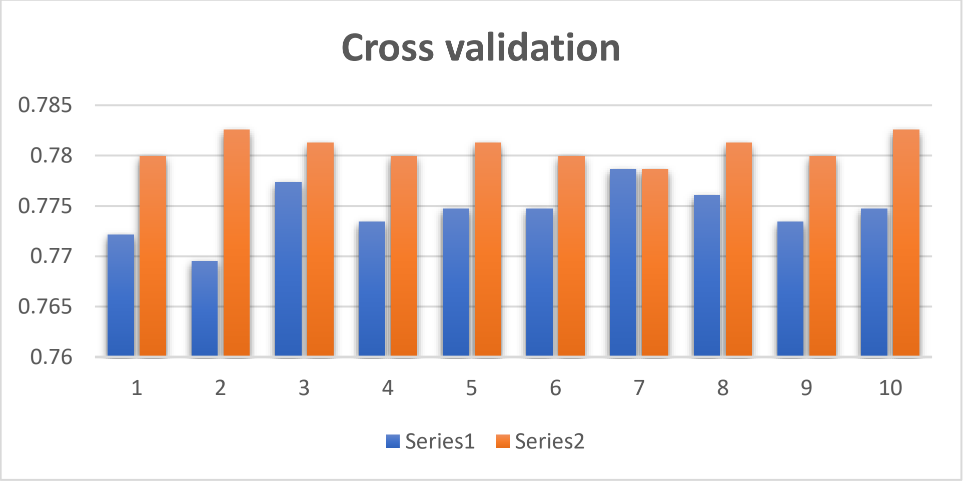


Figure B-23 Comparison between original and obtained accuracy (T_{12})

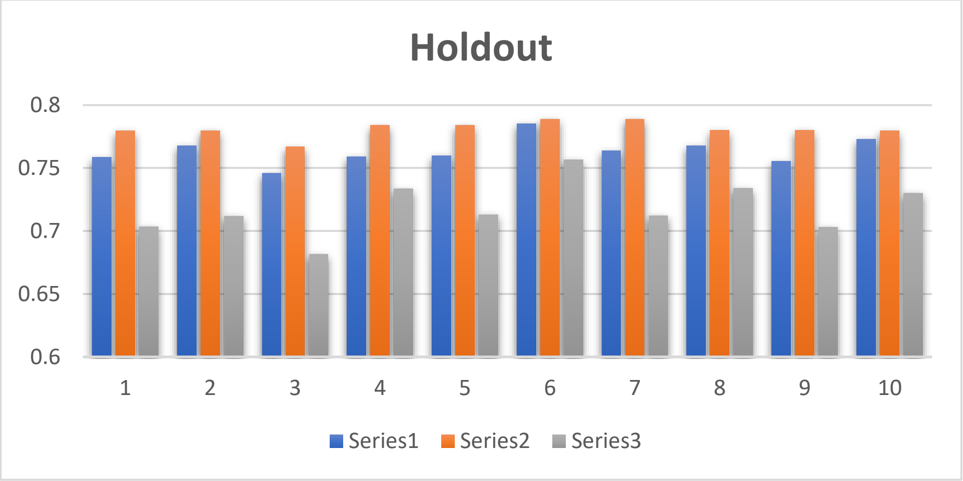


Figure B-24 Holdout results obtained using all classifiers (T_{12})

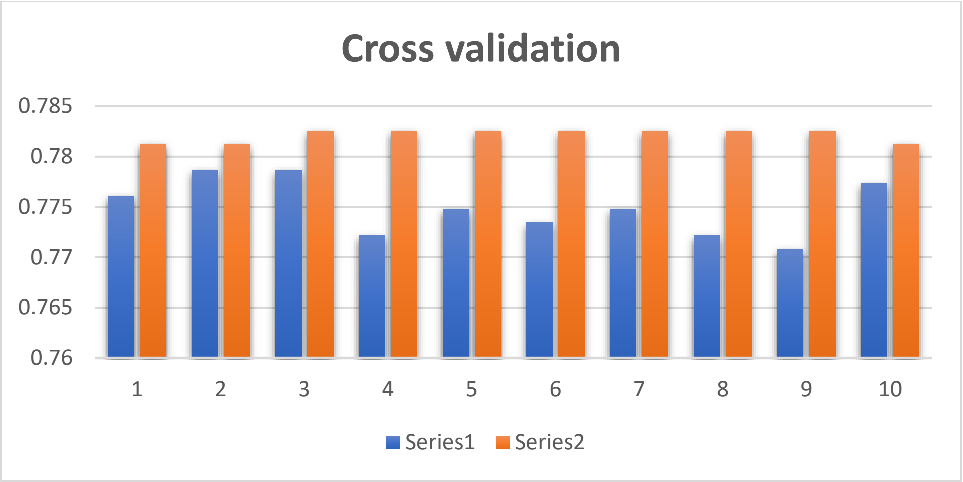


Figure B-25 Comparison between original and obtained accuracy (T_{13})

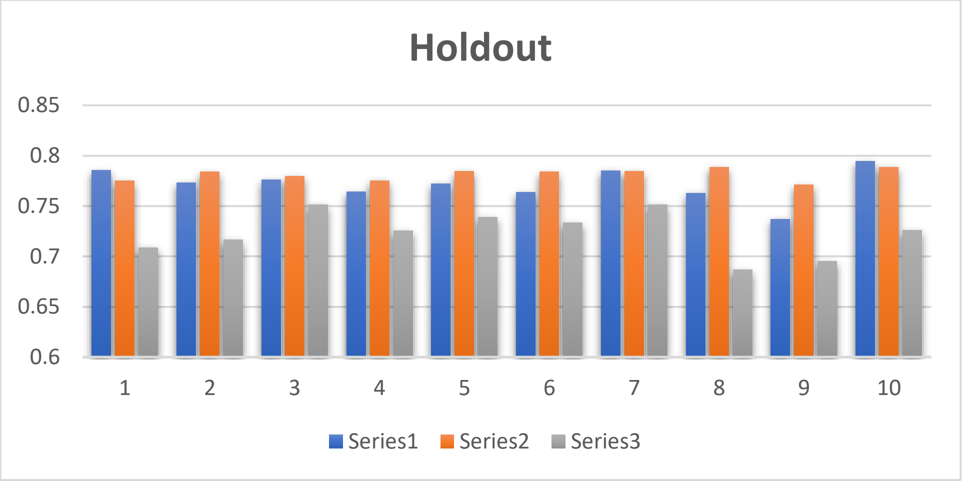


Figure B-26 Holdout results obtained using all classifiers (T_{13})

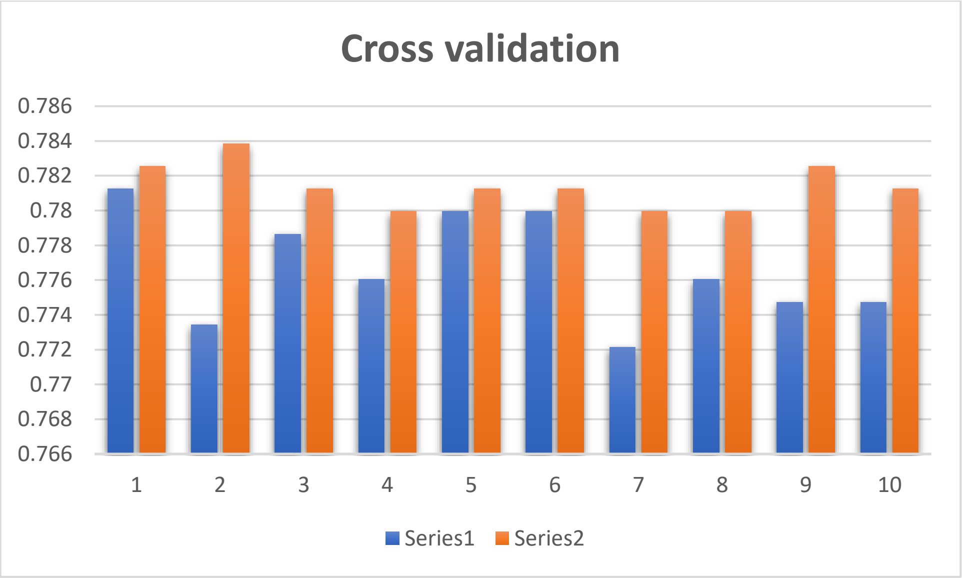


Figure B-27 Comparison between original and obtained accuracy (T_{14})

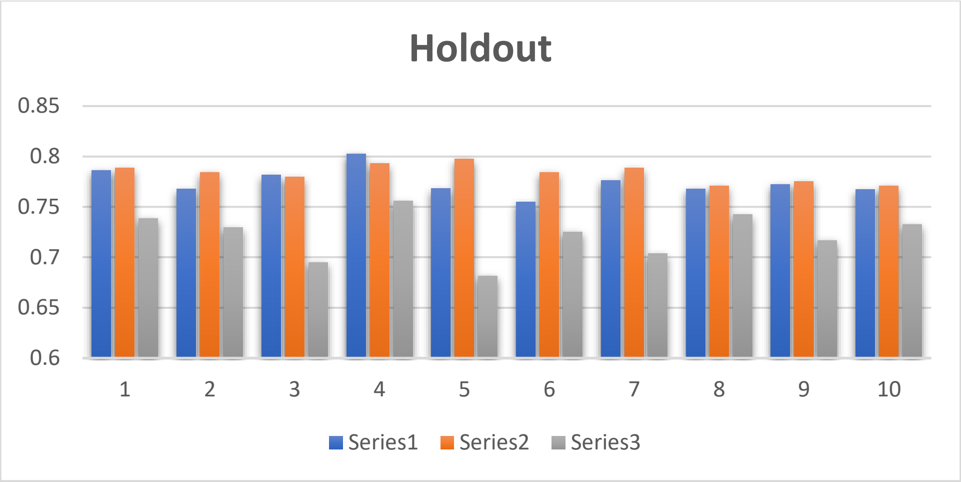


Figure B-28 Holdout results obtained using all classifiers (T_{14})

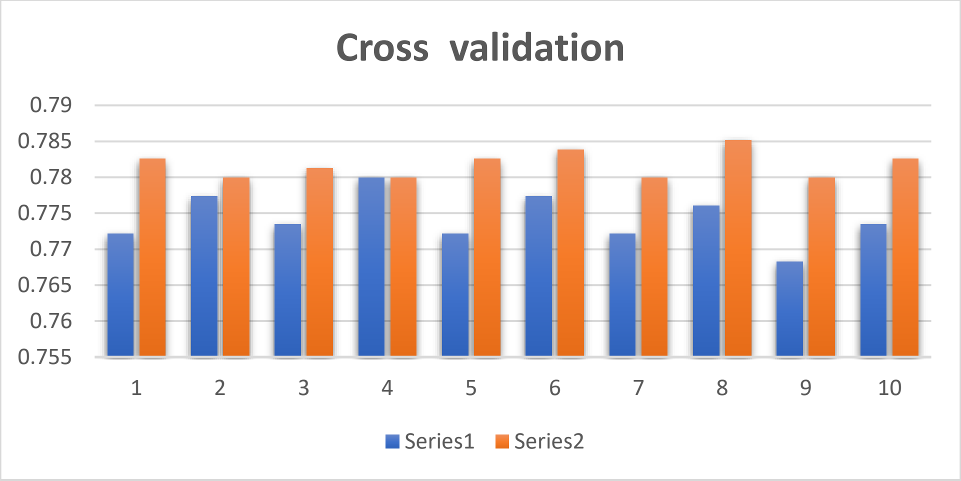


Figure B-29 Comparison between original and obtained accuracy (T_{15})

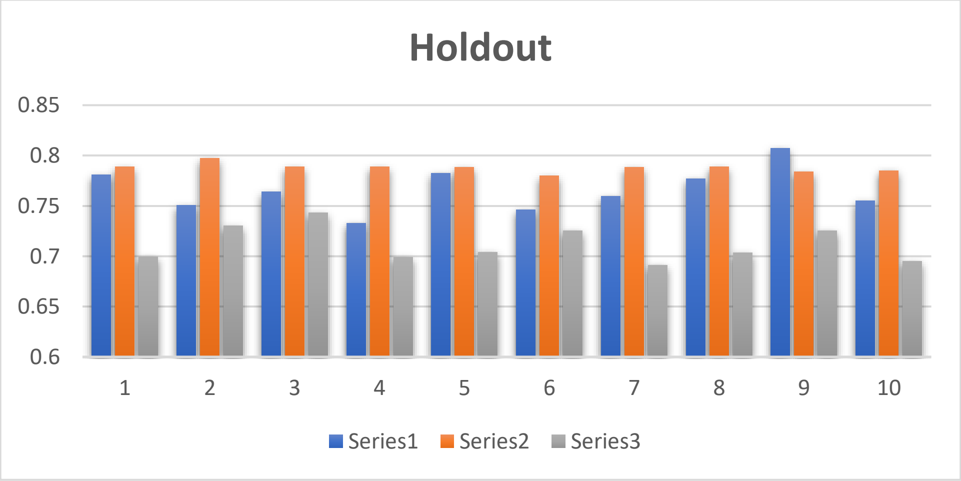


Figure B-30 Holdout results obtained using all classifiers (T_{15})

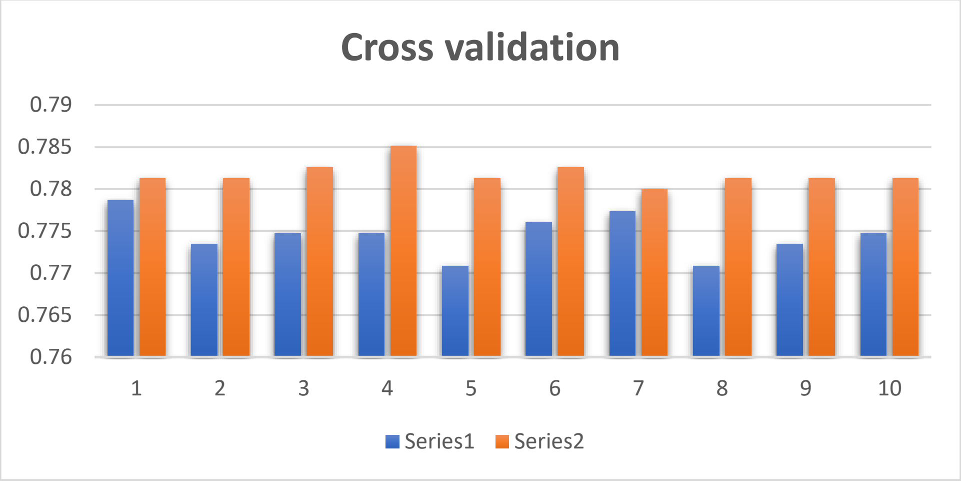


Figure B-31 Comparison between original and obtained accuracy (T_{16})

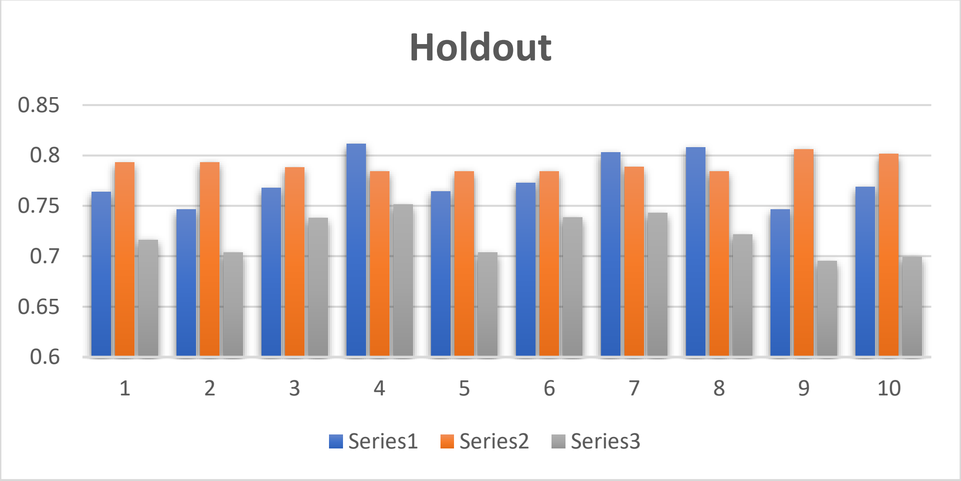


Figure B-32 Holdout results obtained using all classifiers (T_{16})

[redacted]APPENDIX C: A RESULTS OF NBC

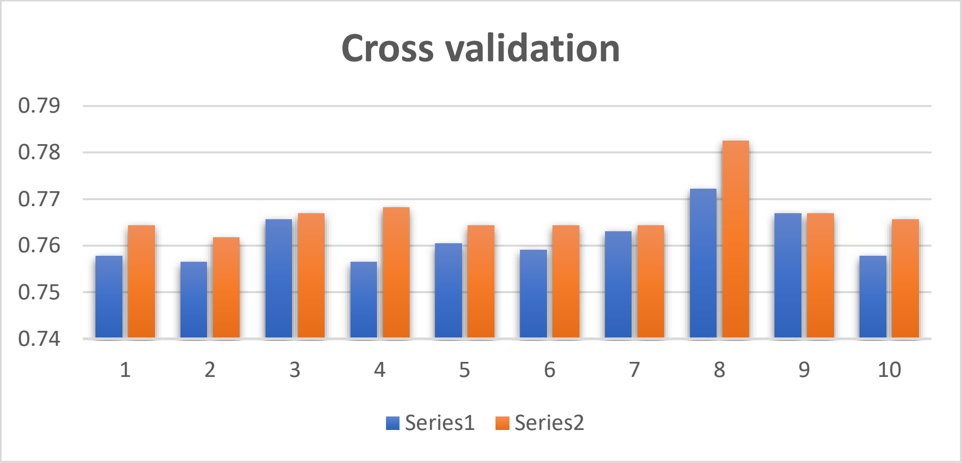


Figure C-1 Comparison between original and obtained accuracy (T_1)

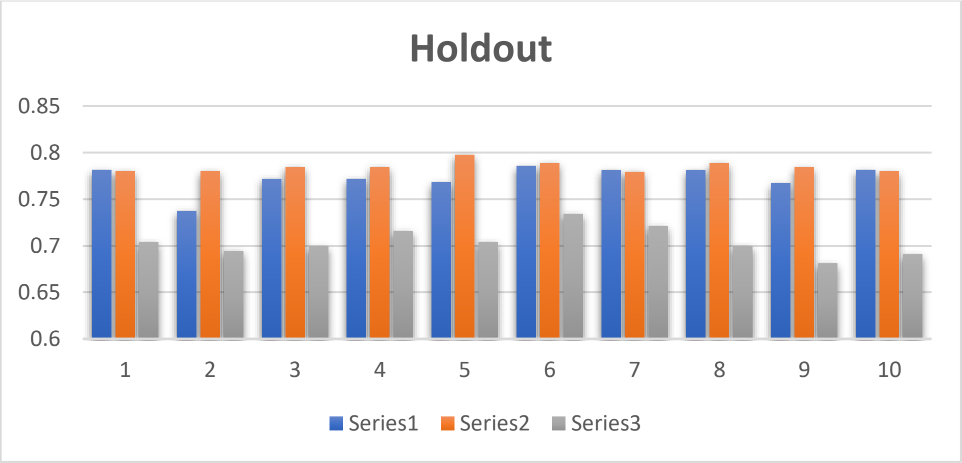


Figure C-2 Holdout results obtained using all classifiers (T_1)

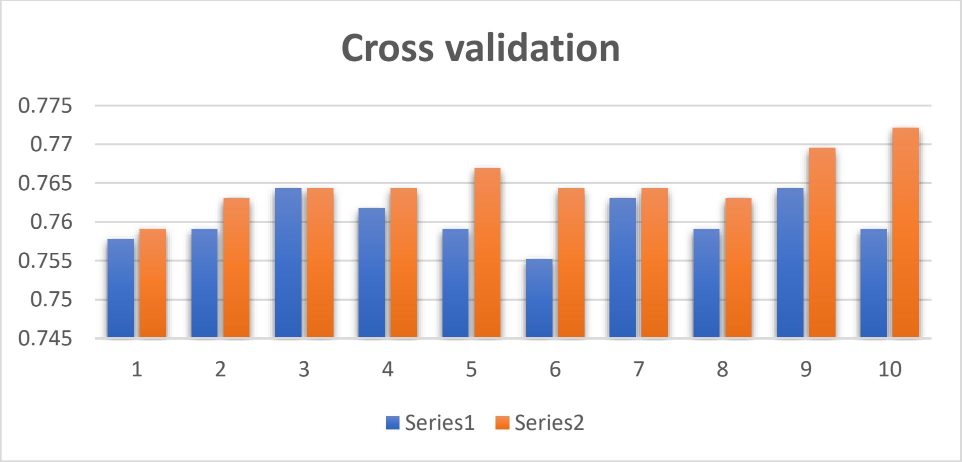


Figure C-3 Comparison between original and obtained accuracy (T_2)

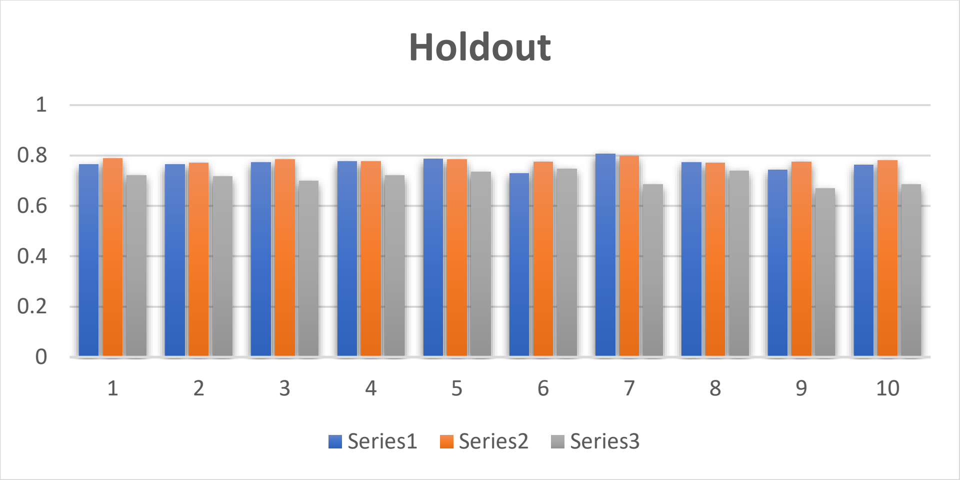


Figure C-4 Holdout results obtained using all classifiers (T_2)

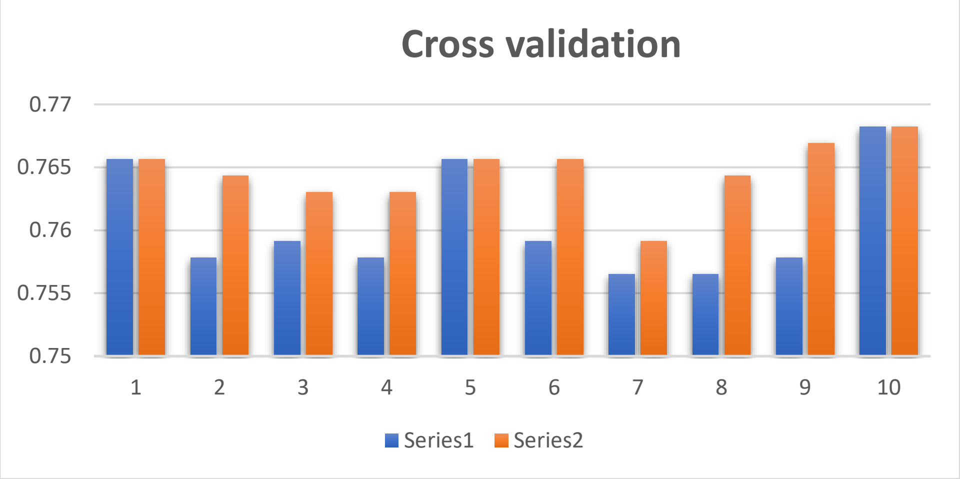


Figure C-5 Comparison between original and obtained accuracy (T_3)

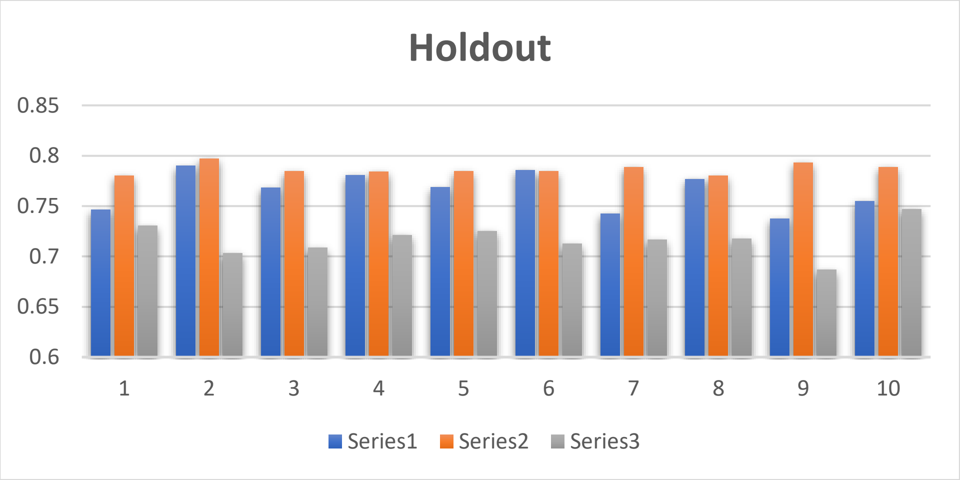


Figure C-6 Holdout results obtained using all classifiers (T_3)

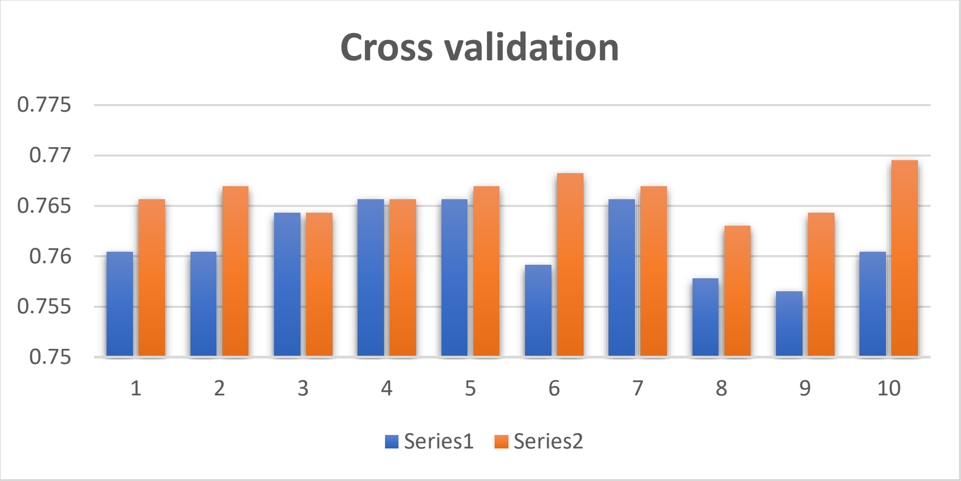


Figure C-7 Comparison between original and obtained accuracy (T_4)

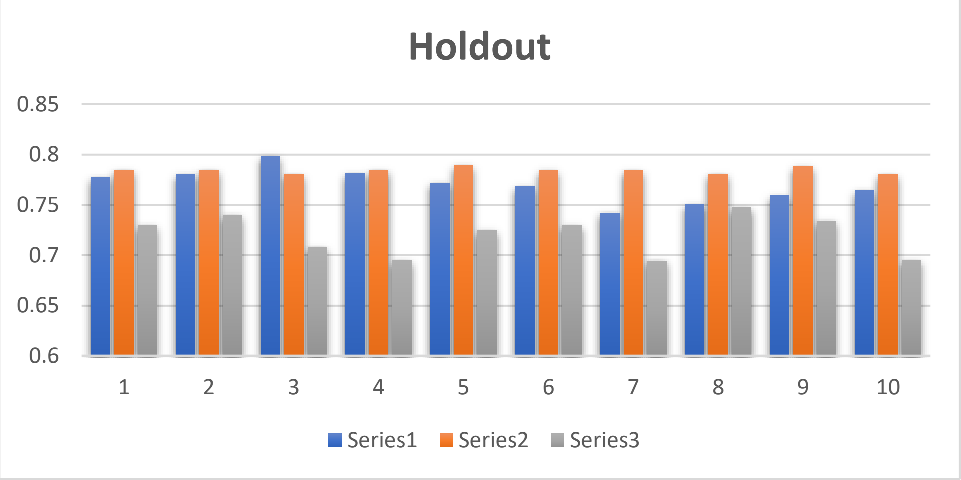


Figure C-8 Holdout results obtained using all classifiers (T_4)

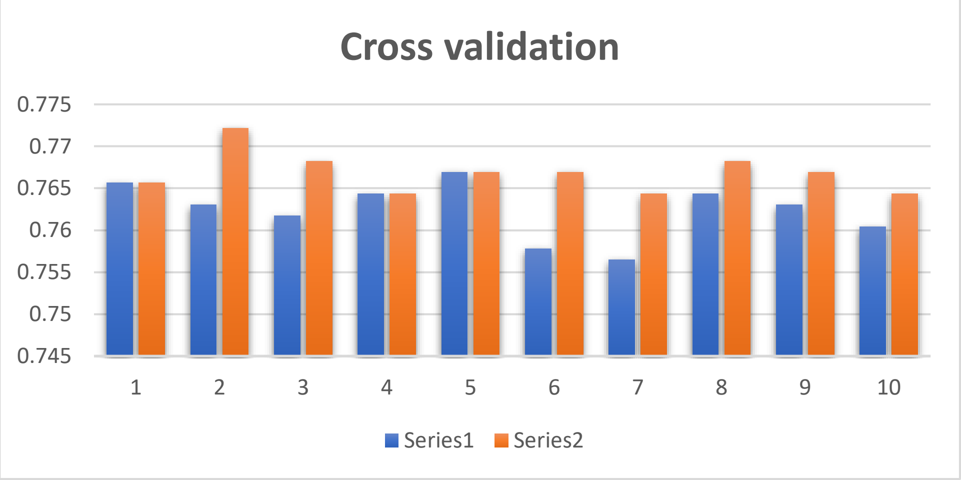


Figure C-9 Comparison between original and obtained accuracy (T_5)

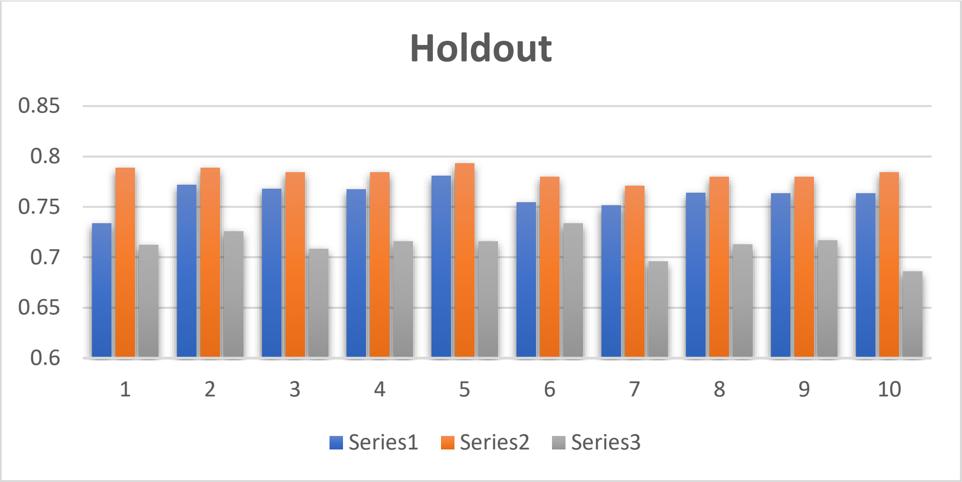


Figure C-10 Holdout results obtained using all classifiers (T_5)

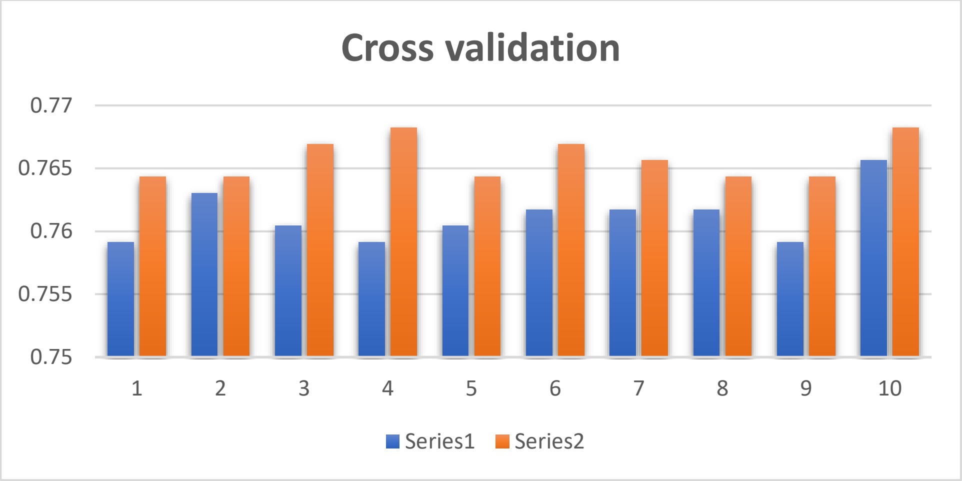


Figure C-11 Comparison between original and obtained accuracy (T_6)

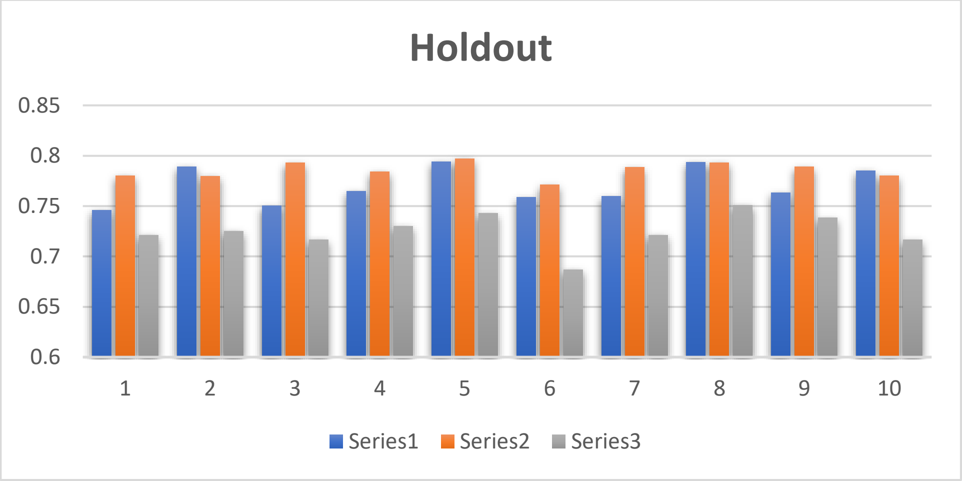


Figure C-12 Holdout results obtained using all classifiers (T_6)

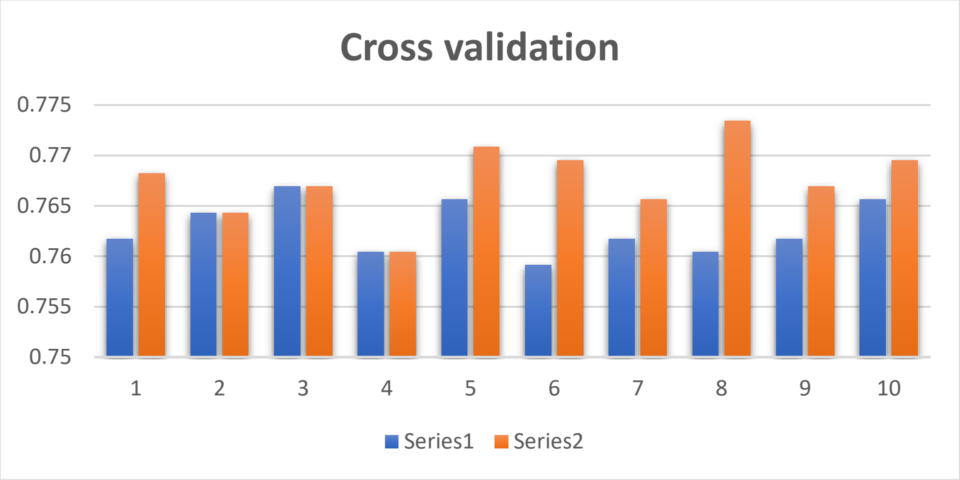


Figure C-13 Comparison between original and obtained accuracy (T_7)

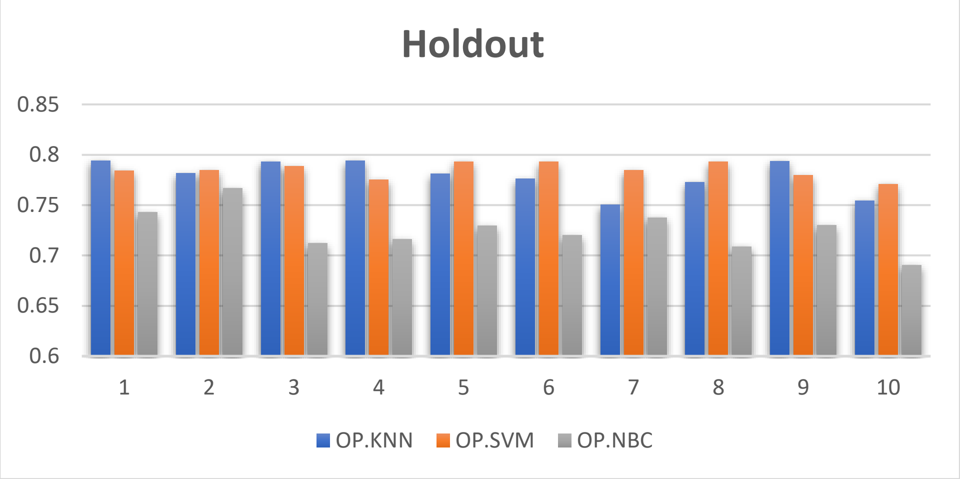


Figure C-14 Holdout results obtained using all classifiers (T_7)

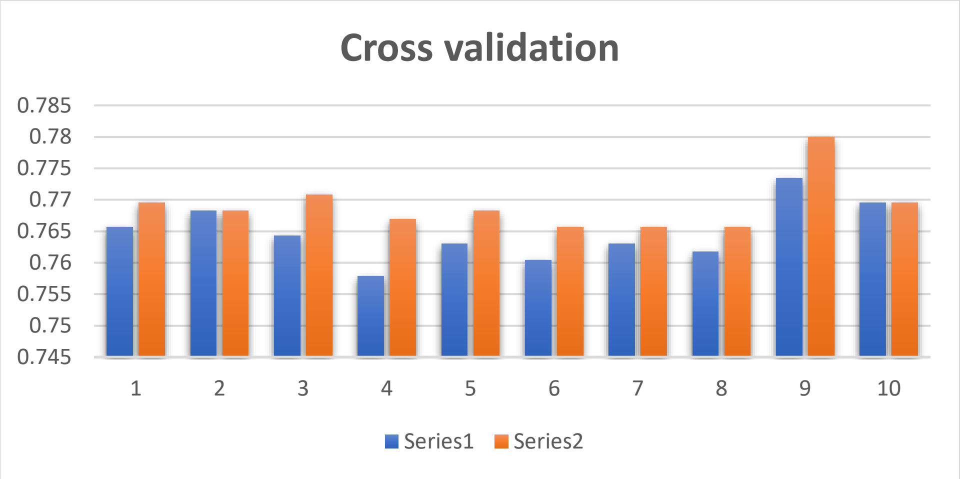


Figure C-15 Comparison between original and obtained accuracy (T_8)

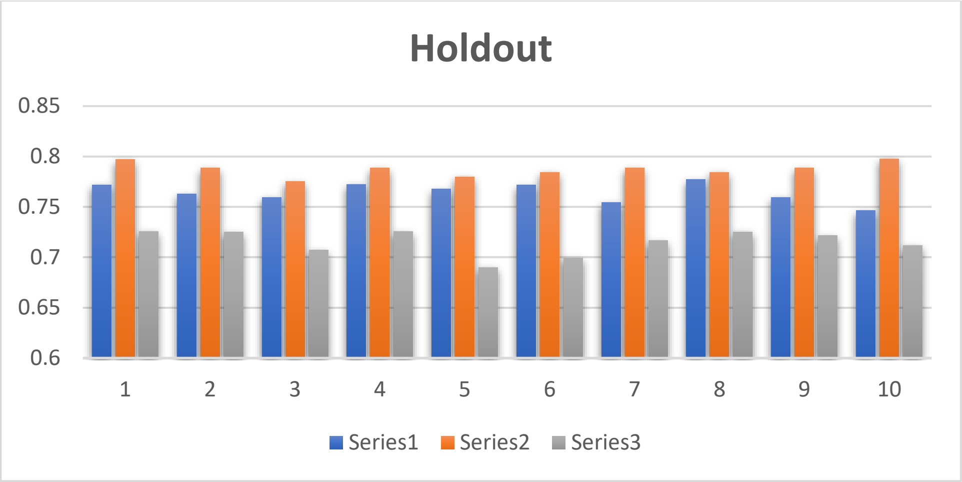


Figure C-16 Holdout results obtained using all classifiers (T_8)

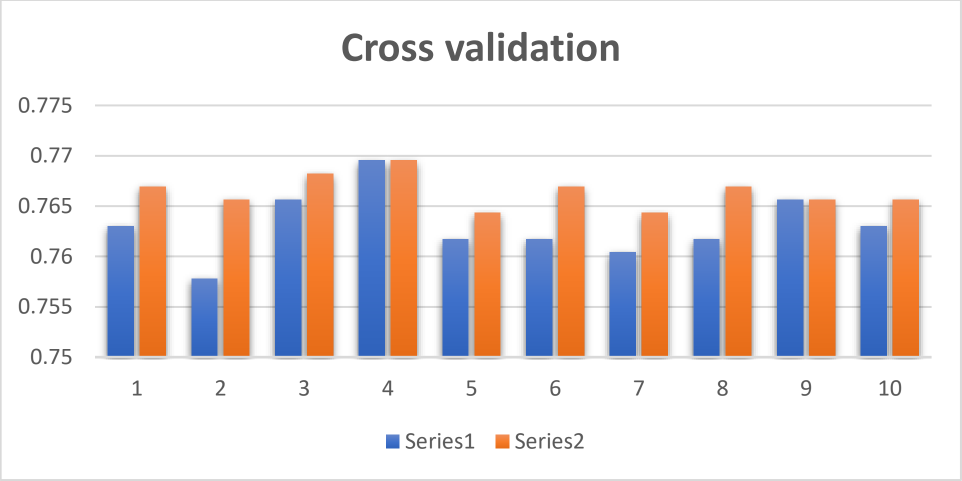


Figure C-17 Comparison between original and obtained accuracy (T_9)

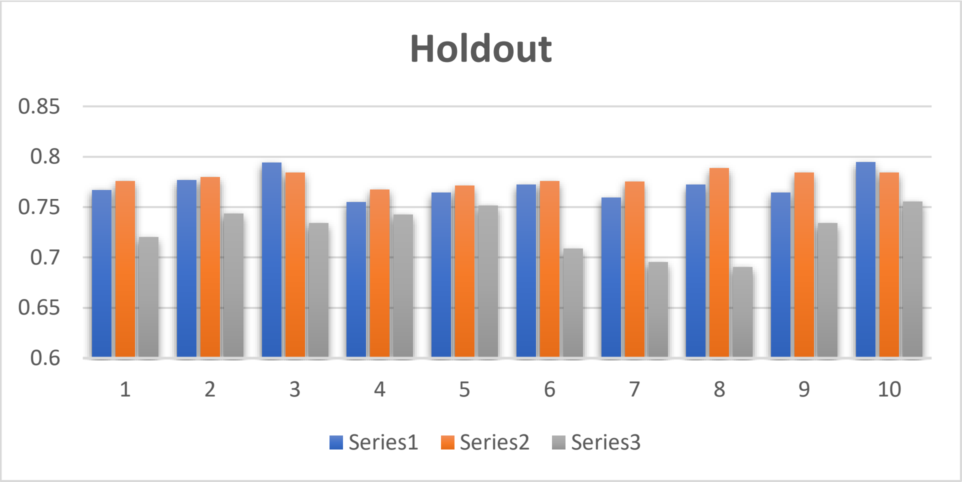


Figure C-18 Holdout results obtained using all classifiers (T_9)

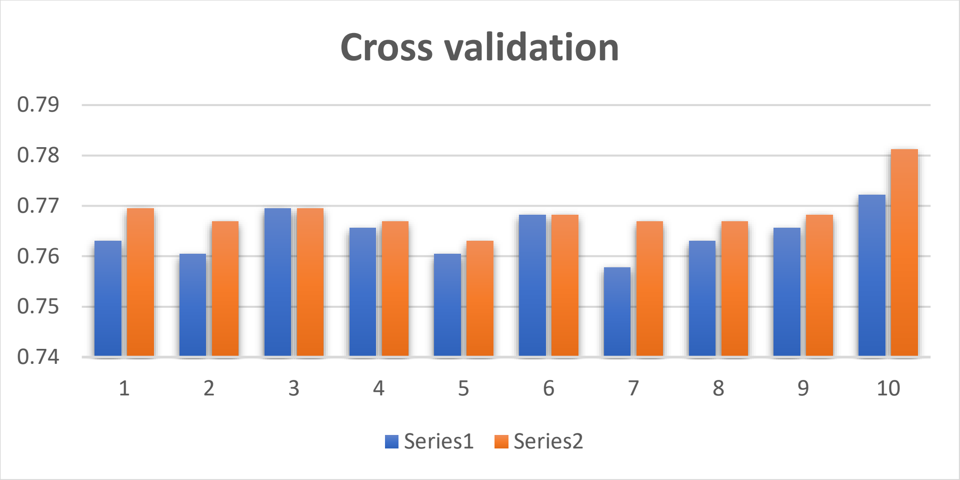


Figure C-19 Comparison between original and obtained accuracy (T_{10})

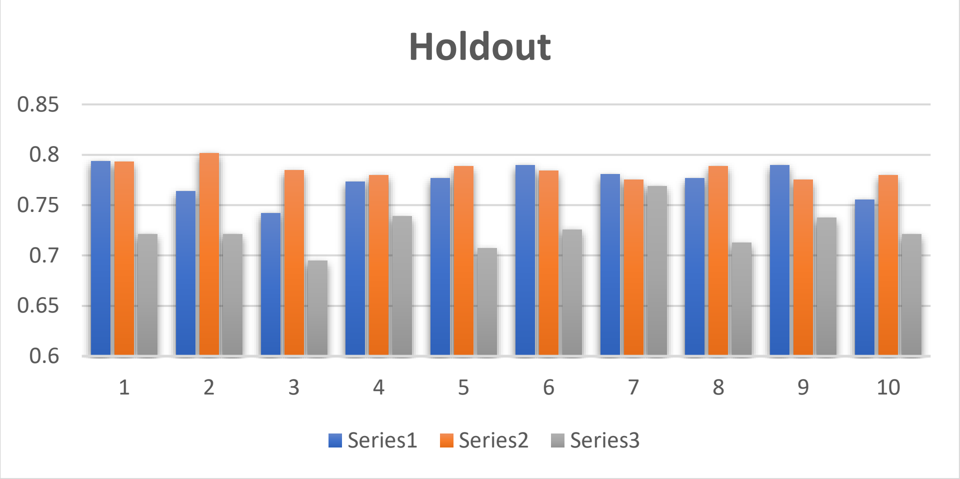


Figure C-20 Holdout results obtained using all classifiers (T_{10})

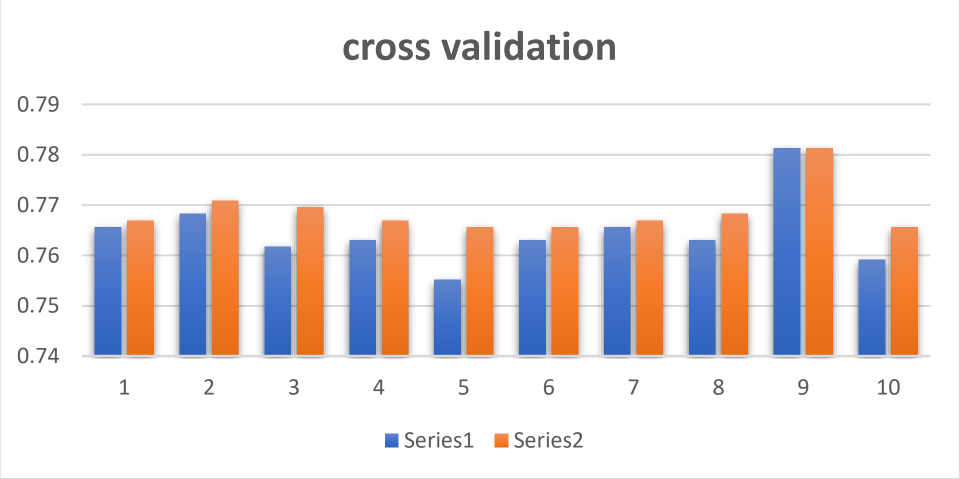


Figure C-21 Comparison between original and obtained accuracy (T_{11})

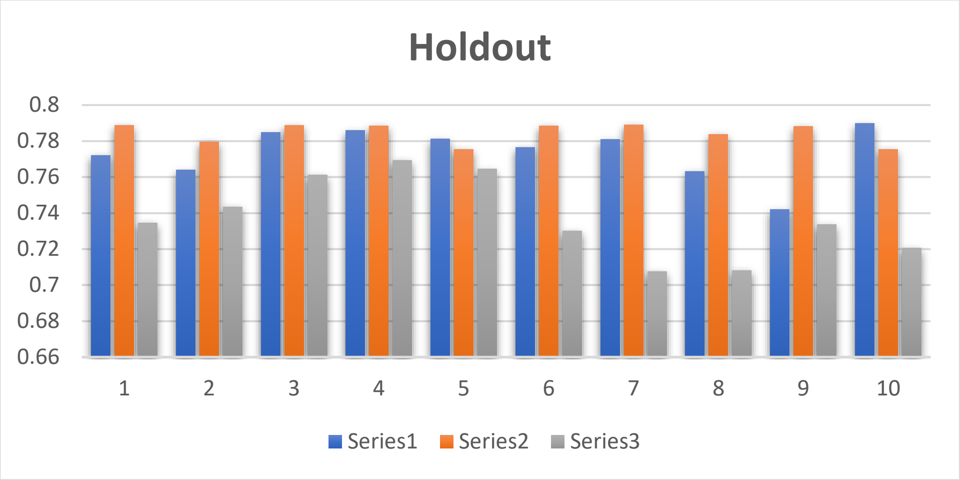


Figure C-22 Holdout results obtained using all classifiers (T_{11})

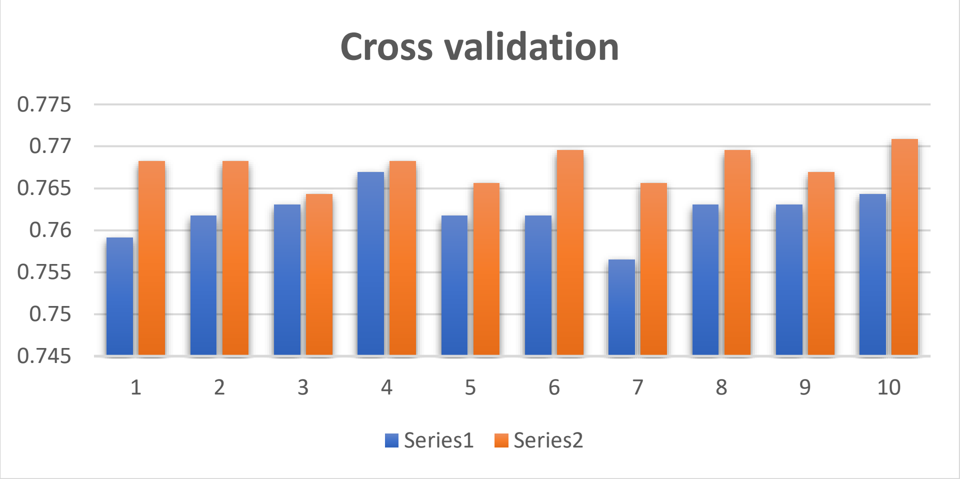


Figure C-23 Comparison between original and obtained accuracy (T_{12})

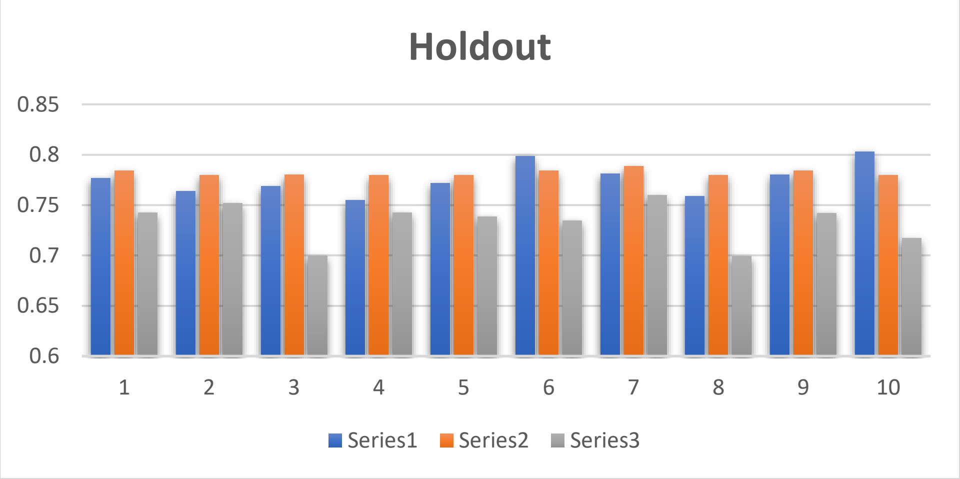


Figure C-24 Holdout results obtained using all classifiers (T_{12})

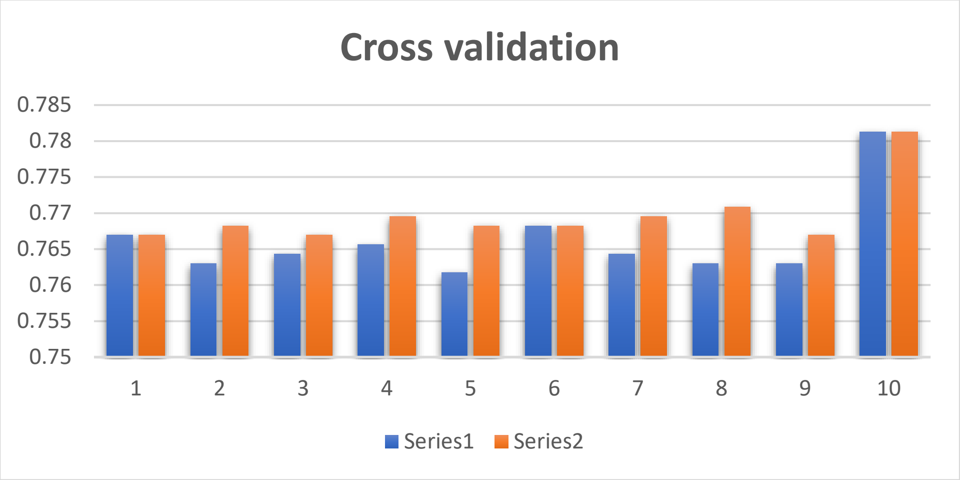


Figure C-25 Comparison between original and obtained accuracy (T_{13})

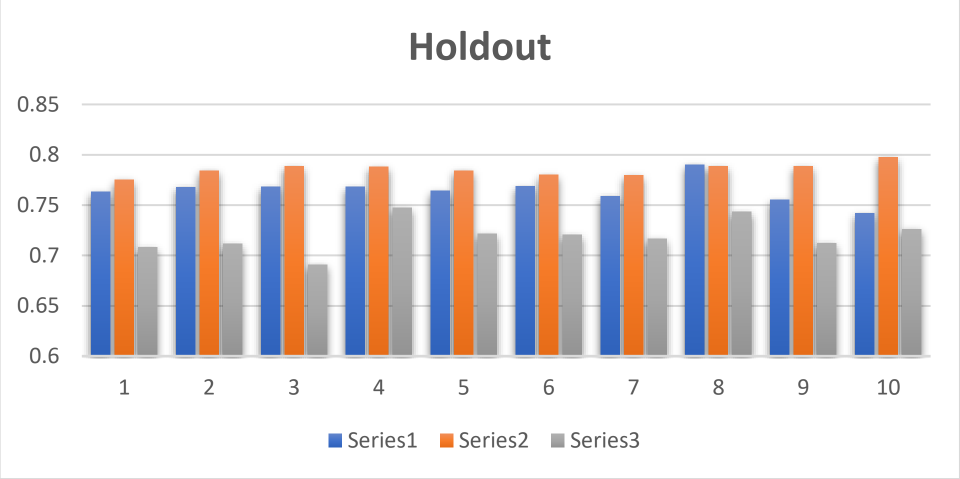


Figure C-26 Holdout results obtained using all classifiers (T_{13})

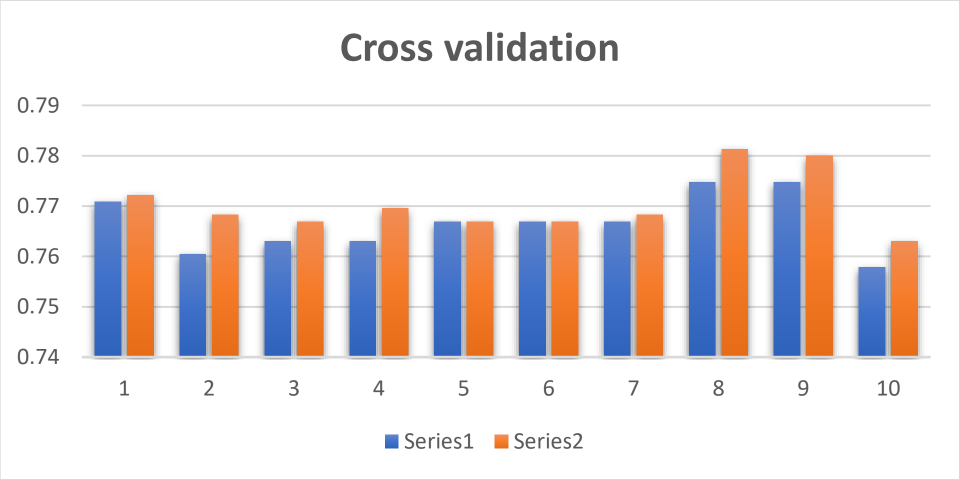


Figure C-27 Comparison between original and obtained accuracy (T_{14})

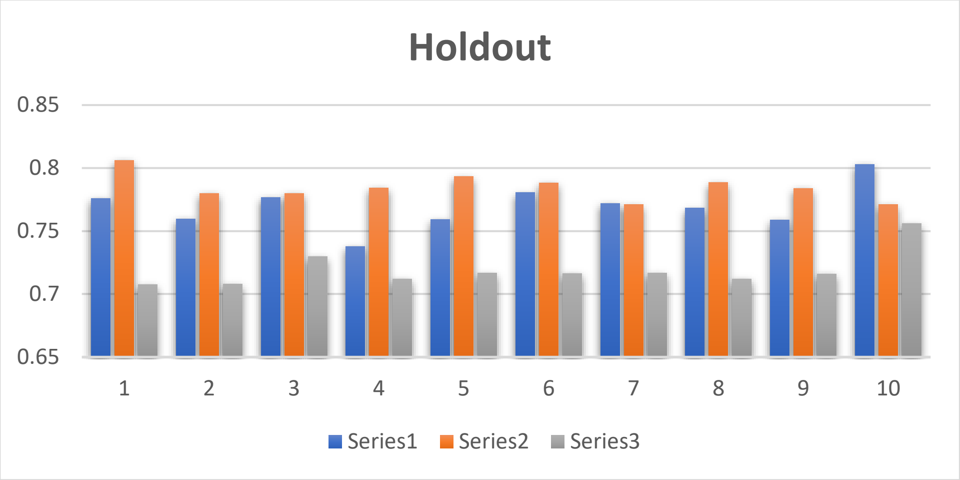


Figure C-28 Holdout results obtained using all classifiers (T_{14})

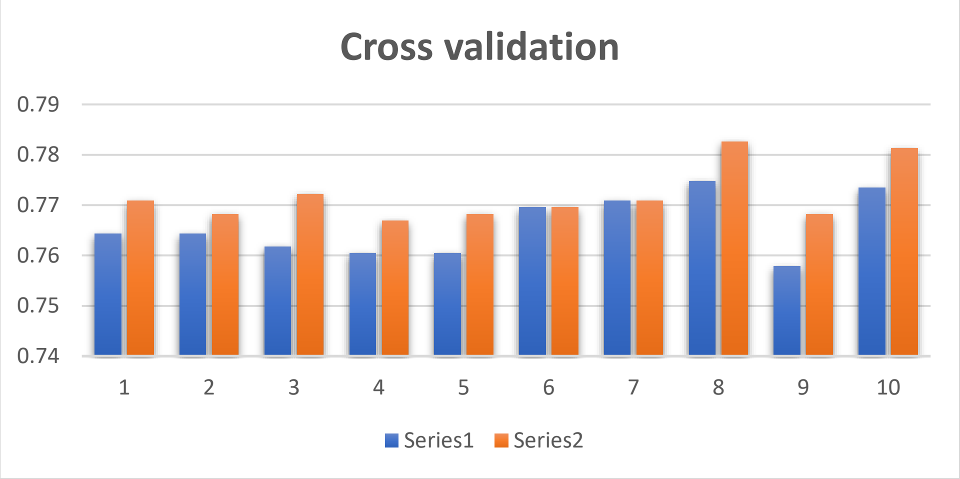


Figure C-29 Comparison between original and obtained accuracy (T_{15})

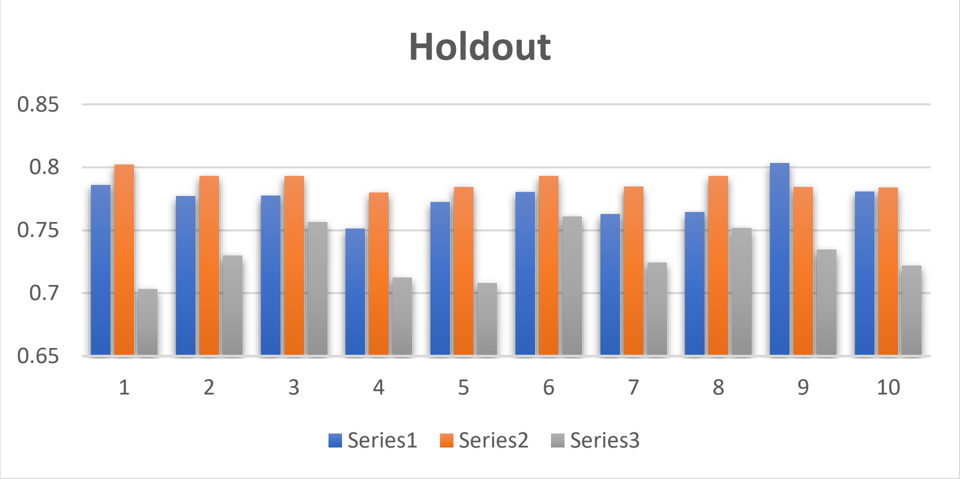


Figure C-30 Holdout results obtained using all classifiers (T_{15})

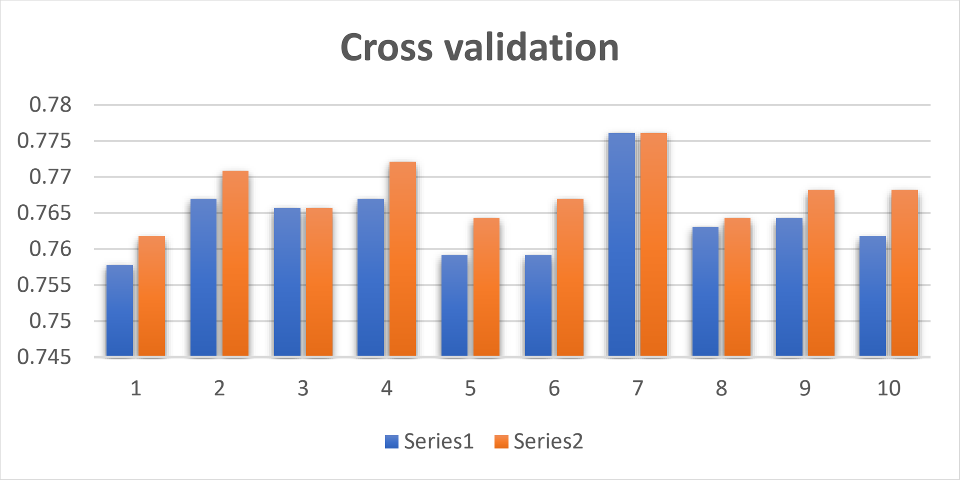


Figure C-31 Comparison between original and obtained accuracy (T_{16})

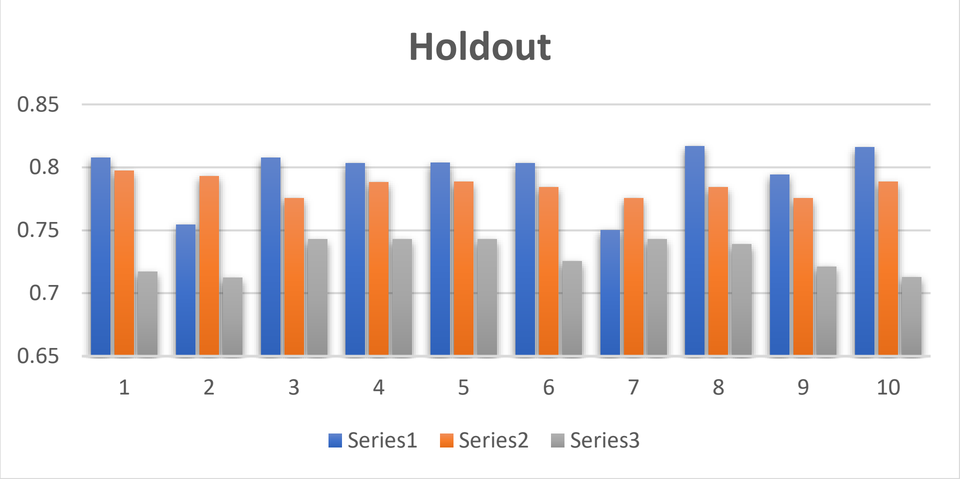


Figure C-32 Holdout results obtained using all classifiers (T_{16})