University of Bahrain

College of Science

Department of
Postgraduate Programs

A Novel Classifier Based on Genetic Algorithms and Data Importance Reformatting

Thesis submitted in partial fulfilment of the requirements for the degree of M.Sc. In Big Data Science and Analytics

Prepared by:

Aysha Khaled Alkhayyat

20082338

Supervised by:

Prof. Nabil Hewahi

Professor

University of Bahrain

Kingdom of Bahrain

March 2023

University of Bahrain

College of Science

Department of Postgraduate Programs

A Novel Classifier Based on Genetic Algorithms and Data Importance Reformatting

Thesis submitted in partial fulfilment of the requirements for the degree of M.Sc. In Big Data Science and Analytics

Prepared by:

Aysha Khaled Alkhayyat

20082338

Supervised by:

Prof. Nabil Hewahi

Professor

University of Bahrain

Kingdom of Bahrain

March 2023

Defense Committee's Approval

University of Bahrain
Deanship of Graduate Studies
And scientific Research

جامعة البحرين
عمادة الدراسات العليا والبحث العلمي

Thesis Discussion Committee Decision

Student Name: Aysha Khaled Alkhayyat	Student ID: 20082338
Department: Choose an item.	College: **Science**
Program: **M.Sc. in Big Data Science and Analytics**	
Thesis Title: A Novel Classifier Based on Genetic Algorithms and Data Importance Reformatting	
Supervisor Name: **Prof. Nabil Hewahi**	Student GPA: **4.00**

The Discussion Committee met on 24-May-23 In S41-1002. Upon discussing the student whose details are stated above in public, the discussion committee recommends the following:

The student is awarded a grade Choose an item. in the thesis subject.

☐ Without making any changes to the thesis.

☒ After making the modifications attached with this decision.

Discussion Committee Members Approval:

Name		Signature
1. **Prof. Nabil Hewahi**	(Advisor)	Nabil Hewahi
2. **Dr. Wael Elmedany**	(Internal Examiner)	
3. **Prof. Mohammad Bsoul**	(External Examiner)	Mbsoul

Approved at the University Council 6th meeting, 2013
In accordance with Decision No. 736/2013 dated 14/3/2013
Modified by University Council Decision No. 2043/2013 dated 20/11/2013.

Abstract

Machine learning (ML) has attracted substantial attention and become progressively more popular in recent years due to its ability to make predictions and decisions based on data. The increasing use of ML in critical fields such as medical diagnosis and fraud detection emphasizes the need to continuously enhance its performance and eventually result in a better decision-making. This can be achieved through the use of optimization methods (OM). However, sometimes the performance of ML algorithms is limited by issues related to the nature of the data which can hinder its performance. Therefore, a novel classification algorithm that is based on Data Importance (DI) reformatting and Genetic Algorithms (GA) named GADIC is proposed in this research to overcome these issues and improve the efficacy and robustness of ML algorithms. The aim of this research is to evaluate the impact of the proposed algorithm on the performance of the classifiers and compare it with other conventional classification algorithms to measure its ability to improve the classifiers' performance. The proposed algorithm comprises three phases which are data reformatting phase which depends on DI concept, training phase where GA is applied on the reformatted training set, and testing phase where the instances of the reformatted testing set are being averaged based on similar instances in the training set. The proposed algorithm has been tested on five existing ML classifiers which are Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Logistic Regression (LR), Decision Tree (DT), and Naïve Bayes (NB). All were evaluated using seven open-source UCI ML repository and Kaggle datasets which are Cleveland heart disease, Indian liver patient, Pima Indian diabetes, employee future prediction, telecom churn prediction, bank customer churn, and tech students. In terms of accuracy, the results showed that, with the exception of approximately 1% decrease in the accuracy of NB classifier in Cleveland heart disease dataset, GADIC significantly enhanced the performance of most ML classifiers using various datasets. In addition, KNN with GADIC showed the greatest performance gain when compared with other ML classifiers with GADIC, with an average increase of 16.79%, followed by SVM with an average increase of 9.03%. LR had the lowest improvement with an average increase of 5.96%.

Dedication

I am honoured to dedicate this research to everyone who supported and inspired me throughout this demanding, yet rewarding, journey.

To my **beloved parents** for their encouragement to constantly seek knowledge and creating something useful and fruitful using this knowledge, a trait that they have instilled in me from a young age.

To my **siblings** for their endless support and encouragement throughout this journey.

To my **dear friend** for her unwavering support and her ability to lift my spirits in this period.

Acknowledgements

I would like to thank the University of Bahrain and Liverpool John Moores University for offering the Big Data Science and Analytics master's program.

I am deeply indebted to **Prof. Nabil Hewahi** for his guidance during the thesis preparation. The passion for Machine Learning was ignited by his unique teaching style which stimulated our curiosity for learning. This work would not have been possible without his generous support.

I am also grateful to **Dr. Sawsan Hilal** for her continuous commitment and support during the program.

Lastly, I would like to extend my sincere thanks to the **program instructors** for their lauded efforts in delivering the program courses.

Table of Contents

Defense Committee's Approval ..... A

Abstract ..... B

Dedication C

Acknowledgements ..... D

Table of Contents ..... E

List of Tables ..... H

List of Figures I

List of Abbreviations ..... J

Chapter One Introduction to the Study 2

1.1. Introduction 2

1.2. Research Problem 4

1.3. Research Questions 4

1.4. Research Objectives 5

1.5. Research Significance 5

1.6. Research Scope and Limitations 6

1.7. Research Organization 7

Chapter Two Background and Literature Review 9

2.1. Background 9

2.1.1. Classifiers 9

2.1.2. GA 12

2.1.3. Data Reformatting 14

2.2. Literature Review 16

2.2.1. Review of the Examined Datasets 16
2.2.2. Optimization Methods 19
2.2.3. Data Reformatting Techniques 22
2.3. Summary 23
Chapter Three Methods and Procedures 26
3.1. Overview 26
3.2. Proposed Algorithm 26
3.2.1. GADIC Procedures 26
3.2.2. Data Reformatting 29
3.2.3. GA 33
3.2.4. Averaging Each Instance in the Testing Set 36
3.3. Tested Classifiers 37
3.4. Performance Metrics 38
3.4.1. Accuracy 38
3.4.2. Precision 39
3.4.3. Recall 39
3.4.4. F1-Score 39
Chapter Four Experiments, Results and Discussion 41
4.1. Datasets Used 41
4.1.1. Data Sources 41
4.1.2. Datasets Description 42
4.1.3. Data Pre-Processing 44
4.2. Experiments 46
4.2.1. Data Reformatting Phase 46

4.2.2. Data Splitting 47
4.2.3. Training Phase 48
4.2.4. Testing Phase 49
4.3. Results 51
4.3.1. Cleveland Heart Disease Dataset 51
4.3.2. Indian Liver Patient Dataset 52
4.3.3. Pima Indian Diabetes Dataset 54
4.3.4. Employee Future Prediction Dataset 55
4.3.5. Telecom Churn Prediction Dataset 56
4.3.6. Bank Customer Churn Dataset 58
4.3.7. Tech Students Dataset 59
4.4. Discussion 60
Chapter Five Conclusion and Recommendations 64
5.1. Conclusion 64
5.2. Recommendations and Future Work 65
References 66
الملخص 78

List of Tables

Table (2.1): Sample of dataset before data reformatting 15
Table (2.2): Sample of the same dataset after data reformatting 16
Table (2.3): Summary of research applied ML algorithms in conjunction with GA... 19
Table (4.1): Datasets used in the experiments 41
Table (4.2): Description of the datasets used in the experiments 43
Table (4.3): Irrelevant attributes in the used datasets 45
Table (4.4): Number of missing and duplicate values in the datasets 45
Table (4.5): Parameters resulted with the best performance in GADIC 50
Table (4.6): Results of Cleveland heart disease dataset 51
Table (4.7): Results of Indian liver patient dataset 53
Table (4.8): Results of Pima Indian diabetes dataset 55
Table (4.9): Results of employee future prediction dataset 56
Table (4.10): Results of telecom churn prediction dataset 57
Table (4.11): Results of bank customer churn dataset 58
Table (4.12): Results of tech students dataset 59
Table (4.13): Increase in classifiers accuracy using GADIC 61

List of Figures

Fig. 3.1: Flowchart of the proposed algorithm 27
Fig. 3.2: Flowchart of GA phase in GADIC 33
Fig. 3.3: Example of single point crossover operator applied on Pima Indian diabetes dataset 35
Fig. 3.4: Example of mutation operator applied on Pima Indian diabetes dataset 35
Fig. 4.1: Distribution of binary class labels datasets 43
Fig. 4.2: Distribution of multiclass labels of tech students dataset 44
Fig. 4.3: Accuracy of heart dataset using different split ratios 48
Fig. 4.4: Change in performance in heart dataset after applying GADIC 52
Fig. 4.5: Change in performance in liver dataset after applying GADIC 54
Fig. 4.6: Change in performance in diabetes dataset after applying GADIC 55
Fig. 4.7: Change in performance in employee dataset after applying GADIC 56
Fig. 4.8: Change in performance in telecom dataset after applying GADIC 57
Fig. 4.9: Change in performance in bank dataset after applying GADIC 59
Fig. 4.10: Change in performance in tech student dataset after applying GADIC 60
Fig. 4.11: Average increase in accuracy per model 62

List of Abbreviations

ML	Machine Learning
OM	Optimization Methods
GA	Genetic Algorithms
GADIC	Genetic Algorithm and Data Importance based Classifier
DI	Data Importance
UCI	University of California, Irvine
SVM	Support Vector Machine
KNN	K-Nearest Neighbour
LR	Logistic Regression
DT	Decision Tree
NB	Naïve Bayes
PSO	Particle Swarm Optimization
TSP	Travelling Salesman Problem
ANN	Artificial Neural Networks
RF	Random Forest
FS	Feature Selection
XGBoost	Extreme Gradient Boosting
MLP	Multilayer Perceptron
GB	Gradient Boosting
CV	Cross Validation
PO	Parameter Optimization
NN	Neural Network
DNN	Deep Neural Network
DL	Deep Learning
CSV	Comma Separated Values

Chapter One

Introduction to the Study

1.1. Introduction

1.2. Research Problem

1.3. Research Questions

1.4. Research Objectives

1.5. Research Significance

1.6. Research Scope and Limitations

1.7. Research Organization

Chapter One
Introduction to the Study

1.1. Introduction

Regardless of the industry and the nature of the business, personnel are always expected to produce more in less time with minimal cost. The phenomenal capabilities of Machine Learning (ML) have helped the industries to be more efficient than ever. With ML, industries were able to automate many repetitive and time-consuming tasks, analyse large amounts of data and make predictions about future events or trends, detect defects or anomalies in products or processes, and optimize resource allocation and scheduling (Forbes Technology Council, 2019). Working with ML on daily basis made it a second nature to the industries. Since the businesses and their needs are continuously evolving, the ML community had to cope with such advancement to cater for the industries' needs.

Continuous improvement is a principle of life. Having an ML algorithm with a better performance is a goal that has been pursued since ever. One may not be able to demonstrate the usefulness of the ML algorithms if they are not achieving the goal that they are designed for. Therefore, it is essential to keep enhancing the performance of such algorithms. ML algorithms can be enhanced via data enhancements, improvements on the algorithms themselves, or Optimization Methods (OM).

In parallel to the exponential growth of the data, a need has emerged for ML algorithms with better prediction capabilities. As mentioned

above, the OM can boost the performance of the applied ML algorithms. In this research, data reformatting has been chosen to be the data enhancement technique to be applied in conjunction with the optimization technique, Genetic Algorithms (GA). The data reformatting concept that is followed in this research is based on research conducted by Hewahi (2019).

In this research, a novel classification algorithm called Genetic Algorithm and Data Importance based Classifier (GADIC) is proposed that is based on GA and Data Importance (DI) reformatting for the ultimate goal of enhancing the performance of the classifiers. GADIC uses existing classifiers to measure its performance. The main approach relies on altering the data representation in accordance with the DI technique presented in Hewahi (2019) and applies GA as an OM to tune and adjust the data. The data reformatting is happening based on the influence of certain values within the attributes in affecting the produced output. Applying GA in both training and testing phases on the reformatted data to tune it and avoid any loss is a key step to enhance the performance of the classifiers.

This research also aims at comparing the performance of GADIC with conventional classification algorithms to assess its ability to improve the classifiers' performance. To achieve this objective, a variety of open-source datasets will be tested in order to investigate GADIC's ability to enhance the performance of the classification algorithms for a number of datasets.

1.2. Research Problem

Data scientists may endeavour to elevate the performance of the tested ML algorithm via altering the hyperparameters. Moreover, they could also aim for better performance by carrying out additional pre-processing steps. Most of approaches used for data pre-processing include up-sampling, down-sampling, removing or filling the missing values, and the removal of redundant records. Nevertheless, there are situations where this isn't possible, and the algorithm's performance is constantly low (Tan, 2020). This can indicate that the problem lies within the data itself.

In several cases, many classifiers do not perform well and get low performance because of the nature of the data. In this research, a novel classification algorithm that uses existing classifiers to improve their performance is proposed. The concept of the proposed algorithm is based on reformatting the data using DI technique proposed by Hewahi (2019) and applying GA to tune the data in both training and testing phases. This algorithm should be capable of overcoming some of data related impediments that might prevent the classifier from performing well, hence, elevating the performance of the classifier.

1.3. Research Questions

Under this section, we will touch upon the questions that triggered the idea of the research topic. In other words, the questions listed below will be answered throughout this research to prove the earlier mentioned assumption regarding GADIC's effectiveness. The primary questions of the research are as follow:

1. How does the proposed algorithm contribute to the classifiers' performance?

2. How effective is the proposed algorithm in comparison to the conventional classification algorithms?

1.4. Research Objectives

The aim of this research is to develop a novel classification algorithm based on GA and DI reformatting that can function properly with different types and sizes of datasets to improve the classifiers' performance. The main objectives of the research are as follow:

1. Develop a novel classifier based on GA and DI reformatting to improve the classifiers' performance.
2. Evaluate the impact of the developed algorithm on the performance of the classifiers using the performance measures, namely, accuracy, precision, recall, and f1-score.
3. Compare the performance of the developed algorithm with other conventional classification algorithms to gauge its ability to improve the classifiers' performance.

1.5. Research Significance

This research proposed a novel classifier based on GA and DI reformatting that can be utilized to different classification datasets to produce highly accurate predictions. To the best of the researcher's knowledge, no other research has experimented GA to tune the data in combination with the data reformatting method proposed by Hewahi (2019).

In vital sectors, providing accurate predictions is a crucial matter. Enhancing the prediction capabilities of ML models can assist diverse

fields to accurately predict the occurrence of certain conditions that are of paramount value to them.

It is anticipated that the proposed algorithm will show superior classification abilities in comparison to the conventional classification algorithms, hence, adding practical value to the evolvement of the ML classifiers and accordingly accelerating its adoption in real life applications.

1.6. Research Scope and Limitations

Applying the proposed algorithm on different sets of data of various sizes and fields is essential to evaluate its potential to increase the classification performance and produce highly accurate predictions. Massive experiments with different settings should be undertaken in order to give a comprehensive and holistic judgement on the performance of the proposed algorithm.

In this research, the proposed algorithm will be tested using open-source datasets from well-known repositories, namely Kaggle and University of California, Irvine (UCI) ML repository. The research is solely limited to classification datasets either of binary or multiclass labels. The sizes of the datasets have been selected carefully to prove the capability of GADIC to enhance the performance of the classifiers. Datasets of random sizes of ranges less than 1000 instances, between 1000 and 10,000 instances, and up to 20,000 instances will be experimented.

1.7. Research Organization

Along with chapter 1, in which an introduction to the research topic, the research problem, the research significance along with the research objectives and research scope were stated, this research comprises 4 additional chapters.

Chapter 2 which is titled background and literature review, presents the background about the classifiers and GA as optimization method along with data enhancement methods. Moreover, it provides a review of studies related to the research topic. Under the methods and procedures in chapter 3, the core work of the proposed novel algorithm is described. Detailed steps are provided for each part of the proposed algorithm.

As for chapter 4, that is titled experiments, results, and discussion, a description of the datasets that were used in the experiments is provided. Furthermore, the experiments conducted on the selected datasets with the results obtained are presented and discussed. The final chapter is chapter 5 and is titled conclusion and recommendations. In this chapter, a summary of the whole research and proposed work is provided as well as recommendations for future work.

Chapter Two

Background and Literature Review

2.1. Background

2.2. Literature Review

2.3. Summary

Chapter Two
Background and Literature Review

2.1. Background

In this section, a background information about the concepts and themes that form the basis of this research is provided. The three core elements of this research are ML classifiers, GA as an OM, and data reformatting.

2.1.1. Classifiers

The classification algorithms that will be tested in the proposed algorithm are briefly explained in the subsections that follow. Because of their extensive applications, which made them the most common ones, those classifiers were selected (Ray, 2019).

2.1.1.1. Support Vector Machine (SVM)

SVM is a supervised method that its decision function is an optimal hyperplane that separates classes of the observations based on their features. Accordingly, the best hyperplane is the one that maximizes the margin between classes by utilizing the support vector points. Recently, the community of ML draw more attention to SVM due to its exceptional generalization capability and discriminative power as it is able to handle high-dimensional data (Cervantes et al., 2020; Pisner & Schnyer, 2020). SVM is widely utilized in a variety of applications such as text

classification, image classification, fraud detection, and bioinformatics (Cervantes et al., 2020; Ray, 2019).

The advantage of SVM is that it can handle both semi-structured and structured data by using a kernel function to transform the input space into a higher dimensional space. However, SVM can be computationally expensive especially with large datasets due to the extended training time. Furthermore, it has the difficulty of selecting the right kernel function (Ray, 2019).

2.1.1.2. K-Nearest Neighbour (KNN)

KNN is a non-parametric algorithm which essentially reflects the number (k) of neighbours in the training data that are closest to a specific testing point. The most crucial step in KNN is choosing the proper k value. The nearest neighbours are selected based on Euclidean Distance Function. Eventually, the label of the given testing point is determined by a majority vote of the chosen k nearest neighbours (Taunk et al., 2019; Pandey & Jain, 2017; Cheng et al., 2014).

Despite being a simple and flexible algorithm, KNN has a high computation cost and is sensitive to the selection of k value as it is the only parameter (Ray, 2019; Taunk et al., 2019; Cheng et al., 2014). Applications of KNN include handwriting identification, medical condition diagnosis, credit scoring, stock market forecasting, pattern recognition, and image recognition (Ray, 2019; Pandey & Jain, 2017).

2.1.1.3. Logistic Regression (LR)

LR is a statistical method for binary and multiclass classification problems in ML as a sort of regression (Connelly, 2020; Maalouf, 2011). It is used to discover the relationship between a dependent variable and

one or more independent variables in addition to predict a binary outcome based on those independent variables. The fundamental notion of LR is that it provides probability estimations that range from 0 to 1 (Maalouf, 2011). The main drawback of this algorithm is its assumption of linearity between input and output. Yet, it is easy to implement, train, and interpret (Ray, 2019; Maalouf, 2011). Numerous fields utilize LR widely, including engineering, finance, marketing, cancer diagnosis, and risk prediction of specific illness (Ray, 2019).

2.1.1.4. Decision Tree (DT)

A DT is a tree-based classification algorithm in which data is recursively splitting based on a specific threshold. Starting with a root node that represents the complete dataset, DT splits the data into smaller subsets based on the features and their values until final leaves are reached. In the final leaves, each subset consists only of instances of a single class (Jijo & Abdulazeez, 2021; Ray, 2019; Priyam et al., 2013). DT is renowned for its ease in implementation and handling quantitative, qualitative, and missing data. Nonetheless, DT has flaws such as instability and difficulty in controlling the tree size (Jijo & Abdulazeez, 2021; Ray, 2019). Applications of DT include forecasting future book usage and tumour prognostic issues (Ray, 2019).

2.1.1.5. Naïve Bayes (NB)

NB is a probabilistic method that implements Bayes Theorem to estimate the probabilities of the dataset's values. This is done by calculating the frequency of their occurrence. The assumption that the attributes are independent in NB is problematic. However, it is known for its scalability, ability of handling missing values and multi-class

problems, as well as its suitability to handle high dimensional data (Peling et al., 2017; Taheri & Mammadov, 2013). Despite that, NB's simplicity may result in lower performance compared to other models that have been adequately trained and tuned (Ray, 2019). NB has been used for a variety of applications in the real world, including fraud detection, email filtering, recommender systems, and medical diagnostics (Ray, 2019; Taheri & Mammadov, 2013).

2.1.2. GA

In ML, OM are algorithms that seek for the best solution to a problem by minimizing or maximizing an objective function. Such methods are used to train ML models and find the optimal model parameter values that minimize the cost function and enhance the model performance (Secherla, 2021; Brownlee, 2021a; Serokell, 2020). The specific requirements and characteristics of the data and problem influence the selection of OM to be applied (Venter, 2010). Different types of OM are employed in ML including Gradient Descent, Adam, Particle Swarm Optimization (PSO), Ant Colony Optimization, and GA (Serokell, 2020; Venter, 2010).

GA has demonstrated its effectiveness as a robust OM. This, together with its flexibility, simplicity in comprehending, capability of finding the global optimum solution, and ability of solving complex optimization problems that are difficult to solve using other OM, contributes to its increasing recognition and adoption (Haldurai et al., 2016; Tabassum & Mathew, 2014). With concentrating on bio-inspired operators, GA is widely employed to provide high quality solutions to optimization and search problems. GA can settle problems in the real

world by replicating this process (Alam et al., 2020; Dahiya & Sangwan, 2018).

The GA lies in the family of evolutionary algorithms, a type of meta-heuristic optimization algorithm, which are computational search methods that draws its inspiration from natural selection (Vié et al., 2021; Savio & Chakraborty, 2019; Dahiya & Sangwan, 2018). The Darwinian theory of "Survival of the fittest" is the cornerstone of the GA (Katoch et al., 2021; Savio & Chakraborty, 2019). GA operations rely on randomness to set the values of their parameters (Alam et al., 2020).

GA is carried out through a series of standardized processes. In this method, the problem space is abstracted as a population of individuals, and it attempts to discover which individual is the most fit by constructing generations repeatedly. The fitness function, which identifies the criterion for rating prospective populations and choosing them for inclusion in the following generation, determines the optimal solution to a given problem (Haldurai et al., 2016; Haldar et al., 2014).

Selection, crossover, and mutation are the three main genetic operators that are consecutively applied with specific probabilities to every individual during each generation until a predefined termination criterion is met (Haldurai et al., 2016). The termination condition is a stopping criterion that determines when the GA should stop iterating and return the optimal solution found. It can be based on a chosen number of generations, a convergence threshold which is met when the objective function value reaches a predetermined value, or after a predetermined number of generations with no improvement in the population (Savio & Chakraborty, 2019). Based on the fitness values of the individuals in the population, the selection operator determines which individuals of the population will be selected for reproduction in the following generation. The crossover operator involves exchanging genes between two selected

individuals (by selection operator) to generate new offspring with genes from both parents, while mutation operator randomly alters some of the genes in an individual. Each GA operator has a variety of forms. The common crossover operators are the single-point, two-point, k-point, and uniform operators. Displacement, simple inversion, and scramble mutation are some of mutation operators forms (Katoch et al., 2021). In Chapter 3, a detailed description of the GA process will be provided.

GA, though, encounter significant challenges that have constrained its uses. One thing is its high computational cost. For satisfactory results, GA may need to run for a lengthy time. Additionally, a larger population or complex problems typically cause a considerable slowdown of GA. Another thing is the difficulty in determining the suitable parameter's settings i.e. selecting the initial population, fitness function, and degree of crossover and mutation (Katoch et al., 2021; Vié et al., 2021).

Notwithstanding the aforementioned challenges, GA's remarkable outcomes have prompted a widening of its use in a number of applications as a result of its successes in solving complex real-world problems. Indeed, its applications go beyond ML. It extends to, for instance, image processing, scheduling applications, engineering, Travelling Salesman Problem (TSP) and sequence scheduling, Internet of Things, aircraft architecture, and way more (Katoch et al., 2021; Alam et al., 2020; Almufti et al., 2019; Dahiya & Sangwan, 2018).

2.1.3. Data Reformatting

As highlighted in the research problem, Chen et al. (2021) stated that the performance of ML algorithms is directly impacted by the nature and quality of the data as it may dramatically reduce the effectiveness of ML. By enhancing the performance and quality of the input data, data

reformatting techniques can contribute to the optimization of ML algorithms.

Within this in mind, many experiments have been conducted to come up with data enhancement techniques via applying some sorts of data reformatting to tackle this issue and increase the classification performance of such data.

The data reformatting method employed in this research was adopted from research conducted by Hewahi (2019). According to Hewahi (2019), the method is formulated on how frequently a specific attribute value is replicated across all dataset instances. Based on potential changes in the attribute's values across the dataset, the importance of each input for each attribute is determined.

The essence of the method is that an attribute's relevance in influencing the output generated decreases in proportion to how small its values changes. Likewise, the more an attribute's values changes across the dataset, the more important it is to the process of producing a certain outcome. The importance of each input is calculated using equations established by the author of that research. All data inputs in the dataset are eventually replaced with their estimated importance values. Table 2.1 and Table 2.2 show an example of data points in one of the experimented datasets (Indian Liver Patient dataset) pre and post data reformatting method applied in this research.

Table (2.1): Sample of dataset before data reformatting

Age	Gender	TP	DB	AP	AA	AM	TP	A1	AG	Outcome
40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
29	Female	0.7	0.1	162	52	41	5.5	2.5	0.9	2
58	Male	1	0.4	182	14	20	6.8	3.4	1	1
72	Male	3.9	2	195	27	59	7.3	2.4	0.4	1

Table (2.2): Sample of the same dataset after data reformatting

Age	Gender	TP	DB	AP	AA	AM	TP	A1	AG	Outcome
38.76	1.23	0.51	0.17	287.72	225.7	239.8	5.52	2.61	0.62	1
60.08	0.78	6.19	3.11	686.42	62.27	97.90	6.09	2.69	0.58	1
28.10	1.23	0.39	0.05	159.08	50.59	40.13	4.22	2.10	0.7	2
56.20	0.78	0.57	0.23	178.72	13.62	19.58	5.52	2.86	0.78	1
69.77	0.78	2.22	1.13	191.49	26.27	57.76	5.93	2.02	0.31	1

2.2. Literature Review

A considerable amount of research has been reviewed on various axes in order to identify the evolution in the ML optimization domain, allowing for a more thorough perception and understanding of the research that has been conducted in this domain to be able to spot areas for improvement and add value to this field. The following subsections will shed the light on the studies that were conducted on the datasets utilized in this research, as well as enhancements in both algorithm and data level.

2.2.1. Review of the Examined Datasets

Abundance of research have evaluated the performance of various supervised learning classification algorithms by using various settings, parameters, or pre-processing steps on the selected datasets for this research. The results vary depending on the platforms or steps employed. Reviewing these studies enables one to comprehend the nature of the data and the extent of the possible improvement for these datasets.

Cleveland heart disease dataset is one of the most common benchmark datasets that was experimented in numerous studies. A research has been conducted to compare the performance of LR, KNN, SVM, Artificial Neural Networks (ANN), NB, and Random Forest (RF)

algorithms using six different platforms. ANN using Matlab platform had the highest accuracy of 85.86% followed by SVM with 83.84% (Tougui et al., 2020). Another research compared KNN and RF performance. With results of 86.88% and 81.96%, respectively, KNN surpassed RF (Garg et al., 2021). A research conducted by Shah et al. (2020) demonstrated that implementing pre-processing steps as well as bagging or boosting on the dataset improved the algorithms' performance. Transformation i.e., smoothing, normalization, and aggregation tasks was used to apply the NB, DT, KNN, and RF algorithms on cleaned data. KNN achieved an accuracy of 90.78%, whereas NB had an accuracy of 88.15%. Meanwhile, when bagging was applied on SVM and DT, the outcomes were 84.15% and 78.54%, respectively (Pouriyeh et al., 2017).

Another common benchmark dataset for experiments is the liver disease dataset. A study was performed using six algorithms namely, LR, KNN, DT, SVM, NB, and RF. The best accuracies were 75% for LR and 74% for RF (Rahman et al., 2019). However, using Feature Selection (FS) with the same tested algorithms as the preceding study had shrunk the accuracy of LR and RF but enhanced the accuracy of SVM to 70% (Rajan et al., 2022). The FS was used in yet another study with RF, perceptron, KNN, SVM, and DT. With 72% accuracy, KNN was the most accurate (Azam et al., 2020). In a research conducted by Ghosh et al. (2021), SVM, RF, Extreme Gradient Boosting (XGBoost), DT, KNN, AdaBoost, and LR are the ML algorithms which outputs were evaluated. The algorithm that had the highest accuracy (83.70%) was the RF. Priya et al. (2018) employed min-max normalization along with PSO FS on the dataset. Through applying RF, SVM, Multilayer Perceptron (MLP), NB, and J48 on that dataset, J48 had an accuracy of 95.04%, the highest of the other tested classification algorithms.

The Pima Indian diabetes dataset has as well undergone extensive research. A research conducted by Rawat and Suryakant (2019) applied AdaBoost, LogicBoost, RobustBoost, NB, and Bagging algorithms on this dataset. With bagging and AdaBoost approaches, the results showed a classification accuracy of 81.77% and 79.69%, respectively. A remarkable result was found while comparing Gradient Boosting (GB) to LR and NB. Compared to NB's 77% and LR's 79%, it had an accuracy of 86% (Birjais et al., 2019). However, in Soni and Varma's research (2020), RF outperformed GB in terms of accuracy. Battineni et al. (2019) used Cross Validation (CV) to apply four ML classifiers NB, J48, LR, and RF. The results showed that LR had the highest accuracy (77%), followed by NB with 76%. A research conducted by Khanam and Foo (2021) performed pre-processing steps on the dataset prior to running seven ML algorithms on it. These steps include FS, normalization, and the removal of missing values and outliers. While Neural Network (NN) obtained an 88.6% accuracy, LR and SVM produced accuracy of around 77%–78%.

To predict the telecom churn, a thorough examination of the IBM Watson dataset has been conducted. A research has been conducted by Pamina et al. (2019) employed KNN, RF, and XGBoost on the dataset. Compared to KNN and RF, the XGBoost classifier's accuracy was the greatest (79.8%). Similarly, RF, SVM, DT, and LR were applied in another research. With 93.5% accuracy, RF led other classifiers, followed by SVM and LR with 78.9% and 76.2%, respectively (Sundararajan & Gursoy, 2020). Fujo et al. (2022) utilized the random oversampling approach in conjunction with label encoding, one hot encoding, and normalization as well as other pre-processing steps. In terms of accuracy, the Deep-BP-ANN model surpasses the XGBoost, LR, NB, and KNN models.

As for bank customer churn dataset, DT and ANN models were applied to the dataset. In contrast to the DT model's accuracy of 79.77%, ANN had an accuracy of 86.52% (Kumar et al., 2019). As expressed earlier, performing FS and imbalanced data techniques increases the classification performance. This finding was supported by Huseyinov and Okocha's research (2022) and Rahman and Kumar's research (2020). In both papers, RF outperformed the other classifiers (KNN, SVM, DT, LR, GB, eXtreme GB, and Light GB) with outstanding accuracy up to 95.74% in the latter research.

2.2.2. Optimization Methods

The optimization of ML algorithms can be carried out in a variety of methods, including FS, Parameter Optimization (PO), regularization, model selection, and ensemble methods (Brownlee, 2021b; Bains, 2020; Brownlee, 2020b). A summary of the reviewed GA-based ML algorithm optimizations is presented in Table 2.3.

Table (2.3): Summary of research applied ML algorithms in conjunction with GA

Reference	Applied ML algorithms	Purpose of applying GA	Datasets
Azad et al. (2022)	DT	FS	UCI Pima Indian diabetes
Demir and Şahin (2022)	SVM RF XGBoost	FS + PO	Vs dataset
Ashri et al. (2021)	Ensemble DT and RF	FS	UCI heart disease
Doewes et al. (2021)	RF	Dimensionality Reduction	Dataset of blood samples for covid-19
Hamdia et al. (2021)	Deep Neural Network (DNN)	PO	Established dataset to predict fracture energy

Table (2.3): Summary of research applied ML algorithms in conjunction with GA (cont.)

Reference	Applied ML algorithms	Purpose of applying GA	Datasets
Kanwal et al. (2021)	SVM Deep Learning (DL) NB NN LR	FS	UCI heart disease datasets (Cleveland and Statlog)
Książek et al. (2021)	LR	FS + PO + weight optimization	Patient data collected at Coimbra's Hospital and University Centre, Portugal
Maleki et al. (2021)	KNN	FS	Lung cancer dataset from data world site
Soumaya et al. (2021)	SVM	FS	Parkinson
Farid et al. (2020)	C4.5	FS	UCI Wisconsin
Khotimah et al. (2020)	NB	FS	UCI datasets (glass, heart disease, WDBC, sonar, record linkage comp patterns)
Safitri and Muslim (2020)	NB	FS	UCI German credit dataset
Damanik et al. (2019)	C4.5	Search rule optimization	UCI accurate occupancy detection
Rahmadani et al. (2018)	DT NB	FS	UCI datasets (iris, glass, credit)
Pawlovsky and Matsuhashi (2017)	KNN	FS	UCI breast cancer
Phan et al. (2017)	SVM	PO + Feature weighting	UCI datasets (German (credit card), Australian (credit card), Pima Indian diabetes, heart disease (statlog), breast cancer, contraceptive method choice, ionosphere, iris, sonar, vehicle, vowel)
Aly (2016)	LR	FS + PO	UCI Pima Indian diabetes

Table (2.3): Summary of research applied ML algorithms in conjunction with GA (cont.)

Reference	Applied ML algorithms	Purpose of applying GA	Datasets
Choubey et al. (2016)	NB	FS	UCI Pima Indian diabetes
Syarif et al. (2016)	SVM	PO + FS	UCI datasets (dexter, internet_ads, Madelon, musk, spambase, spectf heart, intrusion)

All studies reported in Table 2.3 revealed an improvement in classification performance when applying GA as an OM with ML algorithms. Likewise, GA demonstrated superiority to the grid search approach in Syarif et al. (2016). Most studies showed a high rate of accuracy growth. The accuracy reached 100% when KNN was combined with GA using the lung cancer dataset (Maleki et al., 2021). Among various algorithms that were integrated with GA, it was observed that NN, XGBoost, and RF algorithms achieved the highest increase in performance. Furthermore, the performance improvement of SVM and DT is greater than that of NB and LR (Demir & Şahin, 2022; Kanwal et al., 2021; Rahmadani et al., 2018). In conjunction with GA, applying ensemble methods instead of a single algorithm led to yet more improvement in performance (Ashri et al., 2021).

Within the context of incessant improvement processes in the ML domain and the continuing pursuit of the best outcomes, researchers sought to employ additional optimization steps to enhance the classification performance (Kaseb & Rahbar, 2022). From that standpoint, some research has progressed beyond the use of a single OM to the integration of multiple OM for that purpose. A hybrid of GA and PSO OM, for instance, has been applied in several studies to improve classification performance (Kaseb & Rahbar, 2022; Chen et al., 2020;

Jabor & Ali, 2019). This integration was utilized by Kaseb and Rahbar (2022) and Jabor and Ali (2019) for FS only. Chen et al. (2020) on the other hand employed the hybrid GA and PSO, with GA being used for FS and PSO for PO. The results of applying this model on SVM and ANN revealed an improvement in performance with a greater and steadier rise in ANN. Typically, the hybrid optimization model outperformed the single OM in the research that was examined.

2.2.3. Data Reformatting Techniques

Whereas there were advancements at the algorithmic level, there were similarly some studies that have made improvements but at the data level itself. Those improvements took the form of data reformatting in such a way that enhance the classification performance.

A new encoding approach based on existing encoding techniques was proposed by Tabassum et al. (2020) to handle categorical variables in a dataset. The developed approach, known as "one-hot-frequency", combines one-hot and frequency encoding methods. A number of ML algorithms' classification performance was improved by the proposed approach. The best performance, though, came from RF followed by DT and ANN.

A strategy for image quality enhancement was introduced by Hung et al. (2021). This method can enhance image features by incorporating technologies such as noise reduction, sharpening, and color channel adjustment. This method was combined with DL model to improve the classification performance.

The data enhancements used in the aforementioned research are based on pre-existing techniques, hence the extent of the data reformatting is constrained by these techniques. Conversely, Hewahi

(2019) proposed a new method for data reformatting in which all the dataset inputs are eventually replaced based on their importance in affecting the produced output. The breast cancer, iris, and diabetes datasets from the UCI ML repository were subjected to the proposed data reformatting technique. Employing this technique in conjunction with NN pruning algorithm has yielded classification performance that is superior to that of other NN pruning algorithms in most of the tested datasets. Therefore, this data reformatting technique was selected to be applied in this research due to its remarkable outcomes.

2.3. Summary

The efficacy of employing OM to solve ML problems and boost performance has been demonstrated in the previous subsections 2.2.2 and 2.2.3. These outstanding results, however, also came with several shortcomings that limited the extent of its effectiveness.

Being a time consuming is one of the most frequent drawbacks when applying OM (Demir & Şahin, 2022; Książek et al., 2021; Chen et al., 2020; Phan et al., 2017). This is due to the high computational complexity of applying such techniques in some cases, particularly with the growth in data size, number of algorithms' parameters, and number of features in the dataset (Demir & Şahin, 2022; Hamdia et al., 2021; Chen et al., 2020). In addition, it has been noted that selecting the proper fitness function and parameter values of GA can be challenging and that incorrect values can easily cause GA to become trapped in a local optimum solution (Demir & Şahin, 2022; Phan et al., 2017).

Additionally, most of the examined research in subsection 2.2.2 used GA either for FS or PO or both. This is not always advantageous. When

there are few features in the dataset, FS may not always be beneficial (Azad et al., 2022). Moreover, the lack of specific value ranges in PO technique makes finding the best value more time-consuming (Syarif et al., 2016). Regardless of the selected approach in GA, Kaseb and Rahbar (2022) recommended increasing the number of generations in GA as it requires more generations to reach the optimal solutions.

The relevant literature was explored to look for the methods adopted to enhance the performance of ML algorithms. The vast majority of the methods were evolving around enhancing the ML itself. To the best of the researcher's knowledge, almost no papers addressed enhancing the performance via data reformatting. Hence, the idea of this research emerged to fill in the identified gap in the literature. Therefore, a novel classification algorithm named GADIC that is based on GA and DI reformatting is proposed in this research based on the conducted literature review to address those shortcomings and improve the classification performance. The novelty in GADIC is that although it uses existing classifiers to measure its performance, it combines DI reformatting with GA, with GA being employed in both the training and testing stages.

Although the effectiveness of data reformatting depends on the specifics of the problem and the data, the researcher opted for designing a general method that is capable to work with any dataset and give an acceptable improve in performance. Since XGBoost, NN, and RF had achieved the highest rate of increase in performance in the examined literature, SVM, KNN, LR, DT, and NB were chosen to be tested in this study to examine their possibility of further improvement with GADIC. The tested ML algorithms will be trained with default hyperparameters. It is expected that the proposed algorithm will enhance the performance of the tested ML algorithms, even if marginally.

Chapter Three

Methods and Procedures

3.1. Overview

3.2. Proposed Algorithm

3.3. Tested Classifiers

3.4. Performance Metrics

Chapter Three
Methods and Procedures

3.1. Overview

The proposed algorithm in this research is described in this chapter. It comprises three main phases: data reformatting of the whole dataset, training phase where GA is applied on the reformatted training set, and testing phase where the reformatted testing set is mapped similarly to the training phase by averaging each instance in the testing set. The ML classifier in question is then applied on the testing set to assess the effectiveness of the proposed algorithm.

3.2. Proposed Algorithm

The following subsections present the procedures of GADIC followed by detailed steps of the three main phases of the proposed algorithm along with the mathematical formulas of them.

3.2.1. GADIC Procedures

In order to evaluate GADIC's potential to enhance performance, it has been tested on a variety of ML classifiers and datasets. Various experiments were conducted using different datasets. Five ML classifiers were tested on each dataset to assess GADIC's potential to enhance the performance of the classifier in question and compare the results. This subsection presents the entire implementation procedures of the proposed

algorithm. The steps of the proposed algorithm are illustrated in Figure 3.1.

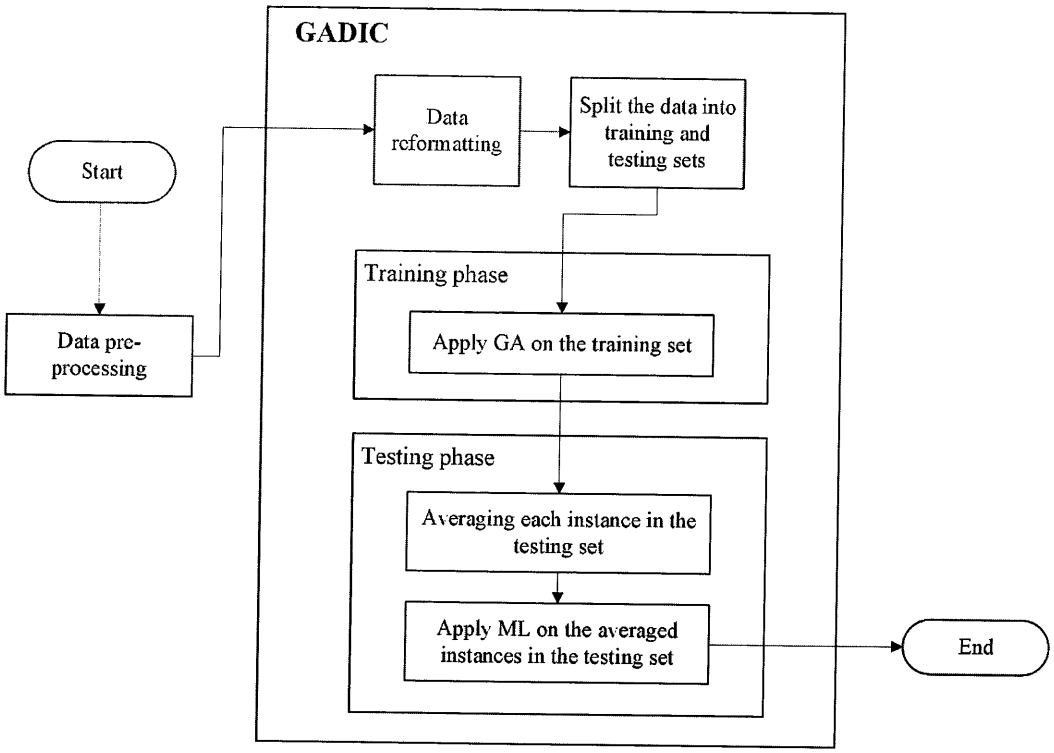


Fig. 3.1: Flowchart of the proposed algorithm

The datasets must go through a few pre-processing steps before undergoing any GADIC procedure. These steps solely entail removing missing or redundant data and applying encoding techniques to categorical attributes. As GADIC focuses on resolving issues related to the nature of data itself, no further data pre-processing steps are carried out in this research.

Data reformatting is the first phase of GADIC. This research adopts the proposed method by Hewahi (2019) for reformatting the data. At this phase, the entire dataset will be reformatted to represent the significance of its values. For each attribute, the importance of its data inputs is calculated to replace those inputs in the dataset as will be described in subsection 3.2.2. In this phase, the proportions of the attribute values are

calculated. These proportions are used to compute the changing factor for each attribute. The importance of each attribute input is then determined using the computed changing factor. Finally, the attribute's importance value is used in place of the attribute input in the dataset. The steps of reformatting vary depending on the type of attribute values. The reformatted dataset is then divided using 70/30 split into training and testing sets.

Next, the training set is used to implement the GA steps that will be described in subsection 3.2.3. In each generation, two instances are selected, and they undergo crossover and mutation operators to produce two new instances. Based on the defined probabilities for crossover and mutation, these operators are carried out. Whenever those GA operators take place in the present generation, the chosen ML in the experiment is applied on the dataset containing the new instances in order to compute its accuracy. It should be stressed that the ML classifiers are only going to be trained using their default parameters. The accuracy of the current generation and the prior generation are then compared using the fitness function. If it is higher, the new instances will take the place of the original instances in the initial training set. Up until the termination condition is satisfied, the procedures of this phase are carried out repeatedly in each generation. The final training set is then extracted to be used in the next phase.

The averaging of the instances of the testing set comes next. The training set is served as the foundation for this phase. At this phase, per instance in the testing set, the Euclidean distance function is used to identify the similar instances from the training set to that instance. Prior to applying the Euclidean distance function, the number of chosen similar instances is determined as will be explained in subsection 3.2.4. Then, the average value for each attribute in the identified similar instances is

calculated. The calculated average values are then considered as the new instance that will take the place of the original instance in the testing set.

Following the completion of all phases, GADIC's final step is to evaluate how well it performed. The testing set is used in the present experiment to apply the chosen ML classifier on it. It should be pointed out that the ML classifier that will be used in the testing phase in the present experiment is the same one used for the training phase. Several metrics will be used to assess GADIC's performance and compare the results. Accuracy, precision, recall, and f1-score will be used to gauge GADIC's effectiveness. Algorithm 1 summarizes the implementation steps of the proposed algorithm.

3.2.2. Data Reformatting

In this phase, based on Hewahi (2019), the importance of each attribute input in producing certain output in a dataset will be calculated. Changing factor β for each attribute will be used to calculate the importance of each attribute input. Each input in that attribute will be replaced by its importance value. All steps will be carried out for each attribute independently in the dataset in order to compute the importance of its inputs and replace their values in accordance with their importance. The main steps of the data reformatting phase are as follow:

1. Select a random sample of the given dataset where s is the number of randomly selected instances.
2. In the case where the attribute has categorical or integer datatypes, the following steps will be followed:
   a. Each attribute i has p_i possible values. For each possible value in the attribute i, the proportion of every attribute input a_i is calculated as shown in Equation (3.1):

$$r_{im} = \frac{v_{im}}{s}, \quad (3.1)$$

where,

v_{im} is the number of m-equivalent values in the random sample for a_i.

b. All r_{im} values calculated by Equation (3.1) is then appended to a set R_i that will contain all proportion values of a_i.

$$R_i = \{r_{im1}, r_{im2}, r_{im3,\ldots,} r_{imp}\}$$

c. Then, the set R_i is utilized to calculate the changing factor β_i for each attribute i using Equation (3.2):

$$\beta_i = \frac{maxR_i - minR_i}{2}, \quad (3.2)$$

where,

$maxR_i$ is the maximum value in R_i,

$minR_i$ is the minimum value in R_i.

d. To reduce the risk of bias in the random selection process, the step of the random selection of instances and the subsequent stages are repeated for w times. Then, the average of β_i is calculated using Equation (3.3):

$$avg\beta_i = \frac{\beta_{i1} + \beta_{i2} + \cdots + \beta_{iw}}{w}, \quad (3.3)$$

where,

w is the number of repetitions of the random selection step.

e. Calculate the importance of each input value a_i for each instance as given in Equation (3.4):

$$I_{iy} = X_{iy}(1 - \beta_i), \quad (3.4)$$

where,

X_{iy} is the input value for the attribute i of instance y,

I_{iy} is the importance of X_{iy}.

f. Replace each input value a_i in attribute i with its calculated importance value I_{iy} in the dataset.

3. While in the case where the attribute has real numbers (float datatype), only parts a and b will be different than those of integer or categorical data (step 2) due to the nature of data inputs that requires different set of calculations. Remaining steps for real data (c until f) will be the same as identified earlier in step 2. Steps a and b for real data are as follow:

   a. Calculate the size of interval int_i for the attribute inputs a_i of the attribute i as given in Equation (3.5):

$$int_i = \frac{maxValue_i - minValue_i}{pt_i}, \tag{3.5}$$

   where,

   $maxValue_i$ is the maximum value in attribute i,

   $minValue_i$ is the minimum value in attribute i,

   pt_i is an integer number that specifies the number of data partitions for the dataset in which the more is this number, the less is the size of the interval and vice versa.

   b. Find the first and last value of every interval based on the calculated int_i. After that, assign each attribute input a_i in the attribute i to the interval to which it belongs. Then calculate the proportion of each a_i using Equation (3.6):

$$r_{im} = \frac{c_{im}}{s}, \tag{3.6}$$

   where,

   c_{im} is number of values in a certain interval within the attribute i in the random sample s.

The calculated changing factor β_i is a value between 0 and 0.5. Smaller values of β_i indicates that the value of a_i is fluctuating and thus have a greater impact on producing the output. Whereas higher B_i values indicate that the value of a_i does not fluctuate significantly and may thus have less impact on producing the output.

Algorithm 1: GADIC Algorithm

Input:
random sample size s, number of repetitions w, number of intervals pt_i, termination generation tg, crossover probability cpb, crossover point cpt, mutation probability mp, mutation rate mr, mutation value mv, number of closest rows n
Start
Data reformatting phase
Input of phase: random sample size s, number of repetitions w, number of intervals pt_i
For (i in range (attributes in the dataset))
    Select s
    If (datatype = integer)
        Calculate proportion of possible values r_{im} using Eq. 3.1
        Calculate βi using Eq. 3.2
    Elseif (datatype = float)
        Calculate size of interval int_i using Eq. 3.5
        Calculate proportion r_{im} using Eq. 3.6
        Calculate βi using Eq. 3.2
    Repeat w times to find averageβi using Eq. 3.3
    Compute importance I_{iy} using Eq. 3.4
    Replace attribute values with their I_{iy}
Output of phase: reformatted dataset based on data reformatting phase
Data splitting
Split the reformatted dataset
Training phase
Input of phase: termination generation tg, crossover probability cpb, crossover point cpt, mutation probability mp, mutation rate mr, mutation value mv
Set count = 0
Set initial population = training set
While (count < tg)
    Select 2 instances as parents from population randomly
    Apply crossover on parents based on cpb
    Apply mutation on offspring obtained from crossover based on mp
    Compute fitness function for the new offspring
    If (fitness = True)
        Replace parents with offspring in the initial population
    Else
        Keep parents and don't replace with offspring in the initial population
    Count+=1
End while
Output of phase: final training set
Testing phase
Input of phase: number of closest rows n
For (i in range (instances in the testing set))
    Find n closet instances in the training set using Euclidean distance function
    For (i in range (attributes in the testing set))
        Calculate average attribute value of the retrieved closest instances using Eq. 3.8
Apply ML classifier on the averaged instances
Assess the performance
Output of phase: performance measures results
End

3.2.3. GA

In this phase, the training set will be subjected to GA operators in order to fine-tune the data. The whole training set constitutes the population in the proposed algorithm. Also, the fitness function is built to find the attribute values that will yield to the best performance. The steps in this phase will be repeated until the termination condition specified at the beginning is met. The sequence of the steps of GA phase in GADIC is illustrated in Figure 3.2.

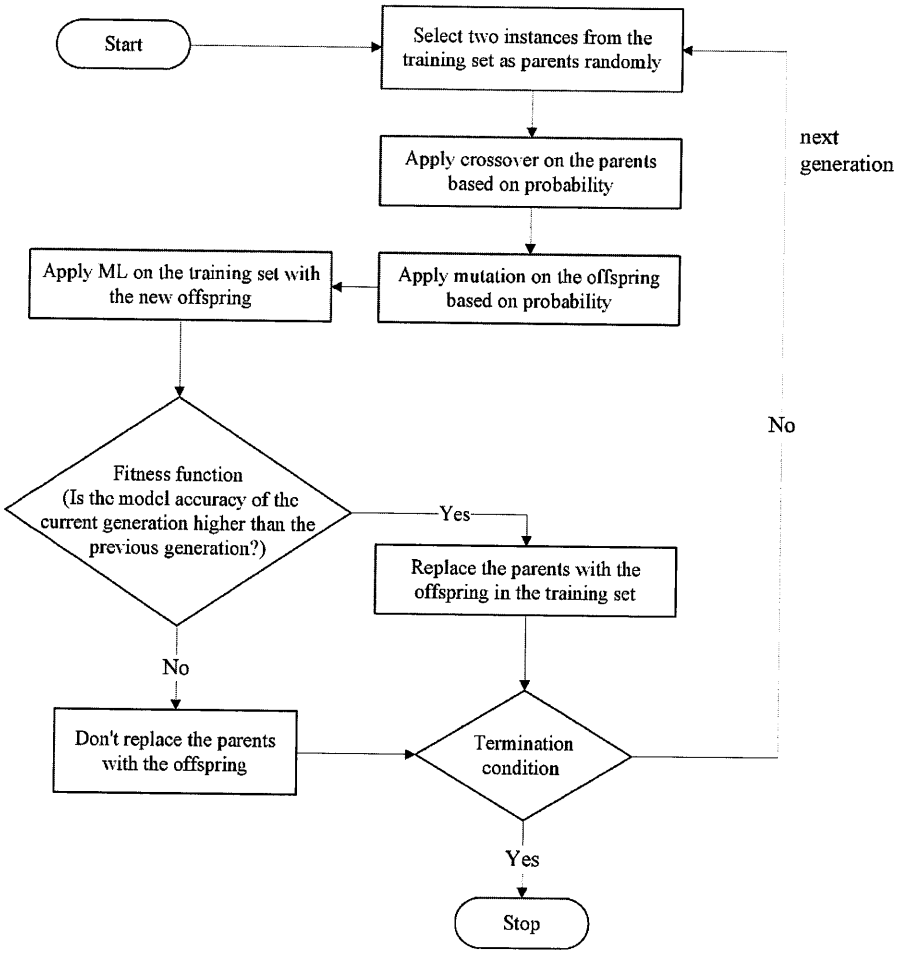


Fig. 3.2: Flowchart of GA phase in GADIC

The steps of GA phase in the proposed algorithm are as follow:

1. Set a termination condition for GA to stop and return the final results. In the proposed algorithm, the termination condition is satisfied when a number of generations g is reached. This number is specified before applying any GA operators.
2. Select two instances randomly from the training set as parents to perform the GA operators on them.
3. Apply single point crossover on the selected parents. This type of crossover is carried out based on two main parameters:
   a. Crossover probability which is a parameter that regulates the possibility that crossover will happen between the parents. It is more probable that crossover between the parents will take place if the crossover probability is set to a high number and vice versa (Datta, 2023).
   b. Crossover point which is a location selected randomly in the instance. The two parents will be cut at that point and the data to the right of this location are swapped between the two parents to produce two offspring (Savio & Chakraborty, 2019).

In this proposed algorithm, the crossover will be applied only on the independent attributes. The class labels will remain untouched. The crossover is performed at the selected crossover point only if it satisfies the crossover probability criteria. Figure 3.3 illustrates an example of the crossover operator applied on one of the experimented datasets showing the class labels of those chromosomes.

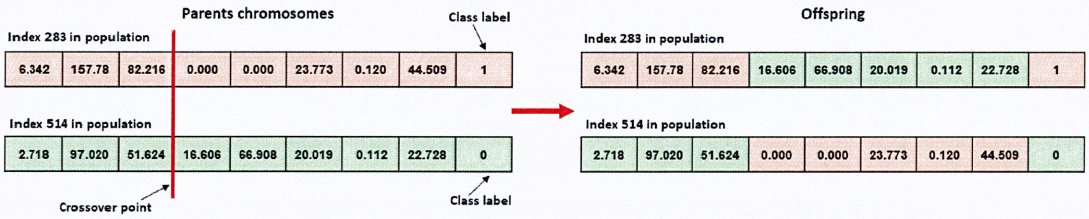


Fig. 3.3: Example of single point crossover operator applied on Pima Indian diabetes dataset

4. Apply mutation on the offspring produced in the previous step. The mutation is carried out based on three main parameters:
   a. Mutation probability which is a parameter that regulates the possibility that an instance will be subjected to the mutation process. The probability should ideally be set to a small number so that only a very small portion of the population mutates in each generation (Datta, 2023).
   b. Genes to be mutated in the offspring. A mutation rate that is specified at the beginning determines how many genes will be mutated while the locations of these genes are selected randomly.
   c. Mutation value which is a value chosen at random between 0 and 0.1. This value will be added to the values of genes that were selected in step b.

It is worth mentioning that the mutation is performed only if it satisfies the mutation probability criteria. Figure 3.4 illustrates an example of the mutation operator applied on one of the experimented datasets.



Fig. 3.4: Example of mutation operator applied on Pima Indian diabetes dataset

5. Apply the fitness function to the produced training set after crossover and mutation operators. The fitness function in the proposed algorithm is designed to assess whether applying GA operators on the reformatted dataset will yield to an increase in the performance or not. Only if the fitness function returns a True value, the offspring will replace their parents in the initial dataset. Otherwise, they will be discarded. The fitness function process is as follows:
   a. At the first generation, the fitness function will return True if the model accuracy after the crossover and mutation operators is higher than the initial model accuracy. If not, it will return False.
   b. At the subsequent generations, the fitness function will return True if the model accuracy after the crossover and mutation operators in that generation is better than the previous generation's model accuracy. Otherwise, it will return False.

3.2.4. Averaging Each Instance in the Testing Set

As a precaution against potential failure brought on by the applied GA in the training phase, the testing set is mapped in a way that is similar to that of the training set. This is done by averaging the attributes values of each instance in the testing test using the attributes values of their similar instances in the training set. The following are the steps of the testing phase. These steps will be repeated for each instance in the testing set.

1. For instance n in the testing set, find the h instances in the training set that are closest to n, where h is the chosen number of the similar instances from the training set. The value of h will be

selected based on trial and error to find the best value for it. Euclidean Distance function will be used to find the h similar instances from the training set to instance n. Such distance is calculated using Equation (3.7):

$$d_{i,j} = \sqrt{\sum_{p=1}^{k}(x_{ip} - x_{jp})^2}, \quad (3.7)$$

where,

p=1, …, k are the attributes,

x_{ip} is the input value of attribute p in instance i,

x_{jp} is the input value of attribute p in instance j.

2. Calculate the average value for each attribute in the similar instances independently. Equation (3.8) is used to find the average $avgAtt_i$:

$$avgAtt_i = \frac{v_1 + v_2 + \cdots + v_h}{h}, \quad (3.8)$$

where,

v is the input value of the specified attribute in the selected instances.

3. The average values $avgAtt_i$ obtained in the prior step for each attribute will be used in place of the values of instance n in the testing set.

3.3. Tested Classifiers

As stated in the literature review, some classifiers consistently outperformed others and achieved the highest increase in performance when combined with OM. On that basis, the classifiers with lower performance have been selected to be tested in this research to examine the effect of GADIC on them. The proposed GADIC algorithm are tested

on five ML classifiers namely, SVM, KNN, LR, DT, and NB. Detailed descriptions of these classifiers were provided in section 2.1.1.

3.4. Performance Metrics

This section provides a brief explanation of the performance metrics that are used to assess the proposed algorithm. Model evaluation is the practice of employing several performance metrics to assess the performance of ML model. In the early stages of research, model evaluation is crucial to determine a model's effectiveness (Domino, 2023). In this research, four well-known performance metrics are utilized to evaluate the effectiveness of the proposed algorithm and compare the results. These metrics are accuracy, precision, recall, and f1-score.

3.4.1. Accuracy

Accuracy is a metric of how well a model is able to correctly classify data. It is the proportion of correct predictions to the total number of predictions (Bressler , 2022). The accuracy is calculated using Equation (3.9):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \tag{3.9}$$

where,

TP is the number of true positives,

TN is the number of true negatives,

FP is the number of false positives,

TN is the number of true negatives.

3.4.2. Precision

Precision is a metric of how many positive predictions of a model are actually correct (Brownlee, 2020a). It measures the proportion of true positives to the sum of both true and false positives. The precision is calculated using Equation (3.10):

$$Precision = \frac{TP}{TP+FP}, \tag{3.10}$$

where,

TP is the number of true positives,

FP is the number of false positives.

3.4.3. Recall

Recall is a metric of how well the model is able to identify true positives (Huilgol, 2023). It is the proportion of true positives to the sum of true positives and false negatives. The recall is calculated using Equation (3.11):

$$Recall = \frac{TP}{TP+FN}, \tag{3.11}$$

where,

TP is the number of true positives,

FN is the number of false negatives.

3.4.4. F1-Score

The F1-score metric takes the two matrices, precision and recall, and combines them into a single metric. It strikes a balance between recall and precision (Abhigyan, 2020). F1-score is calculated using Equation (3.12):

$$F1 = 2 \times \frac{precision \times recall}{precision+recall}, \tag{3.12}$$

Chapter Four

Experiments, Results and Discussion

4.1. Datasets Used

4.2. Experiments

4.3. Results

4.4. Discussion

Chapter Four

Experiments, Results and Discussion

4.1. Datasets Used

In this section, the datasets which are used in the experiments for this research are described. We shall provide a high-level overview of these datasets in this research since they are open-source datasets and have been thoroughly described and examined in prior studies.

4.1.1. Data Sources

The datasets used in this research were obtained from Kaggle and UCI ML repository, two well-known open-source repositories (Kaggle, n.d.; UCI, n.d.). The proposed algorithm experiments were run on seven datasets. Table 4.1 lists the datasets used in this research along with their sources. All datasets were obtained in Comma Separated Values (CSV) format.

Table (4.1): Datasets used in the experiments

Source	Dataset name	About dataset
UCI	Cleveland Heart Disease	Predict whether a patient has heart disease or not
	Indian Liver Patient	Predict whether a patient has liver disease or not
	Pima Indian Diabetes	Predict whether a patient has diabetes or not

Table (4.1): Datasets used in the experiments (cont.)

Source	Dataset name	About dataset
Kaggle	Employee Future Prediction	Predict whether some employee would leave the company in the next 2 years
	Telecom Churn Prediction	Predict customer churn in telecom company
	Bank Customer Churn	ABC bank customer data for predicting customer churn
	Tech Students	Group students according to their developer profiles

4.1.2. Datasets Description

The chosen datasets for the experiments in this research ranged in size from a few hundred to 20,000 instances. All of them are classification datasets with either binary or multiclass class labels. Input values of various datatypes are present in these datasets. The descriptive analysis of the seven datasets utilized in the experiments is displayed in Table 4.2. The class label is not included with the number of attributes given in the table.

With the exception of the tech students dataset, which includes six class labels, all datasets have binary class labels. Figure 4.1 and Figure 4.2 show the distribution of the class labels of the used datasets. As it is apparent from the figures, the only datasets with balanced or almost equal classes are Cleveland heart disease and tech students. Imbalances exist in the other five datasets.

Table (4.2): Description of the datasets used in the experiments

Dataset	No. of instances	No. of attributes	Datatypes of attributes
Cleveland Heart Disease	297	13	• Integer • Real • Categorical
Indian Liver Patient	583	10	• Integer • Real • Categorical
Pima Indian Diabetes	768	8	• Integer • Real
Employee Future Prediction	4,653	8	• Integer • Real • Categorical
Telecom Churn Prediction	7,032	20	• Integer • Real • Categorical
Bank Customer Churn	10,000	11	• Integer • Real • Categorical
Tech Students	20,000	15	• Integer • String

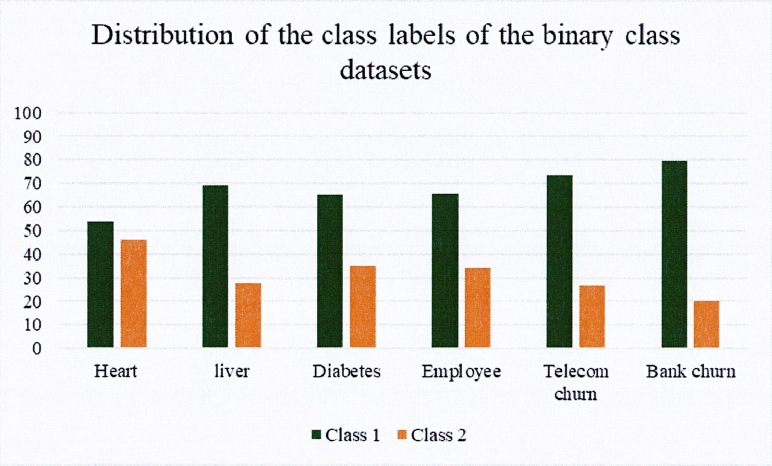


Fig. 4.1: Distribution of binary class labels datasets

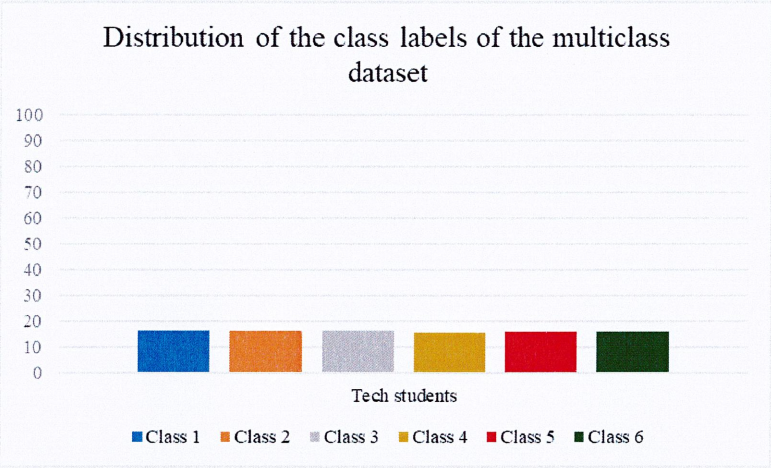


Fig. 4.2: Distribution of multiclass labels of tech students dataset

4.1.3. Data Pre-Processing

Data preparation is the first and most important step before deploying an ML model. It involves cleaning up the raw data to make it appropriate for an ML model. Pre-processing is essential because authentic data frequently has noise, missing or duplicate values, or in a format that makes it impossible to build ML models on it directly (javatpoint, n.d.).

The pre-processing steps employed in this research include the elimination of irrelevant attributes, the removal of missing and duplicate data, and the application of encoding techniques on the categorical attributes. For the imbalanced datasets, data balancing was tried. Yet, when conventional ML classifiers were used on the balanced datasets, the results were worse than those obtained from unbalanced datasets. Therefore, data balancing was disregarded in all models. The data pre-processing was kept at its minimal level to maintain the slightest degree of data manipulation. All of the employed datasets were subjected to the same pre-processing steps, which were performed in Jupyter notebook using python libraries.

First, the datasets were explored to remove any irrelevant attributes that are not needed and will not contribute to the building of the ML models. Among the seven datasets, three of them had irrelevant attributes. The attributes that were eliminated from the datasets are shown in Table 4.3.

Table (4.3): Irrelevant attributes in the used datasets

Dataset	Irrelevant attributes
Telecom Churn Prediction	• Customer ID
Bank Customer Churn	• Customer ID
Tech Students	• Serial • Name • User ID

Following that, duplicate records and records containing missing values were removed from the datasets. The removal was carried out using Pandas python library. Table 4.4 demonstrates the number of missing and duplicate values that were removed from each dataset along with the number of instances of each dataset after removing those values.

Table (4.4): Number of missing and duplicate values in the datasets

Dataset	No. of missing values	No. of duplicate records	No. of instances after removal
Cleveland Heart Disease	0	0	297
Indian Liver Patient	5	13	565
Pima Indian Diabetes	0	0	768
Employee Future Prediction	0	1,889	2,764
Telecom Churn Prediction	0	0	7,032
Bank Customer Churn	0	0	10,000
Tech Students	672	0	19,328

Finally, encoding techniques were applied to the categorical attributes in the datasets. For each dataset, one hot encoder was used for the independent attributes whereas label encoder was used for the dependant attribute. This step was carried out using the scikit-learn python library. The datasets were then saved to be used later in implementing GADIC procedures.

4.2. Experiments

The proposed algorithm comprises three phases: data reformatting phase, training phase, and testing phase, as explained in Chapter 3. To find the best parameters that ought to be employed in each phase of the proposed algorithm, extensive experiments were conducted throughout every single phase. This section presents the experiments performed to evaluate the proposed algorithm's performance and compare its findings with those of other conventional classification ML algorithms. Python libraries were used in Jupyter notebook to conduct all experiments and compare the results.

4.2.1. Data Reformatting Phase

After pre-processing the datasets as described in subsection 4.1.3, it was time to put GADIC procedures into action. In the data reformatting phase, the experiments were performed to determine the optimal values for the parameters: random sample size, number of repetitions, and number of intervals. It was observed that 50 for the random sample size, 10 for the number of repetitions, and 4 for the number of intervals were the optimal values for the datasets of sizes up to 5,000 records. Moreover, it was found with experiments that it is better to raise the values of the

parameters with larger datasets in order to more accurately represent their values and nature. As a result, for larger datasets, the parameter values were raised to 100 for the random sample size, 50 for the number of repetitions, and 10 for the number of intervals. The number of repetitions was set to 10, since this would provide a decent depiction of the average changing factor value. Afterwards, all datasets were reformatted based on these parameter values and saved to be used in the following phases.

4.2.2. Data Splitting

It is important to test several data splitting ratios when employing ML algorithms to find the best ratio that will result in the best performance of the algorithm. This is because different data splits can considerably influence the performance of the algorithm. To find the optimal data split for this research, the proposed algorithm was tested with 70/30, 60/40, and 80/20 split ratios. The findings indicated that 70/30 split produced the optimum performance across most datasets. The results were consistent for all tested classifiers in the proposed algorithm. Accordingly, 70/30 data splitting was chosen to be used throughout all experiments in this research. The reformatted datasets were split into training and testing sets using this ratio. The split was performed using train_test_split function in scikit-learn python library. Figure 4.3 provides an illustration of the experiments that were run on one of the datasets to determine the best data split ratio.

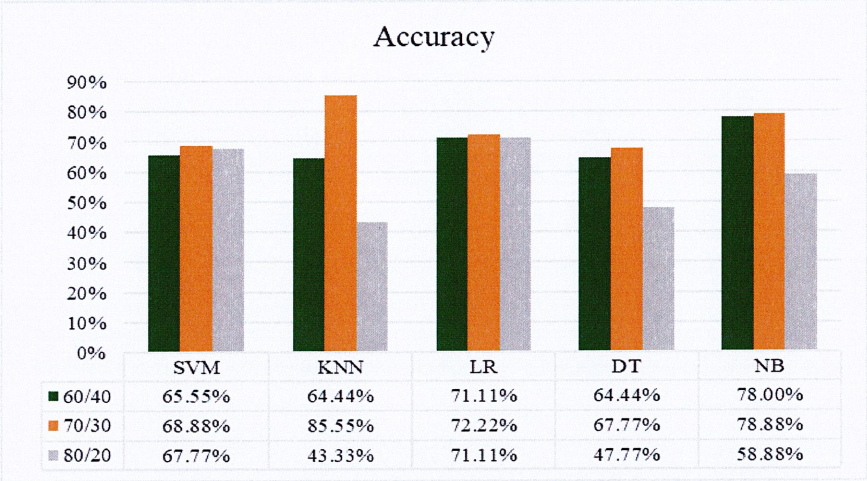


Fig. 4.3: Accuracy of heart dataset using different split ratios

4.2.3. Training Phase

The tested classifiers in the proposed algorithm were trained at this phase by repeatedly passing training sets through GA operators. For each operator, a number of parameters must be specified and employed to obtain the best possible performance from these classifiers. Different parameters values for GA operators were tried in order to find the best values to fine-tune the training set and achieve a better performance.

The completion of 25,000 generations was chosen as the termination condition for the proposed algorithm. This number was selected because, in the experiments that were conducted, a larger number of generations did not lead to any improvements in the performance of the classifiers.

The single point crossover parameters: crossover probability and crossover point have been subjected to experiments to find the most suitable values for them. However, based on a recommendation by Datta (2023) and Savio and Chakraborty (2019) that the mutation probability should be set to a low number, all mutation parameters in this research were set at low values. The mutation rate was set to maximum 0.2 randomly, the mutation probability was set to 0.3, and the mutation value

was set to a random value between 0 and 0.1 for all experiments. Table 4.5 displays the parameter values for crossover point and crossover probability that, according to the experiments, led to the best performance along with the number of generations it took to achieve that result.

Three points should be highlighted in this phase. First, in all experiments, the tested classifiers were solely trained using their default hyperparameters. Second, the proposed algorithm was tested also on normalized datasets, but its performance was inferior to that of the original datasets. Third, both single point crossover and unform crossover were put to the test in the experiments, but single point crossover produced better results, hence it was adopted in the proposed algorithm.

4.2.4. Testing Phase

Following the procedures of the proposed algorithm, the testing sets went through the testing phase to average their instances values based on the similar instances in the training sets once the training phase was finished. The number of similar instances (n) to be used in the Euclidean distance function is the only parameter in this phase. Several values were tried to find the suitable n value to be used in averaging the instances of the testing sets. The SciPy library in python was used to perform the operations of this phase. The n values that led to the best performance in each experiment are given in Table 4.5.

Table (4.5): Parameters resulted with the best performance in GADIC

Dataset	Classifier	Crossover point	Crossover probability	No. of generations it took	n value
Cleveland Heart Disease	SVM	3	0.6	24,768	5
	KNN	3	0.6	13,208	6
	LR	7	0.7	8	5
	DT	4	0.7	6,815	3
	NB	4	0.7	19	3
Indian Liver Patient	SVM	3	0.7	2,951	4
	KNN	4	0.6	23,442	5
	LR	4	0.6	10,531	3
	DT	3	0.6	24,647	3
	NB	3	0.5	1,558	4
Pima Indian Diabetes	SVM	3	0.7	24,595	3
	KNN	4	0.6	5,530	3
	LR	3	0.6	15,570	3
	DT	3	0.7	6,942	3
	NB	3	0.6	15,718	3
Employee Future Prediction	SVM	3	0.5	10,869	4
	KNN	6	0.6	667	7
	LR	7	0.7	5,735	3
	DT	7	0.7	42	3
	NB	7	0.6	5,608	5
Telecom Churn Prediction	SVM	7	0.5	521	5
	KNN	4	0.6	43	5
	LR	11	0.6	1,102	3
	DT	9	0.6	179	3
	NB	7	0.7	118	5
Bank Customer Churn	SVM	3	0.7	2,482	6
	KNN	7	0.7	43	7
	LR	4	0.6	1,481	3
	DT	3	0.7	371	5
	NB	3	0.6	1,694	4

Table (4.5): Parameters resulted with the best performance in GADIC (cont.)

Dataset	Classifier	Crossover point	Crossover probability	No. of generations it took	n value
Tech Students	SVM	6	0.6	30	4
	KNN	5	0.7	7	5
	LR	7	0.6	59	3
	DT	4	0.7	5	3
	NB	6	0.7	14	3

4.3. Results

In this section, a detailed analysis of the results obtained from the conducted experiments is presented. To gauge the efficacy of the proposed algorithm, the datasets were evaluated using the conventional ML classifiers first, then the proposed algorithm was tested on the same datasets to compare the results. The results of the datasets are presented in the following subsections.

4.3.1. Cleveland Heart Disease Dataset

The Cleveland heart disease dataset results using conventional ML classifiers are shown in Table 4.6, along with the results obtained from applying GADIC on it using the same ML classifiers.

Table (4.6): Results of Cleveland heart disease dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
SVM	67.77%	0.68	0.72	0.65	81.11%	0.81	0.83	0.82
KNN	67.77%	0.68	0.68	0.67	90.00%	0.90	0.90	0.90
LR	85.55%	0.86	0.86	0.85	86.66%	0.87	0.87	0.87
DT	77.77%	0.78	0.78	0.78	78.88%	0.79	0.81	0.78
NB	88.88%	0.89	0.89	0.89	87.77%	0.88	0.87	0.88

According to the table, most classifiers demonstrated increases in accuracy and other metrics after applying GADIC, with the exception of NB, which witnessed a slight decrease in accuracy from 88.88% to 87.77%. The NB classifier, with an accuracy of 88.88%, was the best model prior to the application of GADIC. The KNN model, however, demonstrated the highest accuracy and increase from 67.77% to 90.00% after applying GADIC. Figure 4.4 depicts the change in performance in this dataset when applying GADIC.

All classifiers, with the exception of NB, showed improvements in recall and accuracy values, with the KNN classifier showing the most gain, rising from 0.68 to 0.90 for both. Similarly, for f1-score, GADIC improved SVM, KNN, and LR performance, with the KNN showing the largest improvement, rising from 0.67 to 0.90. DT remained constant, however, NB marginally declined.

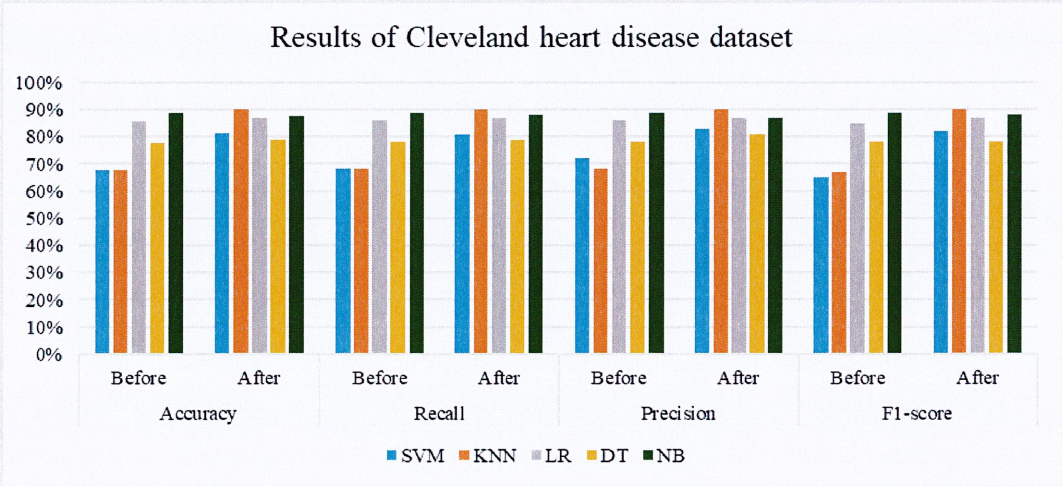


Fig. 4.4: Change in performance in heart dataset after applying GADIC

4.3.2. Indian Liver Patient Dataset

Table 4.7 presents the results of the chosen ML classifiers on the Indian liver patient dataset both before and after applying GADIC. The

findings indicate that after the application of GADIC, all classifiers experienced a substantial improvement in accuracy, recall, precision, and f1-score with the exception of the LR and NB models, which experienced a decrease in precision value.

Table (4.7): Results of Indian liver patient dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	**Recall**	**Precision**	**F1**	**Accuracy**	**Recall**	**Precision**	**F1**
SVM	71.17%	0.71	0.51	0.59	84.70%	0.85	0.87	0.79
KNN	68.23%	0.68	0.65	0.66	91.76%	0.92	0.92	0.91
LR	74.70%	0.75	0.78	0.67	82.94%	0.83	0.69	0.75
DT	61.17%	0.61	0.6	0.6	77.64%	0.78	0.79	0.78
NB	56.47%	0.56	0.81	0.56	77.05%	0.77	0.69	0.72

The KNN classifier was the one which showed the most improvement, with the greatest increases in accuracy, precision, recall, and f1-score. On the other hand, the LR classifier had the lowest increase in performance. The amount of increase in accuracy varied from 8.4% in LR to 23.5% in KNN. The performance variance for this dataset after using GADIC is shown in Figure 4.5. It was observed that there has been a noticeable improvement in most of the metrics for all classifiers when applying GADIC as stated earlier. Nevertheless, only the precision for LR and NB decreased from 0.78 to 0.69 and 0.81 to 0.69 respectively. The increase in recall and precision metrics ranged from 0.08 in LR's recall to 0.36 in SVM's precision. The highest increase in f1-score value was 0.25 in KNN classifier.

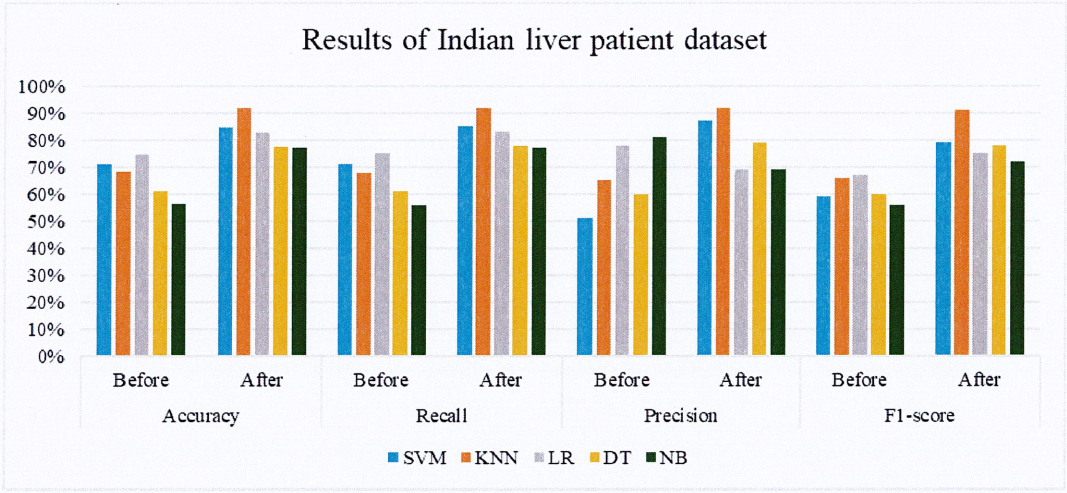


Fig. 4.5: Change in performance in liver dataset after applying GADIC

4.3.3. Pima Indian Diabetes Dataset

Table 4.8 provides the results that were obtained from the metrics used to assess GADIC's efficacy on the Pima Indian diabetes dataset. All classifiers showed a considerable improvement when applying GADIC. However, it is worth noting that the level of improvement differs among the classifiers. For instance, KNN showed the most improvement in all metrics compared to the other classifiers. Conversely, LR showed the least improvement, but the increase is still significant.

In terms of accuracy, KNN got the most improvement in accuracy, increasing from 74.45% to 94.37%. DT also had noticeable improvements in accuracy from 68.83% to 83.11%. Figure 4.6 depicts the change in performance when applying GADIC on this dataset. When it comes to other metrics, KNN, LR, and DT showed consistent improvement across all metrics, with DT having the most notable improvement of them all.

Table (4.8): Results of Pima Indian diabetes dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
SVM	75.75%	0.76	0.75	0.75	85.28%	0.85	0.85	0.85
KNN	74.45%	0.74	0.74	0.74	94.37%	0.94	0.94	0.94
LR	77.05%	0.77	0.77	0.76	84.84%	0.85	0.85	0.84
DT	68.83%	0.69	0.69	0.69	83.11%	0.83	0.83	0.83
NB	77.48%	0.77	0.77	0.77	85.71%	0.86	0.86	0.85

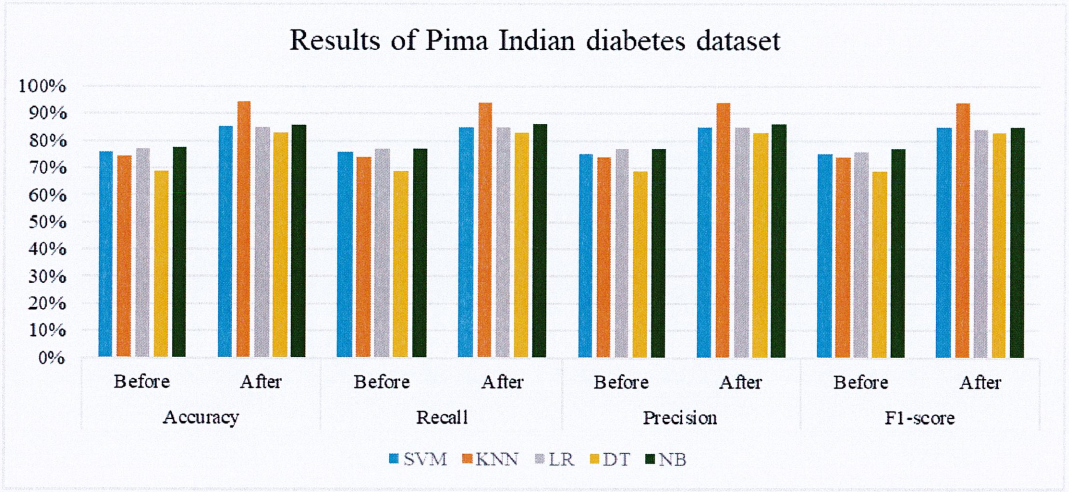


Fig. 4.6: Change in performance in diabetes dataset after applying GADIC

4.3.4. Employee Future Prediction Dataset

The results of the selected ML classifiers on the employee future prediction dataset before and after GADIC deployment are shown in Table 4.9. The application of GADIC led to an increase in accuracy, recall, precision, and f1-score for all classifiers. KNN has improved the most in terms of accuracy, going from 78.15% to 87.1%, followed by NB, which went from 68.98% to 76%. The SVM, on the other hand, exhibited the smallest gain in accuracy, while the DT and LR classifiers showed a moderate improvement. Figure 4.7 illustrates the variation in performance improvements across all models.

Table (4.9): Results of employee future prediction dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
SVM	65.61%	0.66	0.43	0.52	69.69%	0.7	0.49	0.57
KNN	78.15%	0.78	0.78	0.77	87.10%	0.87	0.88	0.87
LR	70.84%	0.71	0.7	0.68	76.00%	0.76	0.76	0.73
DT	81.73%	0.82	0.81	0.81	86.74%	0.87	0.87	0.87
NB	68.98%	0.69	0.68	0.68	76.00%	0.76	0.75	0.74

In most cases, the recall, precision, and f1-score had improved relatively consistently in all classifiers except SVM, where there was a variation in the level of improvement, as is apparent when comparing the results before and after applying GADIC. In those metrics, KNN's increase outpaced all other models, whereas SVM's increase was the smallest with nearly 0.1 and 0.05 increase respectively for each metric.

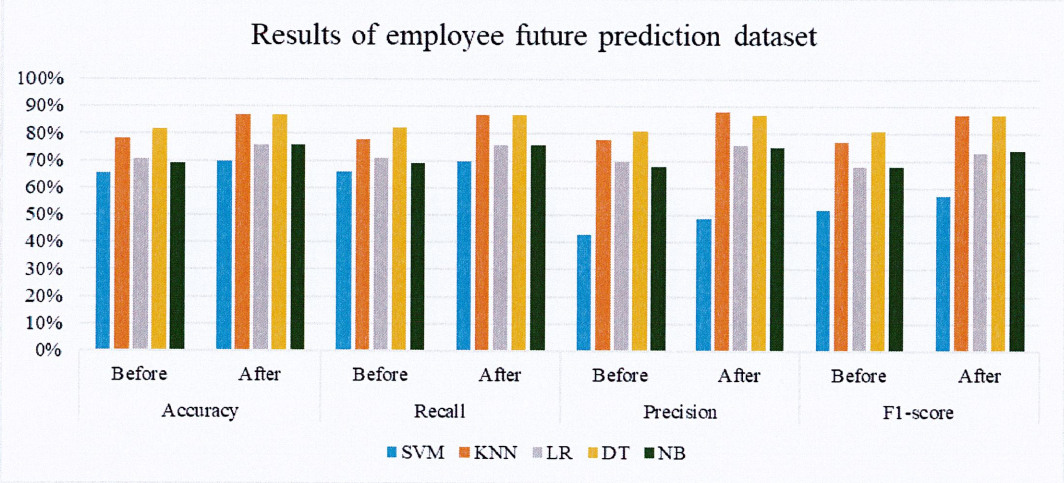


Fig. 4.7: Change in performance in employee dataset after applying GADIC

4.3.5. Telecom Churn Prediction Dataset

The findings from the evaluation of GADIC's performance on the telecom churn prediction dataset are presented in Table 4.10. By

comparing the results of the metrics before and after applying GADIC, we can notice modest increases in the performance of all models.

Table (4.10): Results of telecom churn prediction dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
SVM	73.41%	0.73	0.54	0.73	81.99%	0.82	0.67	0.74
KNN	76.91%	0.77	0.76	0.76	91.23%	0.91	0.91	0.91
LR	79.85%	0.8	0.79	0.79	85.40%	0.85	0.85	0.85
DT	72.93%	0.73	0.73	0.73	79.90%	0.81	0.82	0.8
NB	75.87%	0.76	0.79	0.77	76.68%	0.77	0.87	0.79

With the deployment of GADIC, all classifiers demonstrated performance improvements with different amount as illustrated in Figure 4.8. KNN had the highest increase (14.32%), SVM had the second highest increase (8.58%), while NB had the lowest increase (0.81%). However, the change in f1-score was not as much as the change in accuracy for most classifiers. Apart from KNN, which showed an increase of 0.15 in the f1-score, all models had a minor improvement in the f1-score, particularly SVM and NB, where the increases were 0.01 and 0.02, respectively.

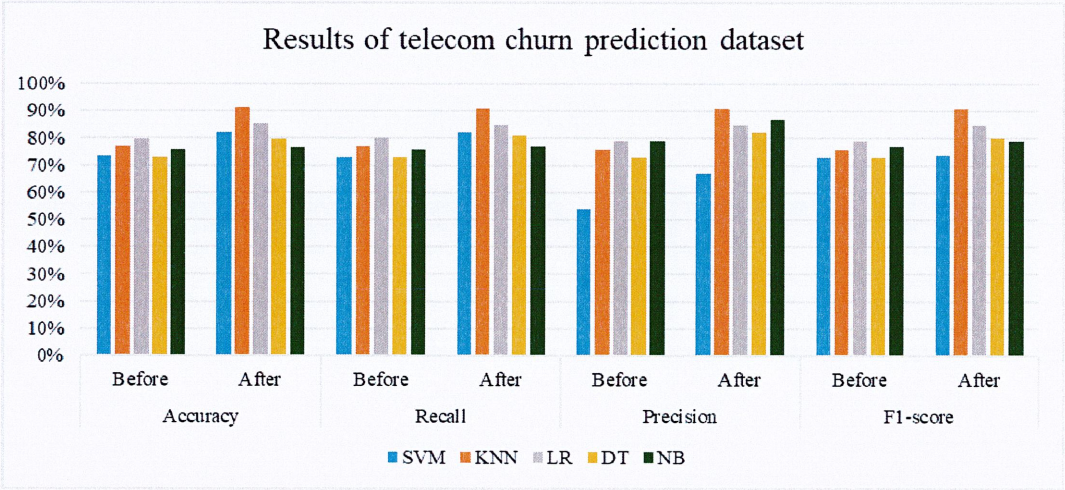


Fig. 4.8: Change in performance in telecom dataset after applying GADIC

4.3.6. Bank Customer Churn Dataset

The findings of the chosen ML classifiers both before and after GADIC implementation on the bank customer churn dataset are shown in Table 4.11. Moreover, Figure 4.9 depicts the change in performance when applying GADIC on this dataset. The comparison of the accuracy, recall, precision, and f1-score metrics reveals that, on average, the classifiers' performance increased after applying GADIC. The KNN classifier specifically achieved the largest increase in all metrics. Its accuracy increased by 15.5%, and its f1-score increased from 0.71 to 0.91. Nevertheless, DT showed the least improvement across all classifiers, particularly in terms of accuracy, recall, and f1-score values with its accuracy increasing by 0.77%, recall by 0.01, precision by 0.12, and f1-score by 0.05. The accuracy and f1-score metrics showed consistent improvement for the other classifiers, SVM, LR, and NB. They all improved with an average accuracy increase of 10% and f1-score increase range of 0.11 to 0.14.

Table (4.11): Results of bank customer churn dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
SVM	79.63%	0.8	0.63	0.71	89.86%	0.9	0.81	0.85
KNN	76.46%	0.76	0.69	0.71	91.96%	0.92	0.91	0.91
LR	78.70%	0.79	0.72	0.72	89.16%	0.89	0.8	0.84
DT	78.13%	0.78	0.78	0.78	78.90%	0.79	0.9	0.83
NB	78.00%	0.78	0.71	0.72	88.70%	0.89	0.79	0.83

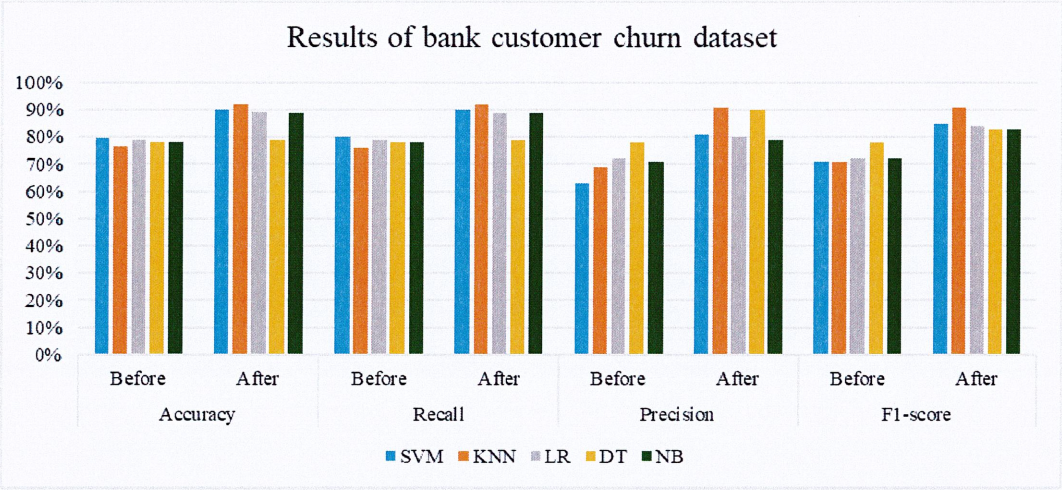


Fig. 4.9: Change in performance in bank dataset after applying GADIC

4.3.7. Tech Students Dataset

The results of the selected ML classifiers on the tech students dataset before and after GADIC implementation are presented in Table 4.12. When comparing the results of each classifier, it can be seen that SVM, KNN, and LR all showed an improvement in accuracy after the application of GADIC, with KNN demonstrating the highest increase in accuracy from 74.35% to 87.44% followed by SVM and LR where they increased from 83.77% and 69.94% to 87.70% and 73.34% respectively.

Table (4.12): Results of tech students dataset

ML	Before applying GADIC				After applying GADIC			
	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
SVM	83.77%	0.84	0.84	0.84	87.70%	0.88	0.88	0.88
KNN	74.35%	0.74	0.74	0.74	87.44%	0.87	0.88	0.87
LR	69.94%	0.7	0.7	0.7	73.34%	0.73	0.73	0.73
DT	76.40%	0.76	0.76	0.76	76.61%	0.77	0.77	0.77
NB	71.44%	0.71	0.71	0.71	72.46%	0.72	0.73	0.72

Figure 4.10 shows that while the accuracy of the NB classifier slightly increased, the accuracy of the DT model remained roughly the

same with merely 0.21% increase. With the exception of KNN, the other classifiers exhibited minor improvements in recall, precision, and f1-score. For these metrics, they only increased at very low values, between 0.01 and 0.04 specially in DT and NB. Interestingly, the accuracy and other metrics for the LR and NB classifiers just slightly increased after the application of GADIC, with no noticeable improvement in performance.

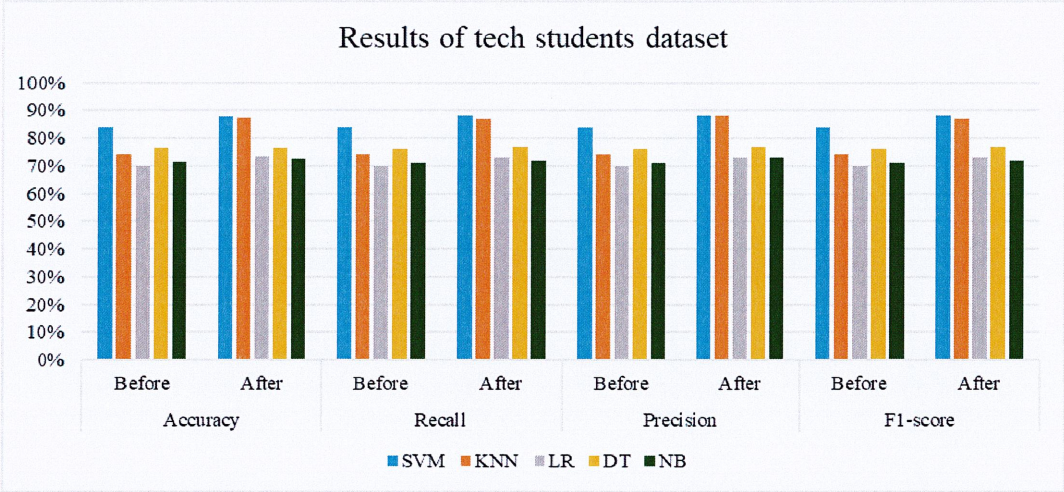


Fig. 4.10: Change in performance in tech student dataset after applying GADIC

4.4. Discussion

The proposed algorithm's findings were obtained after conducting extensive experiments to find the best parameters for each phase. The experiments revealed that, in most cases, DT and NB classifiers often stopped improving after a limited number of generations in the proposed algorithm, and the performance did not improve more with the increase in the number of generations. In contrast, LR took more generations to get its best results, but even then, the performance gain is less than that of other models. This indicates that these classifiers are not significantly influenced by the proposed algorithm.

The extent to which the proposed algorithm enhanced the performance of the chosen classifiers in terms of accuracy is shown in Table 4.13. The table shows that all classifiers demonstrated improvements in their accuracy in all datasets by various amounts except for NB in the Cleveland heart disease dataset. This case is the only one where the proposed algorithm did not contribute positively to the classifier performance despite the massive experiments that were performed.

Table (4.13): Increase in classifiers accuracy using GADIC

Dataset	SVM	KNN	LR	DT	NB
Cleveland Heart Disease	13.34%	22.23%	1.11%	1.11%	-1.11%
Indian Liver Patient	13.53%	23.53%	8.24%	16.47%	20.58%
Pima Indian Diabetes	9.53%	19.92%	7.79%	14.28%	8.23%
Employee Future Prediction	4.08%	8.95%	5.16%	5.01%	7.02%
Telecom Churn Prediction	8.58%	14.32%	5.55%	6.97%	0.81%
Bank Customer Churn	10.23%	15.50%	10.46%	0.77%	10.70%
Tech Students	3.93%	13.09%	3.40%	0.21%	1.02%

According to the table, KNN had the highest increase in performance when GADIC was applied, outperforming the other models in all datasets with remarkable and substantial improvement. SVM had the second highest increase in performance, while LR was the least affected classifier by GADIC by a small difference than NB and DT. LR's lowest increase is likely due to the nature and assumptions of the LR algorithm itself. LR classifier performance can be impacted by the presence of imbalanced data (Salas-Eljatib et al., 2018). This can restrict its potential for improvement. The average improvement in accuracy for each model is shown in Figure 4.11. These findings were consistent for both accuracy and f1-score, as observed in section 4.3.

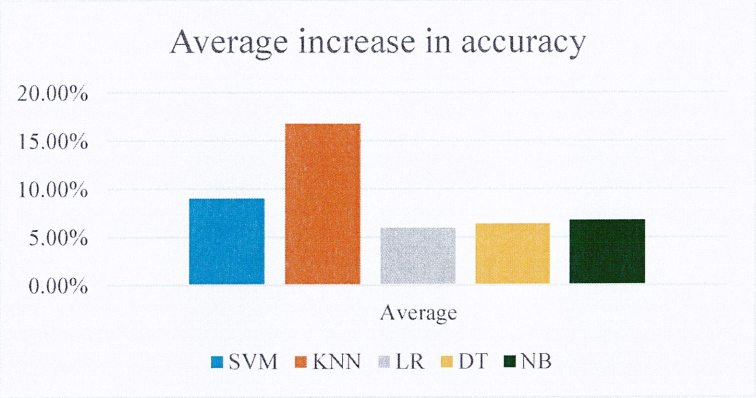


Fig. 4.11: Average increase in accuracy per model

It can be concluded that the performance of the models was significantly influenced by the nature of the data. Greater improvements in performance after applying GADIC indicate that the types and nature of the data have a major influence on the results. On the contrary, lower improvements indicate that the data itself has little impact and that there may be other factors influencing the performance.

Answers to the research questions "How does the proposed algorithm contribute to the classifiers' performance?" and "How effective is the proposed algorithm in comparison to the conventional classification algorithms?" were presented in the research's results and discussion sections. The proposed algorithm showed excellent outcomes and a strong potential for improvement in the performance of ML classifiers when applied on various datasets of varying types and sizes. Moreover, the effectiveness of the proposed algorithm over the conventional ML classifiers was demonstrated by the improvement in the performance of classifiers evaluated by the performance metrics in section 4.3. The proposed algorithm exhibited superior performance compared to conventional ML classifiers, leading to noticeable improvements in the performance following the application of the proposed algorithm.

Chapter Five

Conclusion and Recommendations

5.1. Conclusion

5.2. Recommendations and Future work

Chapter Five
Conclusion and Recommendations

5.1. Conclusion

Enhancing the performance of the ML is essential to address the growing demand for accurate and reliable predictions in various industries and fields. To ensure that the models produce reliable predictions, it is essential to increase the accuracy and efficiency of the algorithms as they are being increasingly used in critical fields. Nevertheless, some classifiers are not able to increase their performance due to issues with the nature of the data being used.

In this research, a novel classification algorithm, namely GADIC, that is based on GA and DI reformatting was proposed. GADIC was intended to be able to overcome some data-related issues that could prohibit the classifiers from improving their performance and hence increasing the performance of the classifiers. To assess its capability to enhance the performance of the classifiers, GADIC was tested on five ML classifiers using seven open-source datasets with various sizes and types. These classifiers are SVM, KNN, LR, DT, and NB.

The findings of the experiments in this research demonstrated that GADIC improved the performance of the vast majority of datasets and classifiers except NB in Cleveland heart disease dataset with only about 1%. Additionally, the performance of GADIC was evaluated against the performance of the conventional ML classifiers. For each dataset, the classifiers' performance improved to a different extent. In terms of accuracy, KNN resulted with the most performance improvement when

comparing it with the conventional ML classifiers when applying GADIC, with an average increase of 16.79%, followed by SVM with an average increase of 9.03%. The average increases of performance of the other three classifiers were 5.96%, 6.4% and 6.75% for LR, DT, and NB respectively with LR showing the least improvement among the classifiers. These results are in line with those from the f1-score measure.

Given that there was an improvement in the performance of the classifiers when GADIC was applied, it can be inferred that the models' performance was relatively influenced by the nature of the data. Based on the obtained results, the research questions can be answered. Since the proposed algorithm proved its effectiveness in enhancing the performance of the tested classifiers, we can draw a conclusion that the proposed algorithm can considerably boost the classifiers' performance and surpass the conventional ML classifiers for datasets of different types and sizes.

5.2. Recommendations and Future Work

The application of the proposed algorithm in this research was limited to open-source datasets and only five conventional ML classifiers. In light of these limitations, it is recommended that future studies consider expanding the scope of the research by testing the proposed algorithm on datasets with larger sizes and other types. Additionally, it is recommended to gauge the proposed algorithm's potential for enhancing the performance of other conventional ML classifiers as well as DL models. Finally, future work may include refining the proposed algorithm by experimenting with and integrating different parameters to optimize the proposed algorithm's steps and boost the potential of improving the performance of the classifiers that are least affected by it.

References

Abhigyan. (2020, March 17). *Calculating Accuracy of an ML Model.* Analytics Vidhya. https://medium.com/analytics-vidhya/calculating-accuracy-of-an-ml-model-8ae7894802e

Alam, T., Qamar, S., Dixit, A., & Benaida, M. (2020). Genetic Algorithm: Reviews, Implementations, and Applications. *International Journal of Engineering Pedagogy, 10,* 57-77. https://doi.org/10.36227/techrxiv.12657173

Almufti, S. M., Zebari, A. Y., & Omer, H. K. (2019). A comparative study of particle swarm optimization and genetic algorithm. *Journal of Advanced Computer Science & Technology, 8*(2), 40-45.

Aly, W. M. (2016). A New Approach for Classifier Model Selection and Tuning Using Logistic Regression and Genetic Algorithms. *Arabian Journal for Science and Engineering, 41,* 5195–5204. https://doi.org/10.1007/s13369-016-2223-2

Ashri, S. E., El-Gayar, M. M., & El-Daydamony, E. M. (2021). HDPF: Heart Disease Prediction Framework Based on Hybrid Classifiers and Genetic Algorithm. *IEEE Access, 9,* 146797-146809. https://doi.org/10.1109/ACCESS.2021.3122789

Azad, C., Bhushan, B., Sharma, R., Shankar, A., & Singh, K. K. (2022). Prediction model using SMOTE, genetic algorithm and decision tree (PMSGD) for classification of diabetes mellitus. *Multimedia Systems,* 1289–1307. https://doi.org/10.1007/s00530-021-00817-2

Azam, M. S., Rahman, A., Iqbal, S. M., & Ahmed, M. T. (2020). Prediction of Liver Diseases by Using Few Machine Learning Based Approaches. *Australian Journal of Engineering and*

Innovative Technology, 2(5), 85-90. https://doi.org/10.34104/ajeit.020.085090

Bains, K. K. (2020, October 10). *Regularization, Optimization & Hyperparameter Tuning*. Medium. https://medium.com/neuron-ai/regularization-optimization-hyperparameter-tuning-4f80115a9e0c

Battineni, G., Sagaro, G. G., Nalini, C., Amenta, F., & Tayebati, S. K. (2019). Comparative Machine-Learning Approach: A Follow-Up Study on Type 2 Diabetes Predictions by Cross-Validation Methods. *Machines, 7*(4).

Birjais, R., Mourya, A. K., Chauhan, R., & Kaur, H. (2019). Prediction and diagnosis of future diabetes risk: a machine learning approach. *SN Applied Sciences, 1*.

Bressler , N. (2022, November 23). *How to Check the Accuracy of Your Machine Learning Model.* deep checks. https://deepchecks.com/how-to-check-the-accuracy-of-your-machine-learning-model

Brownlee, J. (2020a, January 3). *How to Calculate Precision, Recall, and F-Measure for Imbalanced Classification*. Machine Learning Mastery. https://machinelearningmastery.com/precision-recall-and-f-measure-for-imbalanced-classification/

Brownlee, J. (2020b, August 20). *How to Choose a Feature Selection Method For Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

Brownlee, J. (2021a, October 12). *How to Choose an Optimization Algorithm.* Machine Learning Mastery. https://machinelearningmastery.com/tour-of-optimization-algorithms/

Brownlee, J. (2021b, October 12). *Why Optimization Is Important in Machine Learning*. Machine Learning Mastery. https://machinelearningmastery.com/why-optimization-is-important-in-machine-learning/

Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing, 408*, 189-215. https://doi.org/10.1016/j.neucom.2019.10.118

Chen, H., Chen, J., & Ding, J. (2021). Data Evaluation and Enhancement for Quality Improvement of Machine Learning. *IEEE Transactions On Reliability, 70*(2), 831-847.

Chen, W., Chen, Y., Tsangaratos, P., Ilia , I., & Wang, X. (2020). Combining Evolutionary Algorithms and Machine Learning Models in Landslide Susceptibility Assessments. *Remote Sensing, 12*(23). https://doi.org/10.3390/rs12233854

Cheng, D., Zhang, S., Deng, Z., Zhu, Y., & Zong, M. (2014). kNN Algorithm with Data-Driven k Value. *International Conference on Advanced Data Mining and Applications*, 499–512.

Choubey, D. K., Paul , S., Kumar, S., & Kumar, S. (2016). Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection. *The International Conference on Communication and Computing Systems*, 451-455. https://doi.org/10.1201/9781315364094-82

Connelly, L. (2020). Logistic Regression. *MEDSURG Nursing, 29*(5), 353-354.

Dahiya, A., & Sangwan, S. (2018). Literature Review on Genetic Algorithm. *International Journal of Research, 5*(16), 1142-1146.

Damanik, I. S., Windarto, A. P., Wanto, A., Poningsih, Andani, S. R., & Saputra, W. (2019). Decision Tree Optimization in C4.5 Algorithm Using Genetic Algorithm. *The International Conference on Computer Science and Applied Mathematic, 1255*. https://doi.org/10.1088/1742-6596/1255/1/012012

Datta, S. (2023, February 2). *Genetic Algorithms: Crossover Probability and Mutation Probability*. Baeldung. https://www.baeldung.com/cs/genetic-algorithms-crossover-probability-and-mutation-probability

Demir, S., & Şahin, E. K. (2022). Liquefaction prediction with robust machine learning algorithms (SVM, RF, and XGBoost) supported by genetic algorithm-based feature selection and parameter optimization from the perspective of data processing. *Environmental Earth Sciences, 81*. https://doi.org/10.1007/s12665-022-10578-4

Doewes, R. I., Nair, R., & Sharma, T. (2021). Diagnosis of COVID-19 through blood sample using ensemble genetic algorithms and machine learning classifier. *World Journal of Engineering*. https://doi.org/10.1108/WJE-03-2021-0174

Domino. (2023). *Model Evaluation*. DOMINO. https://www.dominodatalab.com/data-science-dictionary/model-evaluation

Farid, A. A., Selim, G. I., & Khater, H. A. (2020). A Composite Hybrid Feature Selection Learning-Based Optimization of Genetic Algorithm For Breast Cancer Detection. *Preprints*. https://doi.org/10.20944/preprints202003.0298.v1

Forbes Technology Council. (2019, September 11). *15 Smart Ways Machine Learning Helps Businesses And Entrepreneurs*. Forbes.

https://www.forbes.com/sites/forbestechcouncil/2019/09/11/15-smart-ways-machine-learning-helps-businesses-and-entrepreneurs/

Fujo, S. W., Subramanian, S., & Khder, M. A. (2022). Customer Churn Prediction in Telecommunication Industry Using Deep Learning. *Information Sciences Letters, 11*(1), 185–198. https://doi.org/10.18576/isl/110120

Garg, A., Sharma, B., & Khan, R. (2021). Heart disease prediction using machine learning techniques. *IOP Conf. Series: Materials Science and Engineering, 1022*. https://doi.org/10.1088/1757-899X/1022/1/012046

Ghosh, M., Raihan, M., Raihan, M., Akter, L., Bairagi, A. K., Alshamrani, S. S., & Masud, M. (2021). A Comparative Analysis of Machine Learning Algorithms to Predict Liver Disease. *Intelligent Automation & Soft Computing, 30*(3), 917-928. https://doi.org/10.32604/iasc.2021.017989

Haldar, A., Biswas, S., & De, S. (2014). A Review on Genetic Algorithm. *International Journal Of Engineering Sciences & Research Technology, 3*(1), 369-372.

Haldurai, L., Madhubala, T., & Rajalakshmi, R. (2016). A Study on Genetic Algorithm and its Applications. *International Journal of Computer Sciences and Engineering, 4*(10), 139-143.

Hamdia, K. M., Zhuang, X., & Rabczuk, T. (2021). An efficient optimization approach for designing machine learning models based on genetic algorithm. *Neural Computing and Applications, 33*, 1923–1933. https://doi.org/10.1007/s00521-020-05035-x

Hewahi, N. M. (2019). Neural network pruning based on input importance. *Journal of Intelligent & Fuzzy Systems, 37*(2), 2243–2252. https://doi.org/10.3233/JIFS-182544

Huilgol, P. (2023, February 15). *Precision and Recall | Essential Metrics for Data Analysis.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/

Hung, S.-C., Wu, H.-C., & Tseng, M.-H. (2021). Integrating Image Quality Enhancement Methods and Deep Learning Techniques for Remote Sensing Scene Classification. *Applied Science, 11*(24). https://doi.org/10.3390/app112411659

Huseyinov, I., & Okocha, O. (2022). A Machine Learning Approach To The Prediction Of Bank Customer Churn Problem. *International Informatics and Software Engineering Conference.* https://doi.org/10.1109/IISEC56263.2022.9998299

Jabor, A. H., & Ali, A. H. (2019). Dual Heuristic Feature Selection Based on Genetic Algorithm and Binary Particle Swarm Optimization. *Journal of University of Babylon for Pure and Applied Sciences, 27*(1), 171-183.

javatpoint. (n.d.). *Data Preprocessing in Machine learning*. javatpoint. https://www.javatpoint.com/data-preprocessing-machine-learning

Jijo, B. T., & Abdulazeez, A. M. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends, 2*(1), 20–28. https://doi.org/10.38094/jastt20165

Kaggle. (n.d.). Kaggle. https://www.kaggle.com/

Kanwal, S., Rashid, J., Nisar, M. W., Kim, J., & Hussain, A. (2021). An Effective Classification Algorithm for Heart Disease Prediction with Genetic Algorithm for Feature Selection. *Mohammad Ali Jinnah University International Conference on Computing.* https://doi.org/10.1109/MAJICC53071.2021.9526242

Kaseb, Z., & Rahbar, M. (2022). Towards CFD-based optimization of urban wind conditions: Comparison of Genetic algorithm, Particle Swarm Optimization, and a hybrid algorithm. *Sustainable Cities and Society, 77.*

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80,* 8091–8126. https://doi.org/10.1007/s11042-020-10139-6

Khanam, J. J., & Foo, S. Y. (2021). A comparison of machine learning algorithms for diabetes prediction. *ICT Express, 7*(4), 432–439.

Khotimah, B. K., Miswanto, M., & Suprajitno, H. (2020). Optimization of Feature Selection Using Genetic Algorithm in Naïve Bayes Classification for Incomplete Data. *International Journal of Intelligent Engineering and Systems, 13*(1), 334-343. https://doi.org/10.22266/ijies2020.0229.31

Książek, W., Gandor , M., & Pławiak , P. (2021). Comparison of various approaches to combine logistic regression with genetic algorithms in survival prediction of hepatocellular carcinoma. *Computers in Biology and Medicine, 134.*

Kumar, G. R., Tirupathaiah, K., & Reddy, B. K. (2019). Client Churn Prediction of Banking and fund industry utilizing Machine Learning Techniques. *International Journal of Computer Sciences and Engineering, 7*(6), 842-846. https://doi.org/10.26438/ijcse/v7i6.842846

Maalouf, M. (2011). Logistic regression in data analysis: An overview. *International Journal of Data Analysis Techniques and Strategies.*

Maleki, N., Zeinali , Y., & Niaki , S. T. (2021). A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. *Expert Systems With Applications, 164.*

Pamina, J., Raja, J. B., Bama, S. S., Soundarya, S., Sruthi, M., Kiruthika, S., . . . Priyanka, G. (2019). An Effective Classifier for Predicting Churn in Telecommunication. *Jour of Adv Research in Dynamical & Control Systems, 11*, 221-229.

Pandey, A., & Jain, A. (2017). Comparative Analysis of KNN Algorithm using Various Normalization Techniques. *International Journal of Computer Network and Information Security, 9*(11), 36-42. https://doi.org/10.5815/ijcnis.2017.11.04

Pawlovsky, A. P., & Matsuhashi, H. (2017). The Use of a Novel Genetic Algorithm in Component Selection for a kNN Method for Breast Cancer Prognosis. *Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges*.

Peling, I. B., Arnawan, I. N., Arthawan, I. P., & Janardana, I. (2017). Implementation of Data Mining To Predict Period of Students Study Using Naive Bayes Algorithm. *International Journal of Engineering and Emerging Technology, 2*(1), 53-57.

Phan, A. V., Nguyen, M. L., & Bui, L. T. (2017). Feature weighting and SVM parameters optimization based on genetic algorithms for classification problems. *Applied Intelligence volume, 46*, 455–469. https://doi.org/10.1007/s10489-016-0843-6

Pisner, D. A., & Schnyer, D. M. (2020). Chapter 6 - Support vector machine. In *Machine Learning* (pp. 101-121).

Pouriyeh, S., Vahid, S., Arabnia, H. R., & Sannino, G. (2017). A Comprehensive Investigation and Comparison of Machine Learning Techniques in the Domain of Heart Disease. *IEEE Symposium on Computers and Communications* . https://doi.org/10.1109/ISCC.2017.8024530

Priya, M., Juliet, P., & Tamilselvi, P. (2018). Performance Analysis of Liver Disease Prediction Using Machine Learning Algorithms.

International Research Journal of Engineering and Technology, 5(1), 206-211.

Priyam, A., Abhijeet, Gupta, R., Rathee, A., & Srivastava, S. (2013). Comparative Analysis of Decision Tree Classification Algorithms. *International Journal of Current Engineering and Technology, 3*(2), 334-337.

Rahmadani, S., Dongoran, A., Zarlis, M., & Zakarias. (2018). Comparison of Naive Bayes and Decision Tree on Feature Selection Using Genetic Algorithm for Classification Problem. *International Conference on Computing and Applied Informatics, 978*. https://doi.org/10.1088/1742-6596/978/1/012087

Rahman, A. S., Shamrat, F. J., Tasnim, Z., Roy, J., & Hossain, S. A. (2019). A Comparative Study On Liver Disease Prediction Using Supervised Machine Learning Algorithms. *International Journal of Scientific & Technology Research, 8*(11), 419-422.

Rahman, M., & Kumar, V. (2020). Machine Learning Based Customer Churn Prediction In Banking. *International Conference on Electronics, Communication and Aerospace Technology*, 1196-1201. https://doi.org/10.1109/ICECA49313.2020.9297529

Rajan, D. P., Kannan, K. S., Divya, P., & S, V. (2022). Comparative Analysis of Liver diseases by using Machine Learning Techniques. *International Conference on Emerging Smart Computing and Informatics*.

Rawat, V., & Suryakant. (2019). A Classification System for Diabetic Patients with Machine Learning Techniques. *International Journal of Mathematical, Engineering and Management Sciences, 4*(3), 729–744. https://doi.org/10.33889/IJMEMS.2019.4.3-057

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *International Conference on Machine Learning, Big Data, Cloud*

and Parallel Computing. https://doi.org/10.1109/COMITCon.2019.8862451

Safitri, A. R., & Muslim, M. A. (2020). Improved Accuracy of Naive Bayes Classifier for Determination of Customer Churn Uses SMOTE and Genetic Algorithms. *Journal of Soft Computing Exploration, 1*(1), 70-75. https://doi.org/10.52465/JOSCEX.V1I1.5

Salas-Eljatib, C., Fuentes-Ramirez, A., Gregoire, T. G., Altamirano, A., & Yaitul, V. (2018). A study on the effects of unbalanced data when fitting logistic regression. *Ecological Indicators, 85*, 502-508.

Savio, I., & Chakraborty, U. K. (2019). Genetic Algorithm : An Approach on Optimization. *International Conference on Communication and Electronics Systems* . https://doi.org/10.1109/icces45898.2019.9002372

Secherla, S. (2021, June 18). *Understanding Optimization Algorithms in Machine Learning*. Towards Data Science. https://towardsdatascience.com/understanding-optimization-algorithms-in-machine-learning-edfdb4df766b

Serokell. (2020, December 8). *Machine Learning Optimization Methods and Techniques*. Better Programming. https://betterprogramming.pub/machine-learning-optimization-methods-and-techniques-56f5a6fc5d0e

Shah, D., Patel, S., & Bharti, S. K. (2020). Heart Disease Prediction using Machine Learning Techniques. *SN Computer Science volume, 1*. https://doi.org/10.1007/s42979-020-00365-y

Soni, M., & Varma, S. (2020). Diabetes Prediction using Machine Learning Techniques. *International Journal of Engineering Research & Technology, 9*(9), 921-925.

Soumaya, Z., Taoufiq, B. D., Benayad, N., Yunus, K., & Abdelkrim, A. (2021). The detection of Parkinson disease using the genetic algorithm and SVM classifier. *Applied Acoustics, 171*. https://doi.org/10.1016/j.apacoust.2020.107528

Sundararajan, A., & Gursoy, K. (2020). Telecom Customer Churn Prediction. *Rutgers University*. https://doi.org/10.7282/t3-76xm-de75

Syarif, I., Prugel-Bennett, A., & Wills, G. (2016). SVM Parameter Optimization Using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control), 14*(4), 1502-1509. https://doi.org/10.12928/TELKOMNIKA.v14i4.3956

Tabassum, M., & Mathew, K. (2014). A Genetic Algorithm Analysis towards Optimization solutions. *International Journal of Digital Information and Wireless Communications, 4*(1), 124-142. https://doi.org/10.17781/P001091

Tabassum, S., Sampa, M. B., Islam, R., Yokota, F., Nakashima, N., & Ahmed, A. (2020). A Data Enhancement Approach to Improve Machine Learning Performance for Predicting Health Status Using Remote Healthcare Data. *International Conference on Advanced Information and Communication Technology*, 308-312. https://doi.org/10.1109/ICAICT51780.2020.9333506

Taheri, S., & Mammadov, M. (2013). Learning the naive Bayes classifier with optimization models. *International Journal of Applied Mathematics and Computer Science, 23*(4), 787–795. https://doi.org/10.2478/amcs-2013-0059

Tan, J. (2020, August 10). *How to improve data quality for machine learning?*. Towards Data Science.

https://towardsdatascience.com/how-to-improve-data-preparation-for-machine-learning-dde107b60091

Taunk, K., De, S., & Verma, S. (2019). A Brief Review of Nearest Neighbor Algorithm for Learning and Classification. *International Conference on Intelligent Computing and Control Systems*. https://doi.org/10.1109/ICCS45141.2019.9065747

Tougui, I., Jilbab, A., & El Mhamdi, J. (2020). Heart disease classification using data mining tools and machine learning techniques. *Health and Technology, 10*, 1137–1144. https://doi.org/10.1007/s12553-020-00438-1

UCI. (n.d.). UCI ML Repository. https://archive.ics.uci.edu/ml/index.php

Venter, G. (2010). Review of Optimization Techniques. *Encyclopedia of Aerospace Engineering*. https://doi.org/10.1002/9780470686652.eae495

Vié, A., Kleinnijenhuis, A. M., & Farmer, D. J. (2021). Qualities, challenges and future of genetic algorithms: a literature review. *ArXiv*.

الملخص

إن التعلم الآلي Machine learning (ML) قد استقطب في الآونة الأخيرة اهتمامًا كبيرًا من العامة وأصبح تدريجيًا أكثر شيوعًا وشعبيةً نظرًا لقدرته على التنبؤ واتخاذ القرارات بناءً على البيانات المتوافرة. ويؤكد الاستخدام المتزايد للتعلم الآلي ML في الميادين الحرجة مثل التشخيصات الطبية والكشف عن الاحتيال على الحاجة إلى تحسين أدائه بشكل مستمر، حيث تسهم تلك التحديثات والتحسينات بتعزيز أداء أدوات التعلم الآلي وهذا من خلال استخدام شتى آليات التحسينoptimization methods (OM) . ومع ذلك، فإن أداء خوارزميات التعلم الآلي ML يكون في بعض الأحيان ملزمًا بالقيود المتعلقة بطبيعة البيانات التي يمكن أن تعيق من أداء تلك الخوارزميات. ولذلك، فإن هذا البحث يقترح خوارزمية التصنيف الجديدة GADIC والتي تعتمد على إعادة تنسيق البيانات حسب أهميتها Data Importance (DI) والخوارزميات الجينية Genetic Algorithms (GA)للتغلب على هذه المشكلات ولتحسين فعالية وقدرات خوارزميات التعلم الآلي ML. كما يهدف هذا البحث لتقييم تأثير الخوارزمية المقترحة على أداء المصنفات ومقارنة أداءها مع خوارزميات التصنيف التقليدية الأخرى لقياس قدرتها على تحسين أداء المصنفات.

وتتألف الخوارزمية المقترحة من ثلاث مراحل وهي مرحلة إعادة تنسيق البيانات حسب مفهوم أهمية البيانات، ومرحلة التدريبtraining phase حيث يتم تطبيق الخوارزميات الجينية GA على مجموعة التدريب المعاد تنسيقها، ومرحلة الاختبارtesting phase حيث يتم حساب متوسط حالات مجموعة الاختبار المعاد تنسيقها بناءً على حالات مماثلة في مجموعة التدريب. وقد تم اختبار الخوارزمية المقترحة على خمسة من مصنفات خوارزميات التعلم الآلي ML وهي Support Vector Machine (SVM) و K-Nearest Neighbour (KNN) وLogistic Regression (LR) وDecision Tree (DT) وNaïve Bayes (NB). حيث تم تقييم المصنفات باستخدام سبع مجموعات بيانات مفتوحة المصدر وهي Cleveland Heart Disease، وIndian Liver Patient، وPima Indian Diabetes، وEmployee Future Prediction، وTelecom Churn Prediction، وBank Customer Churn، وTech Students. من حيث الدقة، أظهرت النتائج أنه باستثناء انخفاض بنسبة 1 ٪ تقريبًا في الدقة في مصنف NB في مجموعة بيانات Cleveland Heart Disease، فإن خوارزمية GADIC قد عززت بشكل كبير أداء معظم المصنفات باستخدام مجموعات بيانات مختلفة. بالإضافة إلى ذلك، أظهرت KNN مع استخدام GADIC أكبر مكاسب في الأداء عند مقارنتها بمصنفات التعلم الآلي الأخرى مع GADIC وهذا بمتوسط زيادة يعادل 16.79٪، يليها SVM بمتوسط زيادة يعادل 9.03٪. كما سجلت LR أدنى تحسن بمتوسط زيادة بنسبة 5.96٪.

جامعة البحرين

كلية العلوم **قسم برامج الدراسات العليا**



مصنف مستحدث يعتمد على الخوارزميات الجينية وإعادة تنسيق البيانات حسب أهميتها

إطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير
في علوم وتحليل البيانات الضخمه

إعداد

عائشة خالد الخياط

20082338

إشراف

أ.د . نبيل حويحي
أستاذ

جامعة البحرين

مملكة البحرين
مارس 2023