A NOVEL FRAMEWORK FOR RULE MINING USING SOFT COMPUTING ALGORITHM

Thesis Submitted for the Award of the Degree of

DOCTOR OF PHILOSOPHY

in

Computer Science and Engineering

By

Mrinalini Rana

Registration Number: 41800535

Supervised By

Dr. Omdev Dahiya (26990)

Computer Science and Engineering (Assistant Professor)

Ph.D. (CSE)

LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2023

DECLARATION

I, hereby declared that the presented work in the thesis entitled "**A Novel Framework for Rule Mining using Soft Computing Algorithm**" in fulfilment of degree of **Doctor of Philosophy (Ph. D.)** is the outcome of research work carried out by me under the supervision Dr. Omdev Dahiya, working as Assistant Professor, in the School of Computer Science and Engineering of Lovely Professional University, Punjab, India. In keeping with the general practice of reporting scientific observations, due acknowledgments have been made whenever the work described here has been based on the findings of another investigator. This work has not been submitted in part or full to any other University or Institute for the award of any degree.

(Signature of Scholar)

Name of the scholar: Mrinalini Rana

Registration No.: 41800535

Department/School: School of Computer Science and Engineering
Lovely Professional University,
Punjab, India

CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled "A Novel Framework for Rule Mining using Soft Computing Algorithm" submitted in fulfillment of the requirement for the reward of degree of **Doctor of Philosophy (Ph.D.)** in the School of Computer Science and Engineering, is a research work carried out Mrinalini Rana, 41800535, is bonafide record of her original work carried out under my supervision and that no part of thesis has been submitted for any other degree, diploma or equivalent course.

(Signature of Supervisor)

Name of supervisor: Dr. Omdev Dahiya

Designation: Assistant Professor

Department/School: School of Computer Science and Engineering

University: Lovely Professional University

ABSTRACT

Data mining represents a process that involves sorting through large datasets based on some peculiar relationships or patterns. This is mainly done to resolve various business problems on a large scale. Further, data mining is a crucial part of any organization's successful initiative in analyzing historical data or data streams collected from various applications. In recent times, the enormously rising data volumes have led to several issues in front of traditional data mining approaches that could only be resolved with advanced rule mining frameworks. In this scenario, the present research involved soft computing algorithms, and mathematical optimization approaches in association with rule mining to provide highly accurate and relevant data mining outcomes that are less time-consuming.

The objective of this study is to explore how artificial intelligence can be applied in the field of rule mining and present a new algorithm called G-ABC, which has been adapted from natural bee colony optimization. G-ABC is well-suited for preprocessing or feature selection when it comes to mining rules. It also offered promising results during its evaluation against traditional algorithms. This research was conducted to reduce the execution time and decrease the itemsets generated without any support and confidence threshold value. The core technique is to perform feature selection first using G-ABC algorithms. The proposed algorithm reduced the execution time and the number of selected features. G-ABC used the technique of five active bees working at one time. The proposed work involved two datasets, namely the Baseball and Twitter datasets that are processed and used for the evaluation of the designed framework at the feature extraction as well as classification stage. It is observed that G-ABC involved several bees working simultaneously, significantly reducing the number of relevant features and execution time of the process. Association rule mining and mean-variance are also involved along with G-ABC. Due to better feature selection, the G-ABC outperformed the other optimization approaches at the feature selection level. Further, at the classification stage, Neural Network outperformed the other cross validators namely, naïve Bayes (NB), K-Nearest Neighbor (KNN), and Support Vector Machine (SVM). The best performer, G-ABC with NN demonstrated an average accuracy of 98% with an execution time of 3secs.

Secondly, Association Rule Mining is performed without using minimum support and confidence. The fitness function used for finding the appropriate rule formula includes minimum and variance calculations. To verify the result, two types of datasets are being used. One is labeled data where direct feature selection and association rule mining can be implemented, and the second dataset is unlabeled data where the first K-means clustering algorithm is implemented, followed by rest implementation. Data was segregated into 3 Ground Truths: Positive, Negative, and Neutral. The proposed algorithm is equated with the basic PSO, ABC, and PSO-ABC algorithms at the feature selection stage. Four classifiers, KNN, NB, SVM, and NN, are used to validate the algorithm, with a 70:30 percent training and testing division. For the second part of the proposed model, t the same four classifiers are used to validate the model implementation again

The complete framework demonstrated that Neural Network is the best-fit classifier and represents the best accuracy, precision, recall, and f-score value. After selecting features using the proposed algorithm, G-ABC data is tested and trained using a 70:30 ratio, and 4 classifiers, KNN, NB, SVM, and NN, are deployed to classify the dataset. For result generation, G-ABC is compared with PSO, ABC, and PSO-ABC. The results show that the G-ABC algorithm using neural networks performs better in terms of feature selection (91 features were chosen out of 100 records), execution time (three seconds), and accuracy (98%). Without any backing or assurance, the results are carried forward for association rule mining. According to the results, the G-ABC with Neural Network algorithm performs better in terms of ACCURACY (G-ABC with NN): 97.56%, PRECISION (G-ABC with NN): 67.78%, RECALL (G-ABC with NN): 96%, and F MEASURE (G-ABC with NN): 94%.

We proposed a novel metaheuristic approach for advanced rule mining using a soft computing algorithm framework. In the future, more metaheuristic algorithms can be used for better optimization. Furthermore, different datasets provide different environments for the algorithm to evolve and therefore produce a better solution that reaches an approximate solution faster than in an environment so more datasets may be used for evaluating the performance measures. In addition to this, deep learning may be involved in the presented research work to further improve the feature extraction accuracy and reduce the overall execution time of the process.

ACKNOWLEDGEMENT

Acknowledgement is not only a ritual but also an expression of indebtedness to all those who have helped in the completion of this thesis. At foremost, I thank the Lord Almighty, the source of wisdom, who has provided me the with ability, the resources, the opportunity, and his kind support. I consider myself fortunate and privileged to have Dr. Omdev Dahiya as my supervisor. I am deeply indebted to him for shaping my research path by guiding me with his extensive knowledge and discussions. I express my deep gratitude to my supervisor for his motivation, persistent encouragement, and keen involvement that persuaded me to complete this research. I am highly thankful to him. Without his guidance, help, and patience, I would never have accomplished this thesis work. He has been an incredible mentor to me.

I wish to extend my thanks to all faculty members of the Ph.D. (CSE) for attending my seminars and for their insightful comments and constructive suggestions to improve the quality of this research work.

I thank all the faculty members of the CSE department, Lovely Professional University, for rendering excellent cooperation. I will remain indebted to my family for providing me the confidence and comfort to undertake this thesis. I find no words to thank my parents for their dedicated support and prayers offered to complete my research work. I wish to express my deep sense of gratitude to my husband Mr. Ujjwal Makkar and my parents, for their constant support and love.

Mrinalini Rana

TABLE OF CONTENTS

DECLARATION	i
CERTIFICATE	ii
ABSTRACT	iii
ACKNOWLEDGEMENT	v
TABLE OF CONTENTS	vi
LIST OF TABLES	viii
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xiii
CHAPTER 1: INTRODUCTION	1-17
1.1 Background Information	1
1.2 Introduction to Data Mining	2
1.3 Literature Review	9
1.4 Problem Statement	13
1.5 Research Objectives	14
1.6 Organization of the Thesis	15
CHAPTER 2: REVIEW OF EXISTING FRAMEWORKS FOR ASSOCIATION RULE MINING	18-43
2.1 Soft Computing	18
2.2 Association Rule Mining	21
2.3 Artificial Intelligence	26
2.4 Machine Learning	28
2.5 Related Work	35
Summary	43
CHAPTER 3: PRE-PROCESSING OF DATA FOR EFFICIENT RULE GENERATION	44-53
3.1 General Architecture	44
3.2 Primary data vs. Secondary data	45
3.3 Python and its usage in data mining architecture	46
3.4 Metaheuristic Algorithms for Pre processing	50

3.5 Artificial Bee Colony Optimization 52

Summary 53

CHAPTER 4: A HYBRID APPROACH USING GROUPED ABC FOR FEATURE SELECTION AND MEAN-VARIANCE OPTIMIZATION FOR RULE MINING 54-71

4.1 Background 54

4.2 Rule Mining Architectures/Algorithms 56

4.3 Propagation Based Rule Mining Architectures 62

Summary 71

CHAPTER 5: PROPOSED METHODOLOGY FOR RULE MINING 72-91

5.1 Proposed work 72

5.2 Pre-processing 78

5.3 Feature Selection 80

5.4 Rule Mining 84

Summary 91

CHAPTER 6: RESULTS AND DISCUSSION 92-160

6.1 Introduction 92

6.2 Evaluation for Feature Selection Approaches 95

6.3 Evaluation of Classification Approaches 97

6.4 Evaluation using Multiple Simulations 146

6.5 Evaluation with more data sample 154

CHAPTER 7: CONCLUSION AND FUTURE SCOPE 161-164

7.1 Conclusion 161

7.2 Future Scope 163

REFERENCES 165

LIST OF PUBLICATIONS 182

LIST OF TABLES

Table 1.1 Clustering Techniques with Their Algorithms 4
Table 1.2 Relation of Data Mining Tasks with Data Mining Techniques 6
Table 2.1 Handful of Popular AI Techniques 26
Table 2.2 Machine Learning Algorithms 29
Table 2.3 Existing Association Rule Mining with Their Advantages and Limitations 37
Table 2.4 Comparison of Existing Techniques using the ML models 41
Table 3.1 Difference between Primary and Secondary Data 46
Table 3.2. Benefits of Optimization Techniques 50
Table 4.1 Comparison between KNN and DT 68
Table 5.1 Optimized feature set 83
Table 5.2 Rule sets 87
Table 6.1 Simulation Ordinals 94
Table 6.2 Variation in the Execution Time using Different Optimization Approaches 96
Table 6.3 Precision Analysis using Baseball Dataset 98
Table 6.4 Precision Analysis using Twitter Dataset 101
Table 6.5 Sensitivity Analysis using Baseball Dataset 108
Table 6.6 Sensitivity Analysis using Twitter Dataset 111
Table 6.7 F-measure Analysis using Baseball Dataset 118
Table 6.8 F-measure Analysis using Twitter Dataset 120
Table 6.9 Accuracy Analysis using Baseball Dataset 126
Table 6.10 Accuracy Analysis using Twitter Dataset 129
Table 6.11 Execution Time Analysis using Baseball Dataset 13
Table 6.12 Execution Time Analysis using Twitter Dataset 137
Table 6.13 Precision Analysis for Multiple Simulations 146
Table 6.14 Recall Analysis for Multiple Simulations 148
Table 6.15 F-measure Analysis for Multiple Simulations 149
Table 6.16 Accuracy Analysis for Multiple Simulations 151
Table 6.17 Execution Time Analysis for Multiple Simulations 153
Table 6.18 Precision Analysis using Twitter Dataset 155

Table 6.19 Recall Analysis using Twitter Dataset 156
Table 6.20 F-measure Analysis using Twitter Dataset 157
Table 6.21 Accuracy Analysis using Twitter Dataset 158

LIST OF FIGURES

Figure 1.1 Data Mining Architecture and Algorithm Supervision 7

Figure 1.2 Data Correlation 8

Figure 2.1 Soft Computing Techniques 19

Figure 2.2 Illustrative Transactional Database 22

Figure 2.3 Taxonomy of Association Rule Mining [Yazgana & Kusakci, 2016] 22

Figure 2.4 Architecture of Artificial Neural Network 32

Figure 2.5 Forward Propagational Neural Network 32

Figure 2.6 Back Propagational Neural Network 33

Figure 2.7 Figure 2.7 Forward and Back Propagational Network 34

Figure 2.8 Spark streaming Process for data distribution [Liu et al., 2021] 36

Figure 3.1 General rule formation and mapping architecture 45

Figure 3.2 Stemming algorithms [Ismailov et al., 2016] 48

Figure 3.3 Data tokenization 49

Figure 4.1 Fuzzy rule engine 55

Figure 4.2 Propagation based rule mining architecture 56

Figure 4.3 Workflow of GA 62

Figure 4.4 Gaussian distribution for zero standard deviation 64

Figure 4.5 [a] Two types of Dataset [b] Classification 66

Figure 4.6 [a] Linear Classification [b] 3D representation 66

Figure 4.7 KNN 68

Figure 4.8 Biological Neuron 70

Figure 4.9 Structure of ANN 70

Figure 5.1 The process of proposed algorithm (R-Miner Using soft computing) 73

Figure 5.2 Stop word sample 79

Figure 5.3 Data Separation against its ground truth value 79

Figure 5.4 The process of proposed ABC algorithm 81

Figure 5.5 ABC Architecture 82

Figure 5.6 Fitness function 83

Figure 5.7 Propagation based rule mining architecture 86

Figure 6.1 Comparison of Number of Selected Features using Different Optimization Approaches 95
Figure 6.2 Comparison of Execution Time required by Different Optimization Approaches 97
Figure 6.3 Precision Analysis using NB 104
Figure 6.4 Precision Analysis using KNN 105
Figure 6.5 Precision analysis using SVM 106
Figure 6.6 Precision analysis using NN 107
Figure 6.7 Sensitivity Analysis using NB 114
Figure 6.8 Sensitivity Analysis using KNN 115
Figure 6.9 Sensitivity Analysis using SVM 116
Figure 6.10 Sensitivity Analysis using NN 117
Figure 6.11 F-measure Analysis using NB 123
Figure 6.12 F-measure Analysis using KNN 124
Figure 6.13 F-measure Analysis using SVM 125
Figure 6.14 F-measure Analysis using NN 126
Figure 6.15 Accuracy Analysis using NB 132
Figure 6.16 Accuracy Analysis using KNN 133
Figure 6.17 Accuracy Analysis using SVM 133
Figure 6.18 Accuracy Analysis using NN 134
Figure 6.19 Execution time Analysis using NB 139
Figure 6.20 Execution time Analysis using KNN 140
Figure 6.21 Execution time Analysis using SVM 141
Figure 6.22 Execution time Analysis using NN 142
Figure 6.23 Precision Comparative Analysis for G-ABC and NN 143
Figure 6.24 Sensitivity Comparative Analysis for G-ABC and NN 144
Figure 6.25 F-measure Comparative Analysis for G-ABC and NN 144
Figure 6.26 Accuracy Comparative Analysis for G-ABC and NN 145
Figure 6.27 Execution Time Comparative Analysis for G-ABC and NN 145
Figure 6.28 Precision Analysis for Multiple Simulations 147
Figure 6.29 Recall Analysis for Multiple Simulations 149

Figure 6.30 F-measure Analysis for Multiple Simulations 150
Figure 6.31 Accuracy Analysis for Multiple Simulations 153
Figure 6.32 Execution Time Analysis for Multiple Simulations 154
Figure 6.33 Precision Analysis using Twitter Dataset 156
Figure 6.34 Recall Analysis using Twitter Dataset 157
Figure 6.35 F-measure Analysis using Twitter Dataset 158
Figure 6.36 Accuracy Analysis using Twitter Dataset 159

LIST OF ABBREVIATIONS

Acronym	Term
GT	Ground Truth
ARM	Associate Rule Mining
GSP	Generalized Sequential Pattern
SPADE	Sequential Pattern Discovery Considering the Equivalent class
FREESPAN	Frequent Pattern Projected Sequential Pattern Mining
FSG	Frequent Discovery Algorithm
GSPAN	Graph-Based Substructure Pattern Mining Algorithm
AI	Artificial Intelligence
NN	Neural Network
SVM	Support Vector Machine
DT	Decision Tree
LR	Logistic Regression
KNN	K-Nearest Neighbor
NB	Naïve Bayes
RF	Random Forest
SML	Statistical Machine Learning
MSE	Mean Squared Error
SE	Standard Error
RMSE	Root Mean Square Error
SD	Standard Deviation
MFP	Mining Frequent Pattern
AMO	Animal Migration Optimization
EFP	Extend Frequent Patter

SAD	Style Aided Design
PSA	Porter Stemming Algorithm
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
PSO	Particle Swarm Algorithm
CS	Cuckoo Search
FS	Frog Search
GA	Genetic Algorithm
ATS	Attribute Sets
FL	Fuzzy Logic

CHAPTER 1:
INTRODUCTION

1.1 Background Information

1.2 Introduction to Data Mining

1.3 Literature Review

1.4 Problem Statement

1.5 Research Objectives

1.6 Organization of the Thesis

1.1 Background Information

With the increasing volume, variety, and modularity of the data present in the modern world, it becomes quite necessary to organise the data in such a manner that useful information can be extracted out of the data. Any prediction architecture, such as the weather app from Google, requires a significant amount of data from previous years in order to produce results for the upcoming years. The human brain has been proven to be the best computation architecture that can identify, think and analyse things based on past experiences. This is possible with a long year of studies of things, objects, behaviours, etc relative to several things. Due to the increased volume of data and increased complexity within the architecture of the collected data, the computation complexity of human computation increases tremendously.

Computation complexity refers to the overall computation time to perform an operation. Hence a system-aided design is required to fulfil the increased volume and variety needs. If a system has to produce a result, three things must be associated with the system namely, the dataset, the Ground Truth (GT) value, and the rule set. Dataset refers to the collection of data representing a particular event, point, or factor and is referred to as GT. The rule sets define the outcome of the input from the test data. The test data is the data that is generated from the real world and is supplied to the

system to quantify its relative class or GT to make a decision. The entire process is referred to as data mining. In the architecture of data mining, the data is bonded with rules to understand the architecture of the data. For example, if there are two classes "Bus" and "Train" and both a feather, for instance, $\mathrm{wt_{train}} \geq 20000\ \mathrm{kg}$ and $5000\ \mathrm{kg} \leq \mathrm{wt_{bus}} \leq 20000\ \mathrm{kg}$ where wt_train represents the weight of the train and wt_bus represents the weight of the bus. In such a scenario, the associated rule would be, Rule: $\text{if } \mathrm{wt_{test}} < 20000\ \mathrm{kg}$ then Refer Bus Else Refer Train .

With the high volume of data and variety in context, the GT requires to be defined by multiple features to establish more co-relation among the data elements. With the increased number of attributes, the associated membership function of the input set also increases. The membership function refers to the variation in the supplied values. For example, a food dish may have three membership functions "Good", "Eatable" and "Avoidable" and they are to be mapped on a statistical scale depending upon the requirement. Fuzzy logic, the Apriori algorithm, and the decision tree algorithm are one of the finest examples of statistical rule mining architectures [**Wang & Gao**, **2021**]. Increasing rule sets will increase the overall computation complexity and hence propagation-based learning behaviour was proposed and works pretty well in real-world applications as well [**Sinaei & Fatemi**, **2018**]. The recommendation system from "Netflix" is one of the perfect examples where Netflix provides suggestions based on the previous history of the user profile. This research draft aims to shine the previous propagation-based architecture by contributing novelty when it comes to training the system.

1.2 Introduction to Data Mining

Data mining is an essential step used to discover unknown patterns from a large database. There are various functionalities, algorithms, models, and techniques used to discover and extract the relevant patterns from the large database repository [**Gheware *et al.***, **2014**; **Morik *et al.***, **2012**]. In the last few decades, data mining has played a vital role in decision-making and is considered an essential tool for performing different operations [**Kiranmai & Damodaram**, **2014**; **Turban**, **2011**]. To discover the knowledge, data mining plays a vital role in applying the algorithms, and data analysis techniques under certain limitations and produces a viable pattern over the data. According to various researchers., data mining is a viable process to discover the interesting patterns, associations, and relation between the significant structures from large database which is used to store in multiple sources such as data warehouse, and data repository [**Favaretto *et al.***,

2019], [**Han *et al.***, **2004**, **2006**], [**Tsai *et al.***, **2015**]. Further, studying the literature review, it is explicit that data mining is divided into the following types such as:-

- **Functions of Data Mining**

The kind of correlations or learning to be found throughout the data mining process can be specified using data mining functions or assignments [**Sumathi & Sivanandam**, **2006**]. Summary, characterisation and classification, relationship, segmentation, categorization, regression problems, extrapolation, and market analysis are a few of the key data mining functions [**Jain & Srivastava**, **2013**], [**Liao *et al.***, **2012], [Sharma**, **2014**]. The functions of data mining are illustrated as follows:

- **Classification**

The classification of data on the predetermined classes is recognized as the process of classification i.e. supervised learning. The classes are forecast using the classification algorithm. In the literature, the researchers have proposed a large collection of classification algorithms. The popular algorithms are stated in Table 1.1. However, other algorithms apart from this are fuzzy set theory, semi-supervised learning, tough set, and fuzzy sets also proposed by some practitioners.

- **Summarization**

A smaller set is produced through summarization, which gives a conceptually based overview of the specific information. Aggregation is a commonly used method for summarization, which can be applied at various levels of conceptualization and viewed from multiple perspectives. By combining different levels of abstraction and dimensions, it is possible to identify new patterns. Data summarization is often accomplished using techniques such as attribute-oriented induction and data cube analysis. [**Hung *et al.***, **2015**].

Data cube technique (also known as "multidimensional databases" or "materialised views") materialises costly calculations involving group functions that are frequently queried and stores the outcome as materialised views for decision assistance and knowledge discovery.

- **Characterization and Discrimination**

Characterization is essentially a data-based description. Characterization is used to create a conceptual hierarchy and characterisation rules. On the other hand, discrimination is employed to identify among distinct data sets, variety. Discriminatory norms are produced as the result of discrimination [**Bhatnagar *et al.***, **2015**].

- **Clustering**

The process of clustering is used to divide or segment data objects [or observations] into smaller groupings or clusters. The objects that are close to one another are grouped. Clustering categorises related data objects in a similar way to classification, however unlike classification, the class labels are not known [i.e., unsupervised learning]. One of the most well-known methods, cluster analysis is utilised not just in data mining but also in a variety of other fields, including statistics, pattern classification, reinforcement learning, object tracking, knowledge representation, biotechnology, etc.

Several researchers have introduced and/or discussed various novel clustering algorithms, in addition to the commonly recognized ones listed in Table 1.1.[**Gupta & Chandra**, **2020**]. Among these approaches is the minimum description length method, which is parameter-free and utilizes parallel computing, as well as density-based clustering. Additionally, there is a genomic data clustering technique that relies on the z-score measure, a fully automated clustering algorithm for high-dimensional categorical data, and a nature-inspired swarm-based intelligent technique. [**Mampaey & Vreeken**, **2013**], [**Wang *et al.***, **2011**].

Table 1.1 Clustering Techniques with Their Algorithms

Category	Algorithm	Concept based
Hierarchical Clustering	Divisive Analysis	Using the divisive technique
	Agglomerative nesting	Using the Agglomerative method
	Chameleon	Dynamic Modelling
	Balanced iterative reducing and clustering using hierarchies	Clustering feature tree
	Hierarchical clustering based on probability	Probabilistic model
	K-means	To determine the centroid
	K-medoid	Representative Object

Partitioning based clustering	Clustering large applications considering the randomized search	Randomized sampling
	Partitioning around medoid	Representative object
Grid based clustering	Clustering in Quests	Monotonicity of dense cells w.r.t. dimensional
Density based Clustering	Density based spatial clustering to eliminate the noise	Regions are connected by having high density
	Ordering Points to identify the clustering structure	Global density parameters used for connected regions considering the high density
	Density based clustering	Using the density distribution function

⇒ **Techniques of Data Mining**

Based on a variety of data mining methodologies or approaches, data mining objective(s) are accomplished. The researchers have thus far examined a wide variety of data mining approaches. Examples include visualisation, evolutionary computation, clustering algorithms, database, and data storage systems, statistics, and machine learning [**Venkatadri & Reddy**, **2011**].

⇒ **Algorithms of Data Mining**

Many researchers have proposed various algorithms, also referred to as methods, to carry out data mining tasks based on data mining techniques. Apriori algorithm, Naive Bayesian, k-Nearest Neighbour, k-Means, CLIQUE, STING, etc. are a few examples [**T. W. Liao & Triantaphyllou**, **2008**].

⇒ **Data Mining Domains**

Data Mining is widely used in different set of domains such as time-series, spatial data mining, temporal data mining, business, medical, engineering, and temporal-spatial data mining, etc. Each domain in data mining can have different applications [**Esling & Agon**, **2012**].

Data mining is the approach to extract relevant information from a high-volume data. In order to do so, four main components play significant role in quantifying the polarity of the returned result.

The essentials of data mining are

a) Data itself
b) The features and the co-relations of the data set
c) The type of training engine to be used
d) The classification mechanism

Table 1.2 Relation of Data Mining Tasks with Data Mining Techniques

Data Mining Techniques	Data Mining Tasks					
	Summarization	Classification	Outlier analysis	Discrimination and Characterization	Clustering	Trend and Regression analysis
Statistics	Yes	Yes	Yes	Yes	Yes	Yes
Machine Learning	No	Yes	Yes	Yes	Yes	Yes
Database system	Yes	Yes	Yes	Yes	Yes	Yes
Neural Network	No	Yes	Yes	Yes	Yes	Yes
Visualization	No	Yes	Yes	Yes	Yes	Yes
Fuzzy set and logic	No	Yes	Yes	Yes	Yes	No

Genetic Algorithm	No	Yes	Yes	No	Yes	Yes

The general data mining architecture is demonstrated in Figure 1.1 as follows.

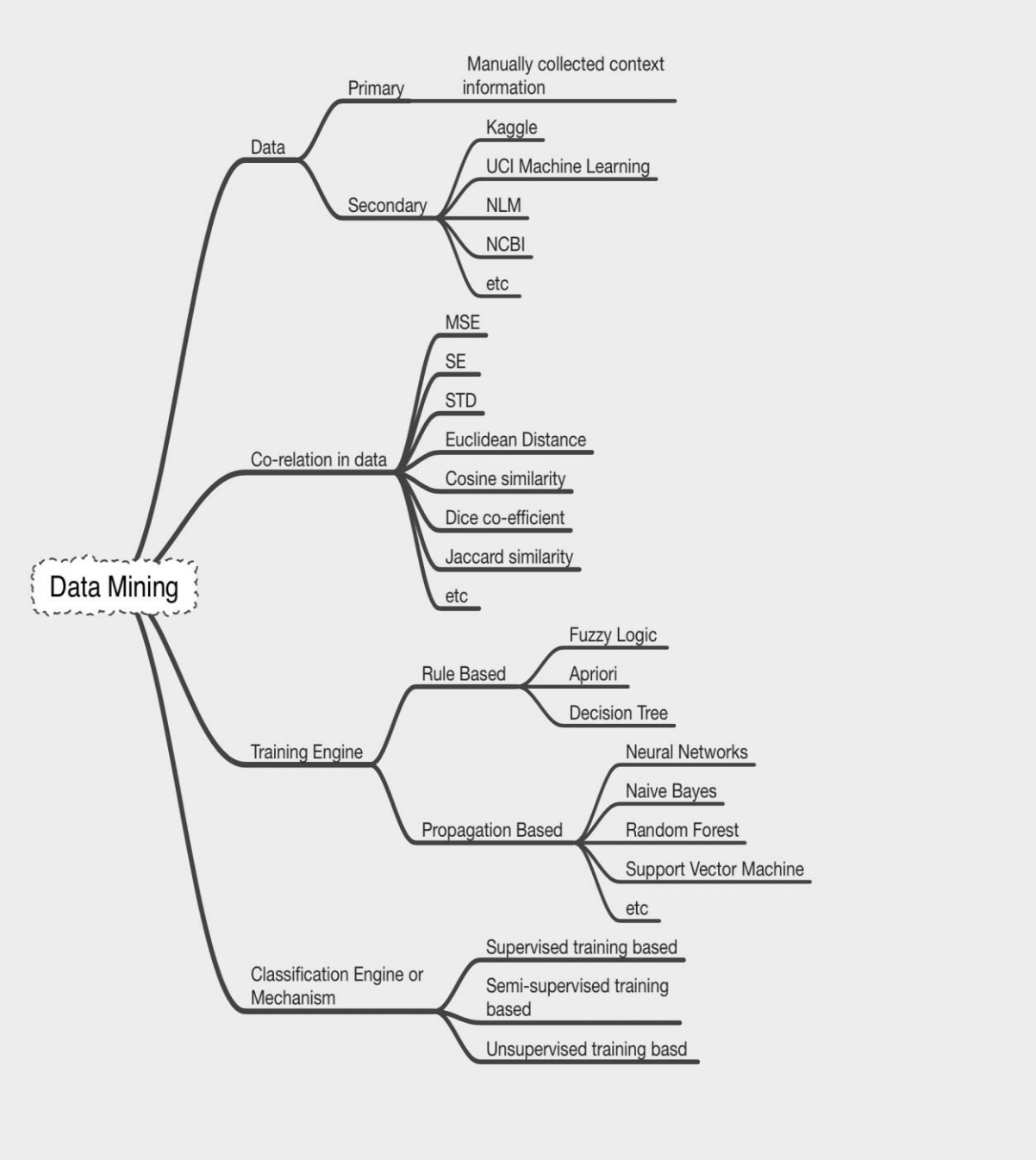

Figure 1.1 Data Mining Architecture and Algorithm Supervision

The data contains either the raw data or the feature extracted data along with the GT that defines the significance of the features [**Krig**, **2014**]. If the features are not provided in the data set, there are several feature extraction algorithms based on the type of data that is being utilized. In the case of the proposed work scenario, it is text data, and hence the number of features is quite limited to Term Frequency(TF), Inverse Document Frequency(IDF), and few similarity indexes that are listed in the proceeding chapters. The dataset can be either primary or represents the behaviour of the collection of the data. The data collection requires a lot of effort even if the data is collected from primary sources, the collected data requires validation so that the data can be used for any future analysis. According to Sir, Darbin, and Watson, if the data contains more than 5% un-corelated features or attributes or elements, the classification based on the data can be inaccurate up to 50%. as shown in figure 1.2 [**Salamon *et al.***, **2019**].

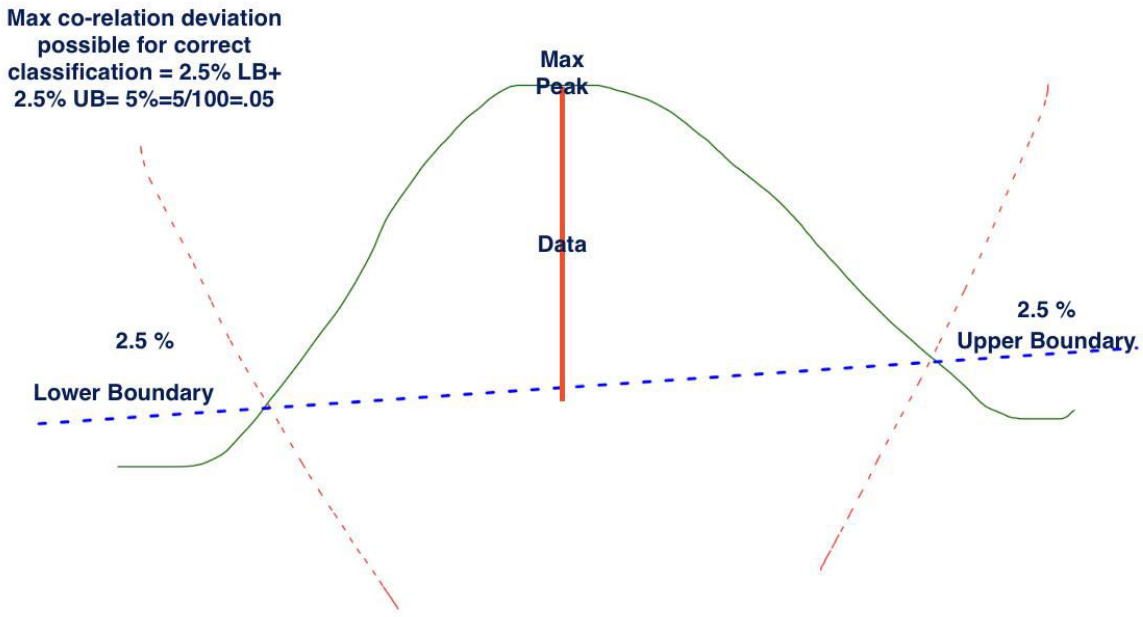

Figure 1.2 Data Correlation

In order to bring maximum co-relation among the data elements, it becomes necessary to put the elements close to its significant value. In order to calculate the significant value, three modes of evaluation are supported widely namely the mean, mode, or median. To train the system, it is essential that the attributes that represent a specific GT, represent high co-relation among the values of other entities of the same group. High co-relation will satisfy the Darbin and Watson

method and the data will produce more preciseness in terms of identifying identities from the same class. There are anonymous ways to evaluate the co-relation among the data elements using soft computing and the most popular architecture is the statistical approach of evaluation [**Shively *et al.***, **1990**].

1.3 Literature Review

The literature survey indicated that efforts are made to develop many association rule mining algorithms. Moreover, it is very important to have fewer meaningful rules rather than having more mixed (relevant and irrelevant) rules. Different research has been available to optimize the association rule mining or data mining output. This literature survey indicates the survey related to different association rule mining algorithm using optimization techniques that generate positive as well as negative rules.

Classification is a fundamental problem in machine learning. It has applications in almost all domains such as biomedical, industrial automation, biometric recognition, and advertising. The objective is to assign every data instance to one of the classes or categories with minimum error and/or maximum score error. Classification with Genetic Algorithm (GA) has been successfully used for clustering problems. It can utilize the existing feature set and maximize its predictive power**[Kuo *et al.,* 2019; Pu *et al.,* 2021; Karunyalakshmi *et al.*, 2017]**. The objective is to find the optimal number of clusters that minimize an appropriate distance measure. The proposed technique is based on a generalized GA as well as on an exhaustive search along a randomized tree algorithm which will explore all possible subsets of attributes available in a given data set or can be generated using some attribute ordering or cluster using rules such that one set can always be found at any point along the tree. Their outcome was to generate the rules needing to discretize the attributes without overlapping of frequent itemsets. **Creighton *et al.*, (2003)** used clustering with parameters support and confidence. The objective was to generate more accurate rules. They used a yeast data set. They have developed a database application to develop the rules. **Shady *et al.,* (2006)** used NLP with text mining. The objective was to create a model to improve text clustering quality. They used Reuters, ACM, and Brown data sets. They have achieved the quality of text clustering by using the proposed model. **Lungeanu *et al.*, (2008)** used rule mining with classification. The objective was to propose an interactive model for easy search. They used a Prima Indian Diabetic data set. They proposed an interactive system aimed to help medical doctors

explore the data and extracting new patterns. **Shijue *et al.*, (2008)** used rule mining with Genetic Algorithm. The objective was to propose a model that mines less rules for communication between educators and learners. They used synthetic and real data sets. They concluded that the proposed algorithm works 2-3 times faster than the traditional algorithm. **Wakabi-Waiswa *et al.,* (2008)** used rule mining with a genetic algorithm. The objective was to propose a model with a combination of genetic algorithm and Apriori to improve the overall performance. They used super market customer purchase dataset. They concluded proposed model is more efficient than the individual tradition algorithm. **Fang *et al.*, (2009)** used rule mining with a genetic algorithm. The objective was to propose a model to reduce computational complexity. They used an Alpha factor data set. They concluded that combined method mine more important and interesting patterns and rules. **Ayubi *et al.,* (2009)** used rule mining with all operators(<=,>=,=,=). The objective was to propose a model (MGR) using all operators, applicable to discrete attributes. They used the Balance dataset. They concluded that the proposed model proved beneficial in terms of the performance period and management of memory.

Rule mining is used to find different rules to use in predictive models. It can be done using decision trees or regular expression matching. Different algorithms can be used to generate these rules. Rule mining has been successfully applied in business, financial, engineering, and scientific domains**[Lungeanu *et al.*, 2008].** These results also show that there are some limitations in current rule mining work when compared to other techniques such as learning/perturbation**[Sharmila *et al.*, 2021]** and big data analysis. One of the main limitations of rule mining is its inability to extract strong relationships between various independent variables. **Indira *et al.*, (2012)** used rule mining with a genetic algorithm. The objective was to propose a model with a genetic algorithm with high accuracy. They used different datasets from various database repositories. They concluded that the proposed algorithm has higher predictive accuracy. **K.Y. *et al.*, (2012)** used rule mining with multi-Objective genetic algorithm. The objective was to propose a model using accuracy, comprehensibility, and definability rules. They used survey data on product design. They concluded that the proposed algorithm generates more crisp and suitable rules. **Divya *et al.*, (2013)** used rule mining using Biogeography based optimization (BBO). The objective was to propose a model using BBO. They used a synthetic dataset . They concluded that the proposed algorithm with the BBO feature is capable of finding accurate rules.

A parallel genetic-fuzzy mining framework is used for rule mining. The objective was to propose a model with master-slave architecture to first find the rule with Genetic Algorithm and then the best membership function used to mine association rules. They used a data set with 64 items and 10,000 transactions. They proposed a model to overcome the problem of low speed to find the fitness evaluation of the original algorithm. **Beiranvand *et al.*, (2014)** used particle swarm optimization for association rule mining. The objective was to propose a selection technique and redefine the "ibest" and "gbest" techniques. They used basketball, bodyfat, and quake data sets. They proposed a model using swarm optimization to mine the rules in one single step. **Luna *et al.*, (2014)** used mining rules with grammar guided genetic programming. The objective was to mine quantitative association rules with context free grammar to display the solution. They used zoo, lymphography, Wisconsin prognostic, sonar, primary-tumour etc including total of 20 datasets. They proposed a model to implement an interesting fitness function that reduce misleading gaps and is beneficial for non-expert users. **Chen *et al.*, (2015)** used rule mining with niche-aided gene expression programming (NEGP). The objective was to create a model for mining rules in big data sources in less execution time than Apriori and fp-growth. They used the iris dataset, an artificial simulation database (ASD). The accuracy rate for iris database is 56% and for ASD is 80.3%. **Djenouri *et al.*, (2017)** used rule mining with particle swarm optimization and compared it with genetic algorithm. The objective was to propose two new approaches GA-Apriori and PSO-Apriori. They used data sets from UCI machine learning. They used a total 20 datasets for analysis. They concluded that the quality obtained by PSO-Apriori is much better than previous technologies.

Rule mining is a powerful technique for aggregating very large volumes of data and proposing useful models for classification or prediction**[Tsang *et al.,* 2007].** However, rule mining approaches are complex and computationally intensive to enable effective mining. This paper presents an empirical evaluation of various rule-mining methods that have been used in the literature. Although rules mining may work well in some domains, it can also be quite ineffective when applied to other domains due to uncertain uncertainty of rules, lack of availability of experts' knowledge on specific domains, and challenges presented by noisy datasets. Rule mining techniques: **[Luna *et al.*, 2014]** two popular techniques which are simple yet effective: Genetic Algorithm (GA) and Artificial Intelligence (AI), rule acceleration methods such as Rule subset selection and Adaptive Gaussian process model, prediction methods such as Support Vector

Machines (SVM), Random Forests (RF), Random Forest (RF), Decision trees and Decision tree ensembles, and interestingness measures like k-nearest neighbours support vector machine with several alternatives and lift measure based on support vector machines. They concluded various unseen patterns and conditions for heart diseases in Bangladesh that helps practitioner. **Djenouri *et al.*, (2018)** compared various Data Mining models. The objective was to present survey or review of various Data Mining techniques in Mental Health Methods used in the past 10 Years. They concluded Decision Tree was used for schizophrenia and bipolar for maximum time and SVM is maximum applied algorithm on the Depression Data set. **Djenouri *et al.*, (2018)** used rule mining with a combination of Cluster and Genetic algorithms. The objective was to propose a model that reduces the execution time even for minimum support, and minimum confidence. They used data Sets from UCI Machine Learning, and Frequent Itemsets Mining Repository (a total of 20 datasets are used). They concluded that the "CGPUGA" proposed algorithm is 600 times faster than the traditional algorithm.

Feng *et al.*, (2019) used rule mining with Genetic Algorithm. The objective was to propose a model, DBSCAN in the algorithm to find traffic congestion. They used various sizes of road networks. They conclude that with the proposed model traffic congestion is predicted with high accuracy. **Wei *et al.*, (2019)** used rule Mining with cluster analysis. The objective was to propose a model using Financial Management Information. They used data sets namely MONKS problem, ABC alphabet data set, A–E alphabet data set, and SEA data set. They concluded that the proposed algorithm showed good results for large data sets.

A literature survey has indicated that many measures may be used to generate appropriate rules. The presentation may be enhanced by abolishing the essential to regulate the levels of the threshold for the standards of support and confidence **[Ghaleb *et al.,* 2019].** Even though some many models and algorithms possess a good performance, yet it is observed as lack of consideration of factors like support, confidence, lift, and certainty. Even though these are highly important and powerful in determining the preferences of participants, their effectiveness can be enhanced by considering such relevant factors. Thus, research should focus on the importance of such factors for better results. Many models are not effective in using algorithms with complex categorical datasets **[Basheer *et al.,* 2013].** This may increase the scope of research in this area.

This study is done to access the influence of different rule-mining techniques. A comprehensive review of research papers in this study provides an overview of the way different algorithms mine the rules

⇒ Research Gaps

The research gaps observed based on the literature survey are as follows:

1. It has been observed that there are many effective ways to enhance the rule mining approaches in which the **support and confidence** were evaluated by traditional method and changes were made to the rule mining approach. There is the possibility of the application of Swarm Intelligence Algorithms that could have been used to enhance the support and confidence by a meta-heuristic approach.
2. It has been observed that a **frequent pattern approach** subsequently provide what the user has demanded based on the ranking that is done on the base of usage pattern. The subtree generation process presented in many approaches, not only increases the computation complexity but also root node shifting produces score variation that sometimes results in bad mining results. The subtree graph generation could have been removed by using training and classification architecture.
3. It has been observed that the **Swarm Intelligence** approach could be utilized to improve the rule mining that was done traditionally using Apriori algorithm.
4. It has been observed that the **Genetic Algorithm** was used to reduce the computation complexity of the rule mining engine. GA belongs to the natural computing approach and requires an adaptive swarm-based architecture if it has to be applied to small set of data. GA requires a bulk amount of data for the generation of the mutation and the crossover. Hence a swarm intelligence-based combination could be adopted for future work

1.4 Problem Statement

Association rule mining has been a way to handle the data when it comes to solving a user query against a supplied set of data values. The data contains the feature set along with its GT value that represents the data as one identity. Association rule mining involves the creation and application of rule sets against supplied input values based on their associated features. Due to increasing versatility in the data, rule-based architectures lack in producing efficient and accurate results in a given interval of time. This is due to the anonymous number of rulesets that have to be surfed to

derive a conclusion against a specific context. The problem of this research work is to enhance the computation efficiency that is to be measured by quantitative parameters by précising the architecture of association rule.

1.5 Research Objectives

The defined objectives of the research are as follows.

a) To study and review the existing frameworks for association rule mining.

Association rule mining is a process of mining or generating interesting rules from the dataset. The literature survey indicates that efforts are made to survey many association rule mining algorithms. It is essential to provide more efficient rules for the betterment of analysis. Moreover, it is very important to have a smaller number of meaningful rules rather than having more mixed (relevant and irrelevant) rules. However, the classical Apriori algorithm generates more results and takes more execution time as well. To overcome this problem soft computing or evolutionary algorithms are used to find frequent items and to develop a global association rule in this proposed work

b) To pre-process the data obtained from various sources for efficient rule generation.

Feature selection is the process of selecting efficient features for the next analysis. In this study, an algorithm has been proposed **(G**rouped **A**rtificial **B**ee **C**olony Optimization **G-ABC**) to find optimized features. In G-ABC instead of using one employee at a time, bees will work together to get more optimized results. For the analysis, 2 datasets have been used. One used clustered data and the other include categorical data. If there is clustered data then K means is used for labeling the data in 3 labels: Positive, Negative, and Neutral. Then pre-processing including removing stop words and then tokenizing is used. Finally Grouped ABC is used for optimizing the feature selection process and then classified by using KNN, SVM, NB, and NN algorithm. According to the results Grouped- Artificial Bee Colony optimization is representing better results as compared to other evolutionary algorithms.

c) To propose a novel framework for rule mining using soft computing techniques.

Proposed a novel framework for performing rule mining tasks with the use of mean, variance optimization method where data elements or population after feature selection is divided into two parts first it is divided into three classes using the mean * variance optimization method afterward mean and variance are calculated for each element or population and if the condition is true then

the value is accepted otherwise the value is rejected. After calculating the final rules again four classifiers KNN, SVM, NB, and NN are used for a complete evaluation.

d) To compare the performance of the proposed work with existing work.

The proposed framework is compared with the PSO, ABC, and PSO-ABC hybrid models. Further to validate the results four classifiers are used such as KNN, NB, SVM, and NN, with a 70:30 percent training and testing division. For the second part of the proposed model, to validate the model implementation again, the same four classifiers are used.

The complete framework demonstrated that Neural Network is the best-fit classifier and represents the best accuracy, precision, recall, and f-score value. After selecting features using the proposed algorithm, G-ABC data is tested and trained using a 70:30 ratio, and 4 classifiers, KNN, NB, SVM, and NN, are deployed to classify the dataset. For result generation, G-ABC is compared with PSO, ABC, and PSO-ABC. The results show that the G-ABC algorithm using neural networks performs better in terms of feature selection (91 features were chosen out of 100 records), execution time (three seconds), and accuracy (98%). Without any backing or assurance, the results are carried forward for association rule mining. according to the results, the g-abc with neural network algorithm performs better in terms of accuracy (G-ABC with NN): 97.56%, precision (G-ABC with NN): 67.78%, recall (G-ABC with NN): 96%, and f measure (G-ABC with NN): 94%.

1.6 Organization of the Thesis

This section summarizes the details description of the thesis chapters. This research draft aims to improve the performance of the association rule mining for both the grounded data and data with GT. Ungrounded data refers to data that is not labeled in its original form. In such a scenario, a GT generation mechanism has to be applied to train any system. The rule mining architecture is defined by propagational behavior in the case of the proposed work and detailed related work has been done in Chapter 2. The organization of the rest of the thesis is given as follows.

Chapter 1: This chapter provides an overview of Data mining and Association Rule Mining, discussing the challenges that exist in the field. It also explores the extent and importance of the research being conducted. Through a review of the literature, several prevalent patterns have been identified. From the conducted literature review it can be concluded that the current research trend lies in using soft computing/ swarm-intelligence for more optimized results.

Chapter 2: This chapter illustrates the frameworks that have been used in association rule mining to solve various issues like sentiment analysis, stock prediction, forecasting, etc. which also covers one of the objectives of the study. The literature survey incorporates the study and implementation results of existing state of art propagation-based rule mining architectures.

Chapter 3: This chapter illustrates different feature selection techniques. Tokenization is the process of breaking up each record into sentences through a machine called a tokenizer. The goal of Tokenization is to take each text and break it down into an unchangeable sequence of words or tokens (procedures for which exist). This has significant implications for data analysis, where the analysis of text must be performed independently on each token. After pre-processing the textual data, it was observed that the number of features to be used for the next phase remains the same. So, to mine optimized features further feature selection is implemented.

Chapter 4: The chapter illustrates a comprehensive introduction to Data mining and Association Rule Mining, addressing the difficulties that arise in this area. Different rule mining algorithms including propagational rule mining algorithms have been discussed in detail. Additionally, it examines the scope and significance of current research. Several prominent trends have been discerned through an extensive literature review.

Chapter 5: This chapter presents the proposed methodology. This paper presents rule mining using the Grouped - Artificial Bee Colony Optimization(G-ABC) technique for feature selection and mean-variance optimization for further rule mining. Classifiers are used to train and test the model for both feature selection and rule mining. For performing the experimental analysis of the work Twitter and Baseball datasets were used. The proposed algorithm demonstrated the most optimized for the number of rules generated, the time required for calculation, and getting supplementary normalized information for rule mining. The best performer G-ABC with Neural Network (NN) classifier represents an average of 97.56% accuracy a precision of 61.11, a recall of 96%, and an f-measure of 75% with G-ABC and mean-variance optimization technique with the Neural Network classifier.

Chapter 6: This chapter includes the proposed technique results and discussion. It includes results based on feature selection approaches and evaluation based on classification approaches. Moreover, it also includes the evaluation of multiple simulations. The results show that the G-ABC algorithm using neural networks performs better in terms of feature selection (91 features were chosen out of 100 records), execution time (three seconds), and accuracy (98%). Without any

backing or assurance, the results are carried forward for association rule mining. According to the results, the G-ABC with Neural Network algorithm performs better in terms of accuracy (G-ABC with NN): 97.56%, precision (G-ABC with NN): 67.78%, recall (G-ABC with NN): 96%, and f measure (G-ABC with NN): 94%.

Chapter 7: This chapter concludes the report draft and the citations are made as per the illustrated papers in the reference section. Moreover, a conclusion based on feature selection, classification, and multiple simulations has been represented.

CHAPTER 2:
REVIEW OF EXISTING FRAMEWORKS FOR ASSOCIATION RULE MINING

2.1 Soft Computing

2.2 Association Rule Mining

2.3 Artificial Intelligence

2.4 Machine Learning

2.5 Related Work

Summary

2.1 Soft Computing

In recent years, efficient techniques and tools have been devised for the discovery of knowledge in large datasets. These approaches are suitable to exploit the ability of systems to determine the massive data effectively. The data used for analysis can be imprecise and can be afflicted with uncertainty. However, considering the heterogeneous sources in the form of videos and texts, the data might be ambiguous and conflicted partly. Besides, establishing the relationship and determining the pattern between the data which can be vague or approximate, the mining process needs to be robust, apart from human-like methods. Moreover, the learning process requires precise tolerance and exception with some capability. There must be approximate reasoning capabilities and handling the partial information efficiently. Such properties are formed in soft computing which is different from conventional computing. For instance, **Diwaker *et al.*, (2018**) used software computing techniques for the prediction of faults in software.

The soft computing term is used to represent the consortium of mechanisms and methodologies that work synergistically in some form to process and deliver information in the best possible form [**Kumari**, **2017**]. It is guided by a number of principles to devise the methods to lead to the most acceptable and approximate solution to the formulated problem.

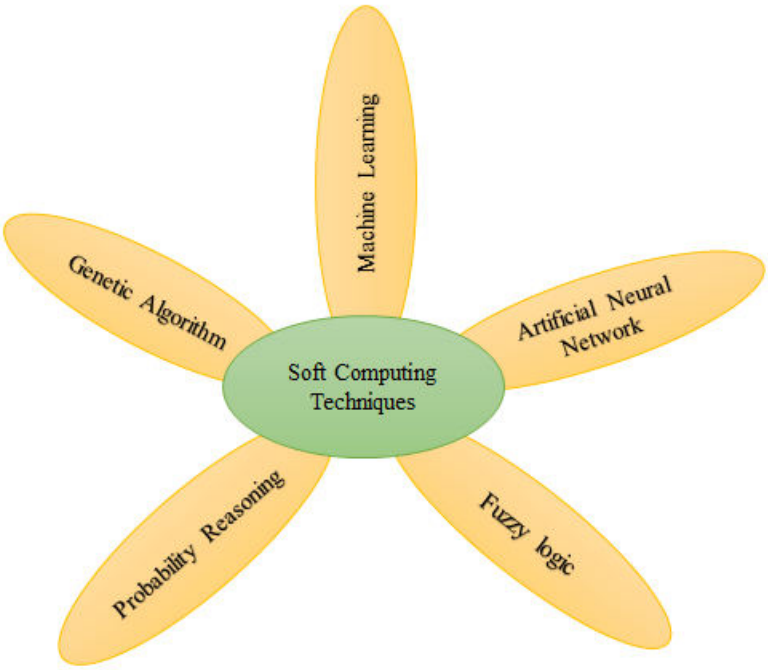

Figure 2.1 Soft Computing Techniques

Soft computing not only involves machine learning techniques such as fuzzy rule sets, neural networks, decision trees, etc. but also involves some natural optimization techniques such as genetic algorithms. These techniques are principal constituents that are complementary rather than competitive and can be considered emerging fields for computational intelligence. The hybridization of such techniques has proved to be a very powerful tool to resolve many mathematical problems [**Damghani *et al.***, **2018**].

- **Importance of soft computing**

The complementarity of different computational intelligence techniques such as fuzzy logic, neural networks, and many more has an important consequence for problem-solving in many cases. The hybridizing of techniques allows probabilistic reasoning such as Neuro-fuzzy systems to fasten the process with a high Machine Intelligence Quotient. The stochastic importance of soft computing techniques is as follows:

- The techniques are used to evolve programs by not relying on others.
- These techniques are used to deal with noisy data.
- The soft computing techniques are used for parallel computations.
- The programs are used to learn on their own.

- The soft computing applications are tolerant to uncertainty in data, imprecision, and data approximation.
- The computation process is very fast as very little time is required for processing data.
- The role model to operate the program is the brain.

- **Applications of soft computing**

The applications of soft computing are found in different emerging fields such as manufacturing, medical imaging, mining, construction, and prediction of the stock market. The different applications are given as follows:

- **Consumer Applications**

Soft computing techniques are used in different consumer appliances such as heaters, refrigerators, heaters, and robotic systems. It has applications in food processing and preparations like cooking rice and microwaving the food and is also applicable in games like poker or checker.

- **Manufacturing Sector**

Instrumentation, management, and data integration are important aspects of industrial businesses. To determine the ideal schedule for the production process, Genetic algorithms, and Neural Networks are applied. Further, to find patterns in a batch of data, networks, and fuzzy logic data, as well as incorporate domain knowledge that can be used to identify faults [**Sarkar**, **2012**]. In the past, the primary focus of research has been on driverless cars. In decades, the majority of transportation decisions have been made under incomplete truths, uncertainty, and imprecision. The efficiency of computational models for dealing with transportation issues lies in inconsistent decision-making.

- **Forecasting**

A key technique for forecasting the future using the past is time series analysis. Forecasting is utilized in scheduling and judgment to efficiently manage the operations of modern organizations. In business, the ability to predict sales is crucial. Business can be estimated if sales can be projected [**Singh**, **2016**].

- **Image Mining**

Image mining is the process of extracting information from photographs. Image feature identification, secret information retrieval, and additional pattern retrieval are all part of image mining, which is an augmentation of data mining [**Khan & Ansari**, **2015**].

- **Applications in the stock market**

People desire returns on stocks that are high and timely using machine learning techniques [**Vanipriya & Thammi Reddy**, **2014**]. People aim to obtain as much as possible and they must be aware of the best times to acquire and sell shares. One can make wise selections by paying attention to the stock markets' operating principles. The automatic control engineering field uses it the most. Soft computing is used to handle the problems of plants that cannot be explained by statistical models.

2.2 Association Rule Mining

The discovery of association analysis among datasets to identify the data objects that satisfy the minimum confidence for support and threshold [**Sherdiwala & Khanna**, **2018**]. For association mining, the item sets have been identified by following the generation of association rules that are strong enough to accomplish the association mining. This also includes the mining of frequent item sets and substructures. Apriori algorithms are also used for the analysis of association. The algorithms for association analysis are classified into condensed representation algorithms and incomplete algorithms for analysis.

- **Classification of ARM**

The taxonomy of Association Rule Mining can be defined based on frequent item sets, sequential patterns, and structured patterns. These are defined as follows:

➢ **ARM based on Frequent Item Set**

The items based on frequent sets can be horizontal layout, vertical layout, and based on project layout [**Duneja & Sachan**, **2012**]. The frequent patterns can be analysed frequently under a certain threshold and considering the minimum support in a business. The frequent item sets can be used in different data mining tasks such as classifiers, association rules, clusters, and sequences. The applications of algorithms using the frequent item datasets are defined as follows:

- Products are arranged on the shelves as listed in the catalogues
- Bundling of products and cross-selling support for different applications,
- Detection of fraud and analysis of technical dependency.

Transaction Database
1:{milk, diaper, bread}
2:{beer, coke, diaper}
3:{milk, coke, bread}
4:{milk, coke, diaper, bread}
5:{milk, bread}
6:{milk, coke, diaper}
8:{beer, coke}
9:{milk, coke, diaper, bread}
10:{milk, diaper, bread}

Frequency of Items			
0 item	1 item	2 items	3 items
0:10	{milk}:7 {beer}:3 {coke}:7 {diaper}:6 {bread}:7	{milk, coke}:4 {milk, diaper}:5 {milk, bread}:6 {beer, coke}:3 {coke, diaper}:4 {coke, bread}:4 {diaper, bread}:4	{milk, coke, diaper}:3 {milk, coke, bread}:3 {milk, diaper, bread}: 4

Figure 2.2 Illustrative Transactional Database

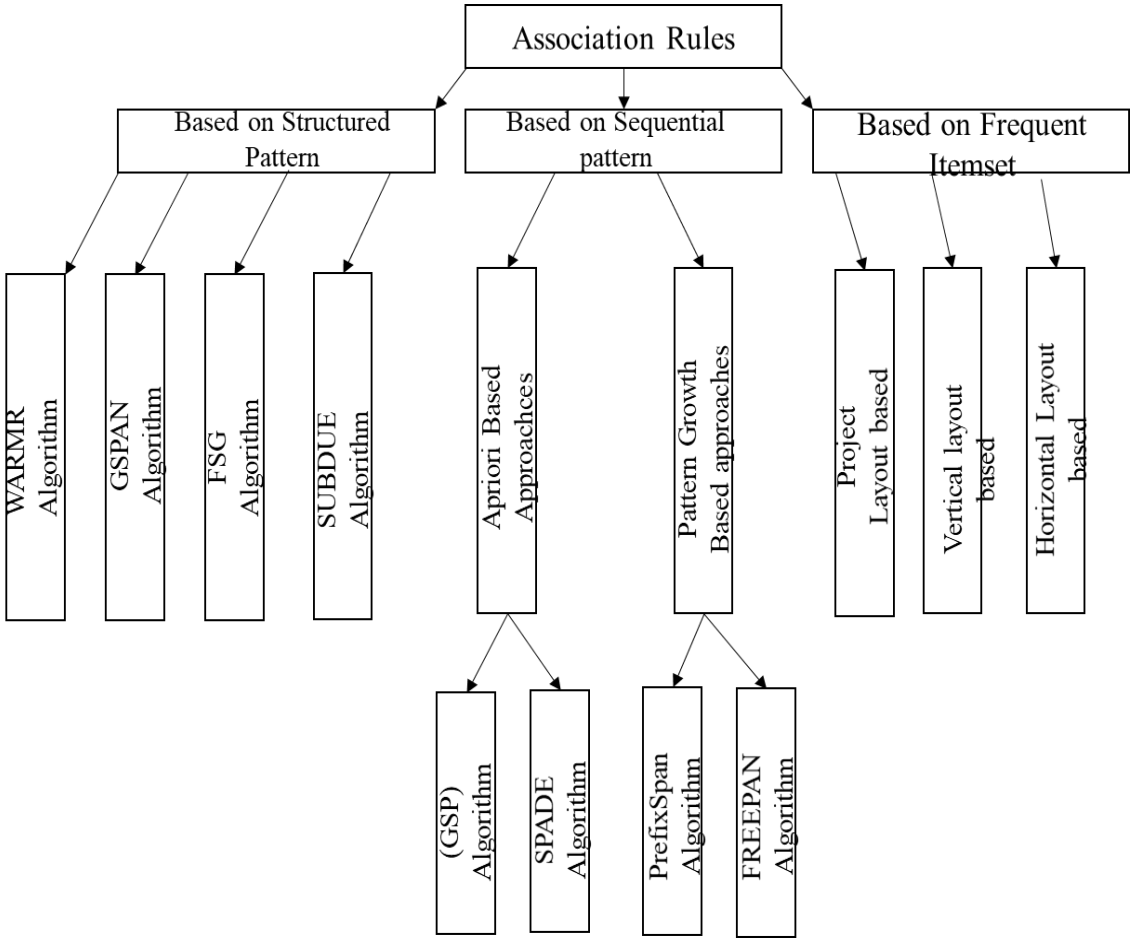

Figure 2.3 Taxonomy of Association Rule Mining [Yazgana & Kusakci, 2016]

Several algorithms utilize a horizontal layout for data mining purposes, including the Apriori algorithm, Direct Hashing and Pruning algorithm, Partitioning algorithm, dynamic itemset counting algorithm, sampling algorithm, continuous association rule mining algorithm, and split and merge algorithm. Each of these algorithms functions uniquely, but all the algorithms use the principle of frequent item sets.

In the case of project layout-based mining, to harvest valuable knowledge, this type of database employs the divide and conquer technique. As opposed to using Apriori methods, it counts the support more effectively. The record IDs are divided by columns in the intended layout. Two different ordering schemes may be used by Tree Projection algorithms: both depth and breadth-first.

- **Sequential Pattern-based Mining**

The patterns have been discovered in a sequential database and events are found sequentially. For example, <v(wx)yx> is a sequence of <v(wxy)(vy)y(yz)>. There are different applications of such mining.

- For the customer shopping sequences customers buy products by maintaining a sequence such as purchasing a Personal Computer and then installing the software followed by memory, printers, and then registering the office papers.
- Analyzing natural disasters and medical treatments
- Determine the telephonic patterns and Weblog click streams
- Science and Engineering process
- Gene's structure and sequences of DNA.

Sequential Mining can be divided into two major groups Apriori Mining and Pattern Growth Mining. Apriori Mining is further categorized as Generalized Sequential Pattern Mining algorithms (GSP, and Sequential Pattern Discovery considering the Equivalent class SPADE). The approaches fall under the category of pattern growth mining considering the large database without the generation of candidate solutions. These are Frequent pattern-projected Sequential Pattern Mining (FREESPAN) [**Han *et al.***, **2000**] and Prefix-projected Sequential Patterns Mining (Prefix Span) [**Pei**, **2001**].

- **Structure Pattern Mining**

More complex patterns beyond frequent itemsets and sequential patterns must be resolved for complex research and commercial applications. For instance, complex patterns include trees, grids, and charts. A significant part of modelling complex structures is the use of graphs. They are utilised in many different applications, including machine learning, chemical bioinformatics, finding the appropriate, video classification, and sequencing. A group of graphs can be used to find recurrent substructures. A survey of graph-based data mining was published in 2003 by Washio and Motoda [**Washio & Motoda**, **2003**]. Several techniques, such as algebraic graphs concept methodologies, have been developed for mining interesting subgraph patterns from graph databases such as SUBDUE which use the substructures to find the subgraphs in a frequent pattern [**Ketkar *et al.***, **2005**]. The frequent Subgraph Discovery Algorithm (FSG) was established in 2004 to determine the relation between data in a large dataset. The Graph based substructure pattern Mining algorithm (GSPAN) is used without candidate creation, the GSPAN algorithm identifies frequent substructures. Patterns, graphs, and grids are just a few examples of the common substructures that can be mined using this approach. More effectively than previous algorithms, GSPAN mines frequently sub graphs [**Yan & Han**, **2002**]. Additionally, it performs better than the FSG algorithm while mining greater frequent sub graphs in a larger graph set with lesser support. The Inductive Logic Programming Algorithm (WARMR) is the first technique used for chemoinformatic data to generate the candidate solution from the Multiple Relations.

- **Application of ARM**

Association Rule Mining can be used in different areas such as Market-based analysis, medical diagnosis, Protein sequences, census data, and maintaining customer relationship management. These are defined as follows:-

- **Market-Based Analysis**

It is one of the typical areas of association rule mining. The market-based analysis is based on the choice of the customer choosing the product considering some probability and such proportion was determined by applying the association rule mining. The knowledge of the customers can be exploited to maintain the products on the shelves as per customer reviews. Thus, customers easily reach these products which can turn the sales rates up.

- **Medical Diagnosis**

Association rule mining can be used to assist doctors in treating patients [**Vijiyarani & Sudha**, **2013**]. For instance, Chang *et al.* determined the illness of the patient using the relational association rules and developed a robust technique to treat and diagnose the patient [**Chang *et al.***, **2021**].

- **Protein Sequences**

There are about 20 different amino acids in the sequences of protein. Each protein is linked with 3- a three-dimensional structure and sequences of amino acids. Islam *et al.* used the Association Rule Mining technique to establish the link between the different acids [**Islam *et al.***, **2018**]. With the help of the ARM technique, where only the useful rules are ultimately sorted out with the aid of interestingness metrics, the authors examined the protein sequences linked to more intricate neurodegenerative protein misfolded illnesses. The presented research uses a quantitative experimental approach to establish more solid association rules between the most dominating amino acids of comparable protein aggregates and to determine the dominant amino acids.

- **Census Data**

The statistical information was linked considering the censuses that may linked to the society. There are different public services such as health, transport, education, and public sector businesses. The information is directly linked to the population and using the association rule mining, one can plan using the economic census data. Zhang *et al.* 2016 determine the relation between the objects to determine the spatial association rules among objects considering the census data [**Zhang *et al.***, **2016**].

- **Customer Relationship Management**

Association rule mining is used to determine the relationship between the credit card customers and the associated bank. This helps to identify the preferences of the customer concerning products, services, and groups as per their choices. Researchers classified the customers into different groups to determine the gold customers [**Khodakarami & Chan**, **2014**].

Furthermore, Association Rule mining is a well-known technique of data mining that was introduced by Agrawal in 1993 and was used to extract the correlation between the data points, frequent patterns, and association between the structures among the test data sets in the repository [**Agrawal *et al.***, **1993**]. The co-related data is passed to the training engine which is again further categorized into two sub-classes namely rule approach and preoperational approach. The rule

approach requires rules to be applied over the input variables with the help of membership functions. Fuzzy logics, Apriori, and decision trees are suitable examples of rule-based approach [**Kumbhare & Chobe**, **2014], [Telikani *et al.***, **2020**]. The preoperational approaches require rules in weight form to predict un-trained data as well. The classification architecture remains completely dependent upon the training mechanism. The training and classification architecture is defined under the subsection of Artificial Intelligence and the ordinal measures are as follows.

2.3 Artificial Intelligence

Artificial Intelligence is a very fascinating concept that is strongly associated with the concept of data mining and rule mining. These terms are being widely involved in present-day research to improve the standards of work as well as an individual's day-to-day life. In rule mining, the learning concept of various machine learning algorithms mentioned in the table is used to analyses the correlation, among the frequently occurring patterns or the categorical data from the databases or repositories.

Table 2.1 Handful of Popular AI Techniques

AI Techniques	Advantages	Disadvantages
Neural Network (NN)	• It can be operated for classification or regression. • It has the capacity to represent Boolean functions (AND, OR NOT). • It can be supportable for noisy contributions. • The illustrations of a neural network can be considered over extra output. • The complex relationship between dependent and independent variables can be easily identified.	• Not simply comprehend the algorithmic structure. • The number of attributes leads to overfitting • The enhanced network structure can individually be computed via research. • The Processing of ANN is challenging in the context of interpretation. • Requires high processing time if the neural network size is large.
Support Vector	• It represents non-linear class boundaries.	• If the training data is not linearly separable, it can be challenging to

Machine (SVM)	• Generally, overfitting is unfolding • The algorithm's computational complexity presents a quadratic optimization challenge, which can be difficult to overcome. • This method makes it easy to handle the complexity of decision rules and the frequency of errors.	determine the optimized parameters for the model. • The algorithmic structure is complex and difficult to comprehend.
Decision Tree (DT)	• There is no need for domain knowledge to build the decision tree. • It reduces the ambiguity for complicated decisions and also assigns exact values to outcomes of enormous activities. • Data processing is easier with high dimensions. • Interpretation is easy. • This algorithm also manages both kinds of data such as; numerical and categorical.	• It is limited to one attribute of output. • Categorical outcomes produced by this algorithm. • This classifier is unstable which means the performance of this algorithm depends upon the category of the dataset. • If the dataset is of numeric type then it produces a complex decision tree.
Logistic Regression (LR)	• Enhanced performance in the small size of datasets. • The outcome of this algorithm can be interpreted as a probability.	• The assumptions of data are required to be compiled. • The only linear solution can be provided by this approach.
K-Nearest Neighbor (KNN)	• Its implementation is easy. • Training is performed in a faster manner.	• It requires a large space to store data. • Very sensitive to noise. • The testing of this algorithm is slow.

Naïve Bayes (NB)	• Performance is good in the small size of datasets if the conditional independent assumption holds. • It is easy to implement. • Produced good results in most of the cases.	• Experimentally, the dependencies exist among different variables. • The assumption of independence among features.
Random Forest (RF)	• This learning approach is widely regarded as one of the most accurate, often producing classifiers with very high levels of accuracy. • Executes efficiently even on large size of the dataset. • Thousands of input variables can be managed with the deletion of variables. • This approach is highly efficient for estimating missing data and can maintain accuracy even when a large proportion of the data is missing.	• In some datasets with noisy classification or regression tasks, overfitting has been observed with this algorithm. • This algorithm is biased towards attributes with more levels for categorical variables with distinct numbers of levels. • As a result, the variable importance scores obtained through random forests are not considered reliable for these types of categorical data.

2.4 Machine Learning

Similar to the clustering technique, association rule mining is also a type of unsupervised learning approach. It finds the correlation and dependency among various data items, identifies the relationship, and finally draws the maps to represent the most profitable solution. The correlation data is mainly passed to two broad categories of techniques:

- Rule based Approaches
- Preoperational Approaches

The Rule based approach specifies some rules that should be applied for the training of the system. The techniques such as decision trees, logic etc. discussed in the AI section are the best examples of rule-based approaches. While the preoperational approaches mainly involve weighted rules that propagate through the layers and are used for the prediction analysis. Further, ML involves the classification and training process for different applications such as rule mining, segmentation, and many others.

Machine learning enables systems to self-program, which is an AI-based field that emerged from the need to teach systems how to train and simulate a response to a situation. It is used for various purposes, such as automating mundane tasks and providing insightful analysis. The ML algorithm, while not specifically designed to predict performance, significantly enhances the effectiveness of training and testing processes. The data is used to be trained and tested in the ratio for analysis. For instance, 70% data is used for training and 30% data is used for the testing phase. This allows the system to become effective in solving the complex problem. To this end, various ML algorithms are used to test and train the system which is illustrated in Table 2.2. The learning process can be supervised or unsupervised depending on the work architecture. For instance, clustering requires the process of labelling the data during the formation of clusters that falls under the category of unsupervised learning. Supervised learning requires data labelling and direct feedback for changes and acknowledgement.

Table 2.2 Machine Learning Algorithms

Name of Machine Learning Algorithm	Features
Unsupervised Learning	No Labels No Feedback Determine hidden structure in data
Supervised Learning	Labelled Data Direct Feedback Estimate future outcome

In terms of unsupervised learning, the main idea is to construct techniques that can take in the experience and utilise scientific calculations to forecast a result while upgrading the results as new data becomes available. ML algorithms that are regularly utilised include; SVM, NB, RF, and ANN [**Nti *et al.***, **2022**].

The data is not labelled in unsupervised data as there is not any labelling process that has been done. The system must identify unknown patterns in the data to obtain the correct answer without being informed. Algorithms must be written in such a way that they can discover appropriate patterns and structures in the data on their own. After testing, the classification process has been done to classify the data.

For the classification process, different techniques have been used such as Support Vector Machine, K-means, Random Forest, Decision Tree, and many more. These are explained in detail in the later sections.

According to Han and Kamber define classification as the process of developing a model that can automatically categorize a group of objects to predict the classification or value of future objects, including missing attributes whose class is unknown. The process consists of two stages. In the first stage, a model is created using a set of training data to describe the properties of a specific group of information categories or classes. This stage is known as supervised learning because the classes or categories of the training samples are predetermined. The second stage involves applying the model to predict the classes of new data or items. [**Han *et al.***, **2002**].

- **Statistical Machine Learning**

When it comes to statistical machine learning, the computation of the error metrics gets involved in the analysis. The predictions and the estimations performed using machine learning techniques are analysed using correlation metrics such as standard error, root mean square error, and standard deviation. Statistical Machine Learning (S-ML) refers to calculating the parametric values using stats viz. numeral values. Mean Squared Error (MSE) and Standard Error (SE) are the perfect examples of S-ML architecture. The validation can also be done using similarity indexes, for instance, cosine similarity.

- **Standard Error (SE)**

A statistical term that employs standard deviation to represent a population accurately by using sample distribution to measure precision. It is used to refer SD of different sample statistics, like mean or median. To compute the value of standard error the following equation is used.

$$\mathbf{SE} = \frac{\boldsymbol{\sigma}}{\sqrt{\mathbf{n}}} \qquad \textbf{[2.1]}$$

Where, SE is defined as standard error of the sample, n defines a number of samples, and σ define as sample standard deviation.

- **Root Mean Square Error (RMSE)**

A measure of the distance between the predicted errors and the regression line, the standard deviation of the predicted errors observed during effort estimation is calculated to determine the distribution. To verify the experimental outcomes, it is computed against the desired output and the estimated output of the project using the following equation.

$$\mathbf{RMSE} = \sqrt{[\mathbf{E_{predicted}} - \mathbf{E_{known}}]^2} \qquad \textbf{[2.2]}$$

- **Standard Deviation [SD]**

It is the parameter that is used to denote the extent of dispersion observed in a set of values. When it is low, it means that the computed values are close to the expected values. It is computed based on the following equation.

$$\mathbf{SD} = \sqrt{\frac{\sum \mathbf{E_{value}} - \mathbf{Mean_{value}}}{\mathbf{Proj_{total}}}} \qquad \textbf{[2.3]}$$

Where, E_{value} is the computed value and $Mean_{value}$ is the mean of the computed values over number of project files $Proj_{total}$.

➢ **Propagational Machine Learning**

Artificial Neural Networks are the best example to illustrate the propagational machine learning concept. The term propagation refers to the movement of data from one layer to another layer in multi-layered architecture. It is generalized that in propagational machine learning, the network learns from the weights applied at each layer. The network contains simple processing elements that are connected having some weights. The elements learn from the data that propagates through the layers while getting refined at each layer under the influence of the applied weights. It mimics the biological nervous system as per both architectures including information processing logic.

To make predictions, the neural network needs to be trained with a large dataset using a suitable learning algorithm to estimate the interconnected weights. The multilayer perceptron network is commonly used for classification tasks. The neural network architecture has a general layout as shown in the figure. The first layer is known as the input layer where data is fed to the model for training and classification. The middle layer, also known as the hidden layer, performs all the computation and prediction. The final layer is the output layer, which provides the analysis results. The movement of data through the layers is termed as the propagation.

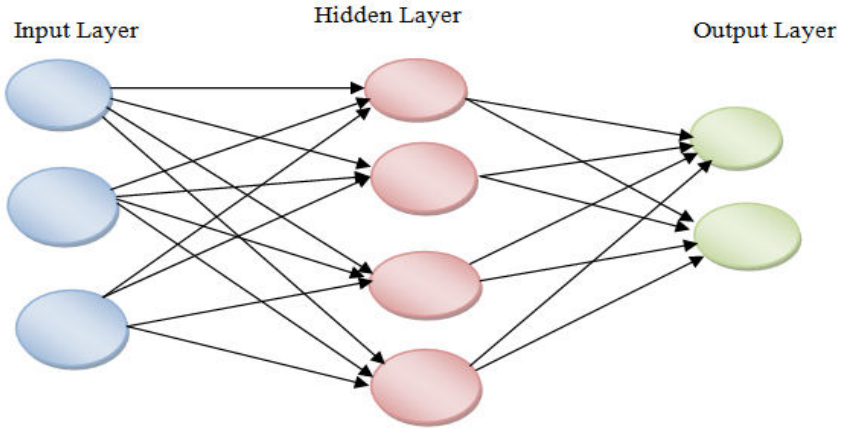

Figure 2.4 Architecture of Artificial Neural Network

Depending on the direction and how the propagation is going on in the neural network, the neural network is further classified into forward propagational, backward propagational, and forward and back propagation neural networks.

- **Forward Propagational Neural Network**

The forward propagational neural network is also known as a feed-forward neural network. Its connections between the nodes do not make the cycle because it goes from the input layer to the hidden layer and the finally output layer does not come back to the previous layer like from the output layer to the hidden layer or input layer.

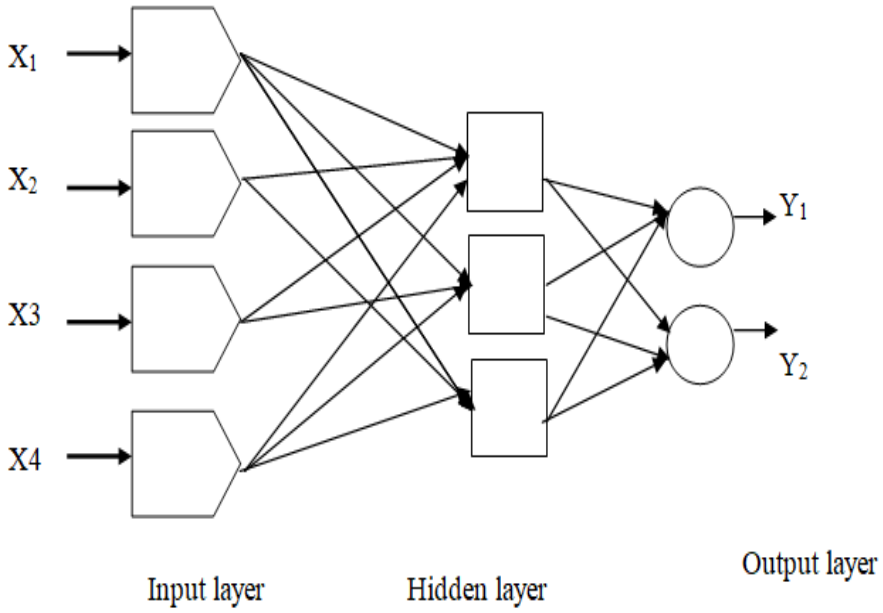

Figure 2.5 Forward Propagational Neural Network

A Forward Propagational Neural Network consists of a minimum three layers of neurons as input layer, an intermediate hidden layer, and an output layer. Generally, neurons are connected in a forward fashion with input units fully connected to neurons of the hidden layer and hidden layers neurons connected with neurons of the output layer. This network is widely used for enormous tasks like recognition of patterns, approximation of function, dynamic modelling, data mining, and time series forecasting.

- **Back Propagational Neural Network**

It a mainly used in artificial neural networks to calculate error which appears at the output layer we can go back and solve these errors. Here the some important reasons why we prefer back-propagation instead of forward propagation is its iterative, recursive , and efficient method for calculating the weights or errors updates to improve in the network until it can perform the task for which it is being disciplined. Back Propagation (BP) is not the network itself, but the instruction or learning algorithm.

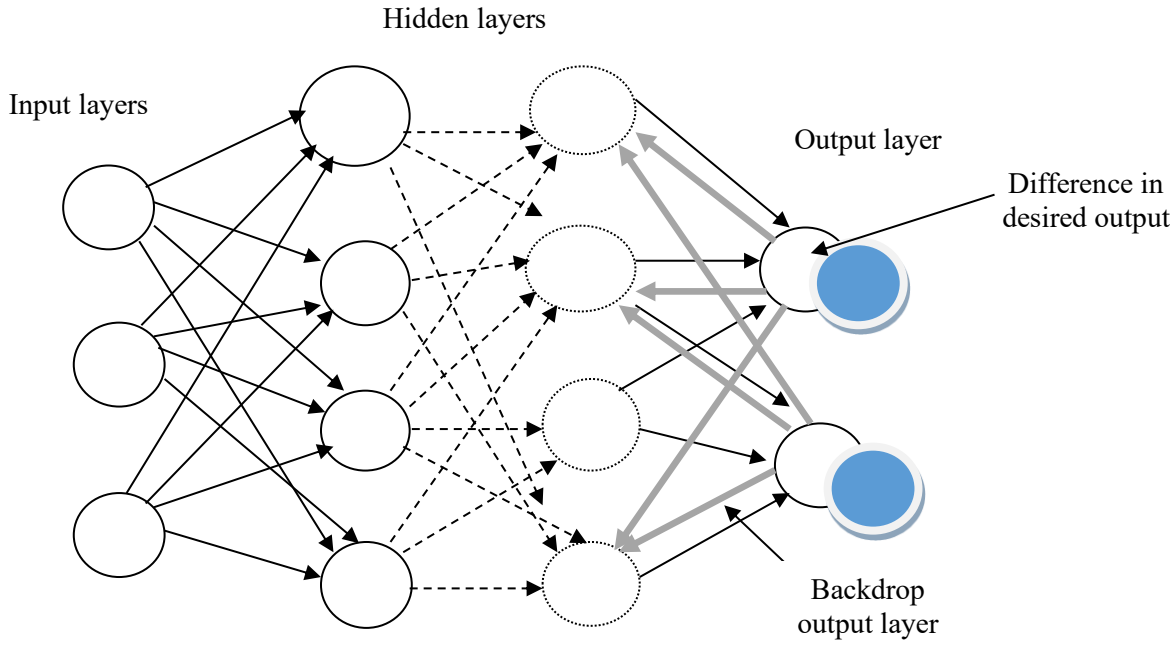

Figure 2.6 Back Propagational Neural Network

To train the network, we need to offer a specific entry to the production called the Target. First of all, the network is initialized by placing all of its weights as tiny normal digits–say between –1 and+ 1. The entry model will then be implemented and the yield will be calculated as the forward

pass. The calculation provides an outcome that is entirely distinct from what is anticipated i.e., the Target, as all weights are arbitrary. Then compute the Error for neurons, which is necessary, this obtained error is then used numerically to change the weights in a way that the error will get smaller. This process is repeated until the error is reduced.

- **Forward and Backward Propagational Network**

This type of propagation involves both forward as well backward propagation based on the feeds obtained from the weights present in the layers. Output error derivatives are propagated back to the network. The fresh input weights and concealed parts of each neuron are then adapted to reduce these mistakes. The learning continues until the objective of success is reached. Minimum square amount or mean square mistake is usually used as a coaching efficiency objective. After the network is trained a new data set that has never been presented to the network will test the network's performance.

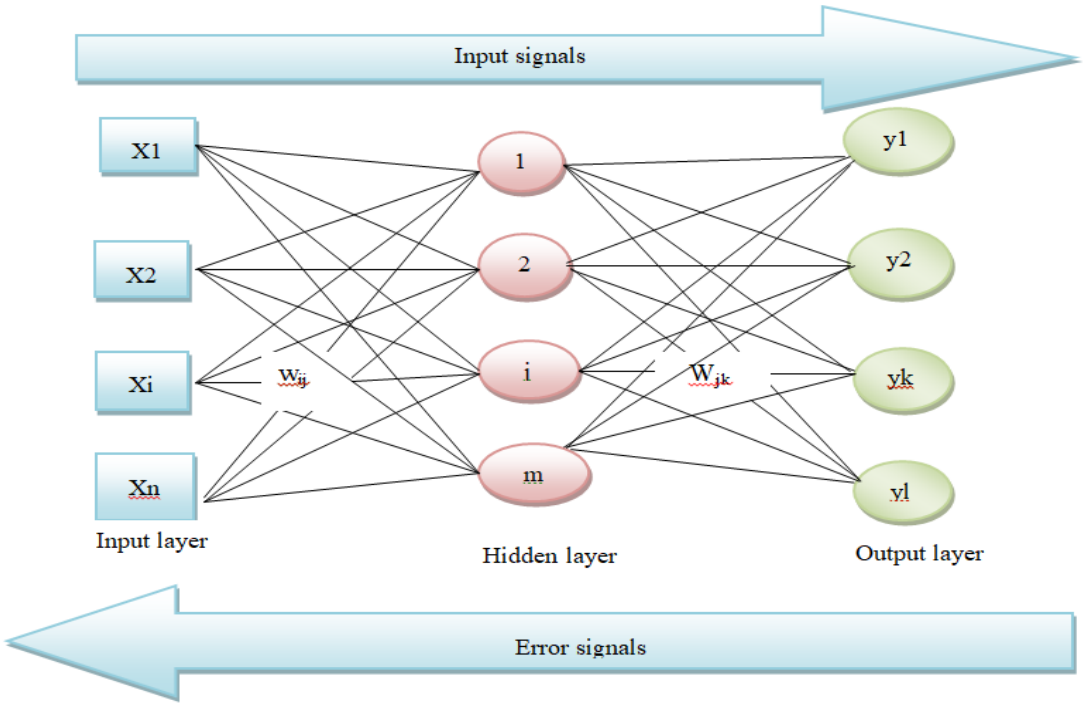

Figure 2.7 Forward and Back Propagational Network

To predict the performance of a network, an error between the output and target is computed. Ability of this type of network depends on some identified factors such as architecture of the network, quality, and quantity of model inputs, the transformation of data and validation of model which are problem-independent.

2.5 Related Work:

Association rule mining architecture is about associating the rules against its input data that varies in a range termed its membership function. This chapter illustrates the traditional and modern way of association rule mining and hence the section is divided into two subsections namely traditional and modern association rule mining engines. The research work is not centric towards any specific GT value and hence multidisciplinary research area that incorporates association rule mining using soft computing for computer science and engineering has been considered in this chapter.

- **Review based on Association Rule Mining**

Huo *et al.*, (**2016**), proposed an improved algorithm to analyse the problems related to the Apriori algorithm. The authors set up the frequent pattern tree structures that were used to maintain the fuzziness in the original datasets and transactions. The incremental strategy was used for implementation and frequent patterns were prioritized for both initial and new patterns. The proposed technique has the advantage of less execution time and memory cost was less when the support threshold was lower in comparison to existing algorithms. The limitation of the study was the weighting methods used that made the system complex.

Pal & Kumar, (**2020),** developed a MapReduce model using the distributed frequent itemset generation and using the association rule mining algorithm. The authors used the distributed integrated technology to generate the association rules and frequent item sets. The mining of the rules in terms of frequent patterns was done in a distributed way and used the association rules with the weighted method. The proposed technique solved the problem of multifarious operation in the case of a large dataset. The limitation of the article was rule mining was difficult in a centralized way.

Bao *et al.*, (**2021),** proposed an effective measurement method to improve the traditional rule mining methods. The authors in this study considered several aspects and then found the defects of the underlying problem. The association rules were reviewed and application in different areas was discussed. The evaluation method in terms of Support and Confidence, Influence, Validity, and many other metrics was discussed. The numerical analysis was presented and different frameworks were compared and verification was done using the public datasets. The accuracy of the existing methods was improved but the limitations of the study as the proposed technique not valid for large datasets, and the robustness in different related fields.

Liu *et al.*, (2021), proposed a parallel Frequent Pattern growth technique using the Spark Streaming for association rules in real time. The authors determined the Support and Confidence, and the Frequent Pattern growth algorithm was developed using the divide and conquer approach. The proposed algorithm was worked in two different steps such as database scanning to determine all the items in the database. The sorting was done in descending order as per the set threshold and the database was scanned. The second step was to construct the Frequent pattern tree and the root node was set as per F-List. The performance metrics in terms of average time were computed for different public datasets. The limitation of the study was that there is a need to improve the proposed FP algorithm using a merging tree that speeds up the process.

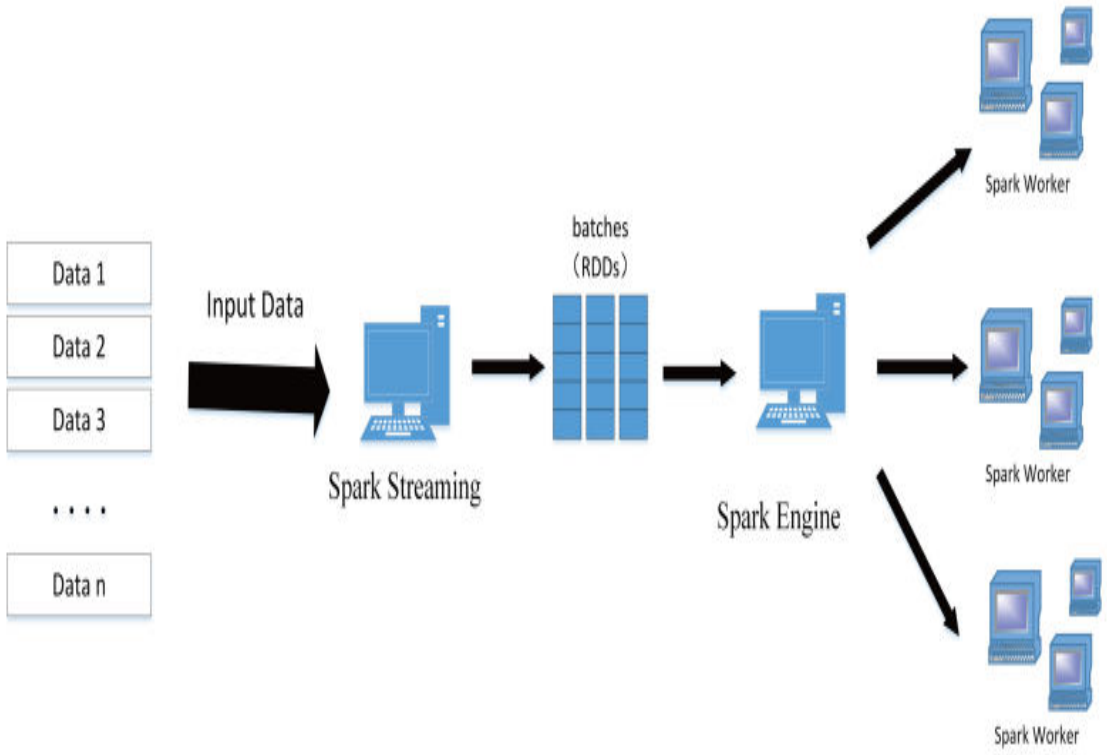

Figure 2.8 Spark streaming Process for data distribution [Liu *et al.*, 2021]

Shawkat *et al.*, (2022), avoid the performance gaps when processing the frequent algorithms in case of huge databases. This paper provides a modified FP-growth technique to improve FP-The proposed approach aims to improve growth efficiency by eliminating the need for repeated conditional sub-tree generation, resulting in a reduction in the complexity of the entire frequent pattern tree. The proposed Mining Frequent Pattern (MFP)-growth algorithm in this study incorporates a header table configuration to improve operational efficiency. To evaluate the

performance of this algorithm, it was compared with other state-of-the-art machine learning (ML) algorithms based on latency, memory requirements, and the effectiveness of generated rules. Further, four experimental series were carried out using various benchmark datasets. The experimental findings support the MFP-growth algorithm's superiority and emphasis its potential for use in a variety of situations. The limitation of the study was that accuracy for real life datasets still need to be computed using the association rule discovery.

Table 2.3 Existing Association Rule Mining with Their Advantages and Limitations

Authors	**Technique**	**Advantages**	**Limitations**
[**Zhang *et al.***, **2015**]	A distributed frequent itemset mining algorithm method was used that uses a matrix pruning procedure considering the spark for big data analytics	• It is used for analytical processes in big data. • It is used to improve the efficiency of iterative computation.	• There is a need for further optimization for more mining. • There is a no support for real-time processing.
[**Zhang *et al.***, **2016**]	A mathematical model was presented for association rules to determine the erroneous data.	• The proposed method is robust against inaccurate data • The proposed method is effective for sensitive data.	The study results were limited as erroneous readings were shown during the experimentation using the sensitive data.
[**Djenouri *et al.***, **2018**]	Frequent itemset mining using the PSO and Bee swarm optimization	The proposed technique is applicable for small, medium, and large databases.	There is a need to develop a parallel extension to decrease the run time.

[**Rajab**, **2019**]	Active Pruning Rules was used which is a new method of associative classifiers worked based on rule pruning for the classification of dataset.	It is used to improve the predictive accuracy and reduce the redundancy for rule sets.	It requires an improvement to deal with the datasets that are sparse variables.
[**Sornalakshmi *et al.***, **2020**]	The sequential minimal optimized for context is a hybrid technique used for mining the association rules that depend upon the working of the Apriori algorithm optimizer.	The proposed technique was used to reduce the quadratic programming problem for each simulation.	The training process of the SVM is very lengthy.
[**Thurachon & Kreesuradej**, **2021**]	A fast incremental FP growth algorithm was used for mining by retrieving the patterns from the dataset.	• The proposed approach is advantageous for a small number of sub-trees. • There is a short execution time.	The constructed subtrees were stored in a large space.

- **Review based on Association Rule Mining using the Optimization Algorithm**

Indira & Kanmani, (**2015),** this work proposes a hybrid GA/PSO [GPSO] technique that combines both genetic algorithms and PSO. Through the careful balancing of exploration and exploitation, the extracted frequent patterns may be predicted with accuracy, leading to consistent performance. The exploitation responsibilities were reduced by GA, and PSO handles the

exploration. When tested on five benchmark datasets at the University of California, Irvine, the GPSO methodology for mining association rules outperforms the individual performances of both GA and PSO in terms of predicted accuracy and consistency

Agarwal & Nanavati, (**2016)**, proposed a multi-objective hybridization of the GA-PSO method, the authors offered an association rule mining scheme. The main benefit of the suggested algorithm is that it integrates the exploration and exploitation jobs by combining multiple objective-GA and multi-objective-PSO, which yields accurate and understandable mined rules. The Bakery dataset evaluation of this hybrid model reveals that it converges four times faster than mono-objective hybridization and generates association rules that are understandable, intriguing, and dependable. The study was limited to providing the desired results due to the complexity of the mutation function.

Heraguemi *et al.*, (**2016),** proposed the cooperative multi-swarm bat algorithm for ARM. The proposed technique was based on an algorithm inspired by bats and modified for the challenge of rule discovery, BAT-ARM. This paper was hampered by the population's lack of communication, which limits the amount of search space that can be explored. However, it features a strong rule-generating process that produces ideal local search results. Therefore, in the suggested strategy, the authors include cooperative tactics between the populations that have already demonstrated their efficacy in the proposed algorithm, to maintain a suitable trade-off between diversification and intensification (Ring, Master-slave). In addition, the authors created a brand-new topology dubbed Hybrid, which combines the Ring method with the Master-slave plan. On nine well-known datasets in the field of ARM, many experiments were conducted, and the effectiveness of the suggested technique was assessed and contrasted with that of other previously published methods. The findings demonstrate the proposal's clear advantage over comparable methods in terms of timing and rule quality. In comparison to multi-objective optimization techniques, the analysis also demonstrates competitive results in terms of quality.

Perera & Caldera, (**2017),** proposed an automated methodology to process and analyse the reviews of customers. The evaluation and categorise the subject as favourable, negative, or neutral is known as sentiment analysis or opinion mining. Since there are many evaluations available in a variety of aspects, it is almost impossible to analyse and extract the true viewpoint from these reviews manually. Opinion mining can be done on three separate levels: report, line of text, and element. The general polarization of the text or sentence is the main focus of document- and line-

of-text opinion mining, which does not accurately describe the key elements of each opinion. Hence, this study focuses primarily on aspect-based opinion mining, a hot topic right now, as it relates to restaurant ratings. The drawback was the method of finding the word opinion still needs improvement.

Jianqiang & Xiaolin, (**2017),** this paper discussed how sentiment classification performance in two types of texts was affected by text pre-processing methods. The authors considered a set of classification tasks and compared the results of six pre-processing techniques using four classifiers, two feature models, and five Twitter datasets. The research demonstrates that the precision and F1-measure of the Twitter sentiment classification classifier improves. The proposed technique was used to enlarge acronyms and replace negation, but eliminating URLs and removing Stop words or numerals. The proposed technique was compared to Logistic Regression, Naive Bayes, and Random Forest classifiers that were more sensitive. The F1-measure of the proposed model using SVM was 0.79%.

Jianqiang *et al.*, (**2018),** offered a word embedding method that uses latent contextual semantic links and co-occurrence statistical properties between words in tweets to produce word embeddings through unsupervised learning utilizing huge Twitter corpora. A sentiment feature collection of tweets was created by combining these word embeddings with n-grams and word sentiment polarity score features. A deep CNN incorporates the feature set to train and forecast sentiment classification labels. The authors experimentally compared the performance of the proposed model with the baseline model—a word n-grams model—and the findings show that the proposed model outperforms the baseline model in terms of accuracy and the F1-measure to classify the Twitter sentiment. The advantage of the proposed model is that error propagation was avoided and classification performance was improved. The accuracy of the proposed sentiment analysis model was 85.63%.

Chiclana *et al.*, (**2018**), decrease the number of association rules by suggesting a new mining technique based on animal migration optimization (AMO), known as ARM-AMO, in this study. The authors predicated on the notion that rules with low support and those that are superfluous were removed from the data. Initially, frequent item sets and association rules were produced using the Apriori method. Then, a novel fitness function that integrates frequent rules and AMO was utilised to decrease the number of association rules. The results show that ARM-AMO significantly reduces the computing time for frequent item set generation, memory for the

generation of AR, and the number of rules generated when compared to the other pertinent methodologies .

Neysiani *et al.*, (**2019),** this paper offers a genetic algorithm-based efficient way for creating cred associations rules with better performances. Evaluations were done considering the Movie Lens data set. The performance metrics such as Rune time, Precision, Recall, and F1 measurement are the assessment's criteria. After conducting experimental evaluations of the proposed multi-objective PSO association rule mining algorithm, it was found that its performance had decreased by approximately 10%. However, the collaborative filtering process remains challenged by issues related to poor accuracy of ideas. Through the use of evolutionary algorithms like PSO and the discovery of association rules, several techniques were developed to improve the accuracy of this method. However, their runtime effectiveness does not meet this requirement.

Sharmila & Vijayarani, (**2021),** the authors used the dimensionality reduction approach in the first step of this study project to significantly reduce the size of the data collection. Low variance and hash table techniques were used in this dimensionality reduction strategy. The suggested approach successfully finds the important database entries and transactions. The suggested technique eliminates pointless data from the transactional database, including items and transactions. The proposed dimensionality reduction method was compared with the extended frequent pattern (EFP) and intersection set theory, as well as a frequency count-based dimensionality reduction method, for both transactions and items. The performance factors were item reduction, reduction in transactions, speeding the execution time, and wider memory space. The limitation of the study was clustering the data in a centralized manner.

Table 2.4 Comparison of Existing Techniques using the ML models

Authors and Citation	**Techniques**	**Dataset**	**Results**
[**Zimbra *et al.***, **2016**]	The authors used feature engineering and ANN techniques for brand-related sentimental analysis.	Twitter dataset	The accuracy for the 3-class problem was 86% while 85% was obtained for 5- the class problem.

[**Kale & Padmadas**, **2017**]	The author used the Naïve Bayes classifier and maximum entropy technique for opinion mining and compared the algorithms for evaluation.	Tweets	The accuracy using the Naïve Bayes classifier was 63.9% while 27.8% was obtained using the Maximum Entropy.
[**Jianqiang *et al.***, **2018**]	The authors introduced the word embedding method through unsupervised learning for sentimental analysis and integrated it with the deep CNN method.	Stanford Twitter Sentiment Dataset	The accuracy using the deep CNN model was 87.36%.
[**Alshari *et al.***, **2018**]	The authors used the Lexicon-based approach and introduced the SentiWordNet to determine the polarity of words which is non-opinion and developed the Senti2vec model.	Movie Review Dataset	The accuracy for positive data was 85.4% and 83.9% was obtained for negative data.
[**Bandana**, **2018**]	The authors proposed the hybrid technique by integrating SentiWordNet, Naïve Bayes, and SVM to describe the heterogeneous feature.	Movie Review Dataset in which 250 samples were trained and 100 samples were tested.	The accuracy using Naïve Bayes was 89% and 76% using the SVM.
[**Ghosh & Sanyal**, **2018**]	The authors introduced the three-feature selection approach such as Sequential Minimal optimization (SMO), Multinomial Naïve Bayes (MNB), and Random Forest	Movie Electronics Product Kitchenware	The F-measure using the SMO was 90.18. The accuracy for MNB was 88.18, 87.73 obtained using RF, and 87.32 was

	(RF) integrated with Logistic Regression (LR).		obtained using the LR.
[**Sumit *et al.***, **2018**]	The authors used the Word2Vec method, Skipgram, and Word to Index with ANN technique for sentimental analysis with the embedding method.	Facebook pages in Bangladeshi language	The accuracy using the Skipgram technique was 83.79%, and 54.40% using Word to Index.

Summary :

Association rule mining represents the mining architecture that is made for a set of input against its membership function values that denote the value range of the input variable. Due to the high volume and variety in the data input values, straight rule-based architecture will have a higher computation complexity and hence propagation-based rule-based architectures were adopted in the later stage of development of software practices. This chapter briefs the data mining and association rule mining architectures that utilizes soft computing methods. It has been observed that a combination of support, confidence, lift, leverage, and conviction may be used to evaluate the interestingness. So, there is a scope for using such measures to generate appropriate rules. Considering more metrics, such as amplitude, may obtain better rules. Therefore, these metrics can be used for better results. Performance may be improved by eliminating the need to determine the extent of the threshold for the criteria of support and confidence. Many models have not used algorithms with categorical datasets. So, this may add some scope to the research. It has been indicated that an increase in support value may give more appropriate rules. Thus, there is some scope to enhance the efficiency of the rules. Hybrid metaheuristics also should be evaluated to generate better rules in the future, further research could explore the effectiveness of alternative machine learning classification algorithms or population-based feature selection meta-heuristics to compare their performance with the proposed approach.

CHAPTER 3:
PRE-PROCESSING OF DATA FOR EFFICIENT RULE GENERATION

3.1 General Architecture

3.2 Primary data vs. Secondary data

3.3 Python and its usage in data mining architecture

3.4 Meta-Heuristic Algorithms for Pre-processing

Summary

3.1 General Architecture

Text data has been a part of any social media platform since the beginning of social media interactions. Later, multi-media data sets were adopted by social media platforms like Facebook, Twitter, etc. Text data is also full of anonymous information that can lead to a different context if it is not analysed in a precise manner. For example: “I am happy today” and “I am sad today”, both have 4 words in common whereas the context of the first statement is completely different from the context of the second statement. If the text is analysed manually, a clear bifurcation can be made based on the analytical meaning studied by the human brain. Due to the high volume and high versatility of data, human beings can't perform the calculations manually and hence a system-aided design (SAD) is required in the same context. All the SAD designs aim to make the rule mapping efficient to conclude a solution for a given input set.

The uploaded data may belong to any specific category and can be majorly categorized into two categories viz. Primary data and secondary data.

In general form, the rule mining architecture can be illustrated using Figure 3.1 as follows.

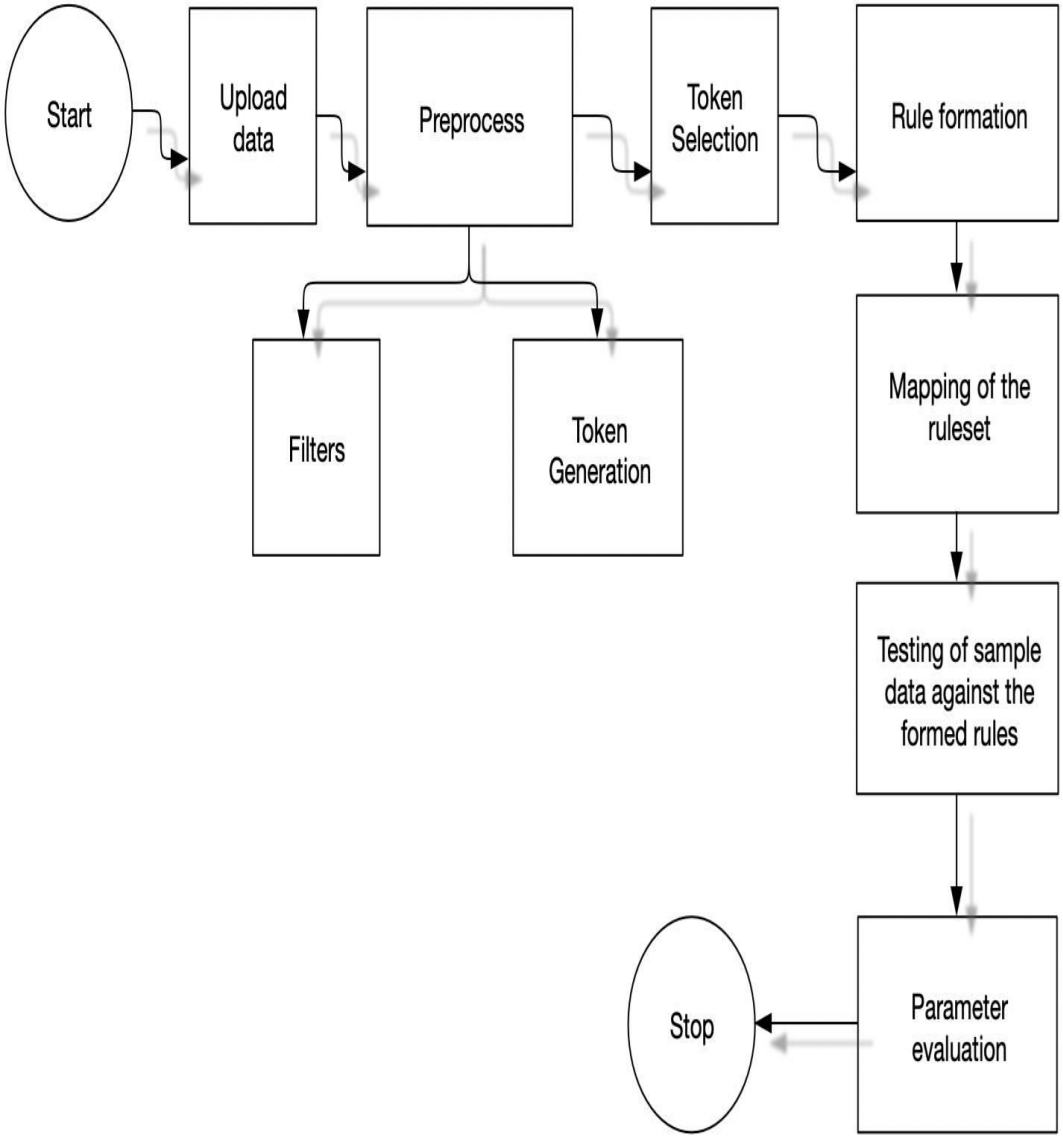

Figure 3.1 General rule formation and mapping architecture

3.2 Primary data vs. Secondary data

Primary data is the data collected by the researchers to research for specific applications. In this, the data collected is structured. The data was collected by the practitioners for the first time and that was not used by any other. The primary data is in the raw form and is more reliable. However, collection of the primary data is an expensive process in terms of time and money. Secondary data is the collection of quantitative data that was collected by other people in surveys and focus groups to obtain consistent results. The data collection method for secondary analysis is very much

different from the primary analysis. The secondary data is unstructured and used sources such as newspapers, books, websites, TV images, and other sources of information. The main difference between the primary and secondary data is highlighted in Table 3.1.

Table 3.1 Difference between Primary and Secondary Data

Primary Data	Secondary Data
Data is collected for specific applications.	Data is collected using surveys and focus groups.
The data collection process is expensive and time-consuming.	Less expensive and requires more time to organize the interviews and surveys
More secure and reliable	Less secure and not very reliable
Can be used by the investigator only	Can be used by any third party
No precaution and the data is not editable	Data is edited and precautions are applied for security.

In the case of the proposed work model, it is not possible to gather primary data as the primary data requires a lot of validations to be done to be utilized in research and it is a humungous task in itself. Hence, most of the researchers use secondary datasets taken from global repositories such as Kaggle, UCI Machine learning, NLM, NSL-KDD, etc [**Amarnath *et al.***, **2016**], [**Mohapatra *et al.***, **2021],** [**Shahin *et al.***, **2021**], [**Subbulakshmi & Deepa**, **2015**]. To be precise on the developed model, the model is tested against various datasets that may belong to various categories or may fall into similar kinds of categories. Taking the validation point quite seriously, the proposed work uses two different datasets from the Kaggle repository and the open set repository itself.

3.3 Python and its usage in data mining architecture

Particularly in the domain of freely accessible tools and libraries, the use of Python in the field of data science has increased to previously unheard-of heights. According to a survey conducted in May 2018 by the reputable website KDNuggets [**Stančin & Jović**, **2019**]] in the category "Top Analytics, Data Science, Machine Learning Tools," 65.2% of about 2000 participants use Python, while its two main rivals RapidMiner and R each get 52.7% and 48.5% of the vote, respectively.

Practically speaking, Python has overtaken R as the preferred programming language for the data science community over the previous three years.

Data preparation from numerous sources for information, such as databases, text files, and streams, as well as data modelling using many techniques, depending on the desired outcome (such as classification, clustering, regression, association rule mining, etc.), are all part of data mining (DM) [**Ward *et al.***, **2018**]. Machine learning (ML) techniques are used by DM to derive new knowledge from the available data. Nowadays, DM is primarily thought of as part of the larger field of data science, which also includes statistics, big data approaches, and data visualisation. Pre-processing stage and data transformation are part of the crucial step of data preparation in the process of analysing data (Wrangler) [**G. Nguyen *et al.***, **2019**].

While wrangling converts the pre-processed information into a data format that can be easily manipulated by the data modelling algorithms, pre-processing seeks to clean, integrate, transform, and reduce the original raw information so that it can be used for data analysis.

- **Advantages of using Python**

The main advantages of using the Python library are given as follows:-

1. There are several reasons why Python has gained popularity, such as its user-friendliness even for those without a computer science background, its vast collection of libraries covering various aspects of data science, and its reliance on NumPy and SciPy wrappers for easy installation of numerous scientific methods written in C and Fortran. [**Browne *et al.***, **1995**].
2. using Python in data mining is its ability to easily incorporate external code into the Python interpreter. This has been particularly useful in the field due to the popularity of the Cython library.
3. Cython, a language based on Python, provides the ability to call C functions, and use C-type variables and classes, and is often utilized in data mining for this reason. [**Behnel *et al.***, **2010**]. Cython can speed up some important areas of code by a factor of several.

➢ **Porter Stemming Algorithm**

Currently, one of the most often used stemming algorithms is Porter's stemming algorithm [**Porter**, **1980**, **2001**], which was first introduced in 1980. The fundamental algorithm for stemming has undergone several modifications and enhancements over time based on various suggestions. The

concept behind the algorithm is that most of the roughly 1200 suffixes in the English language are composed of smaller and simpler suffixes. It consists of five distinct steps where rules are applied until one of them meets the criteria. If a rule is accepted, the suffix is deleted accordingly, and the next step is executed. Once the fifth step is completed, the resulting stem is returned. The algorithm can be generally expressed with the following equation: -

$$< condition >< suffix >\rightarrow< new\ suffix >$$

Porter developed a comprehensive stemming system called "Snowball." The primary goal of the framework is to enable developers to design custom stemmers for various languages or character sets. Currently, there are versions available for numerous languages, including Romance, Germanic, Uralic, Scandinavian, English, Russian, and Turkish.

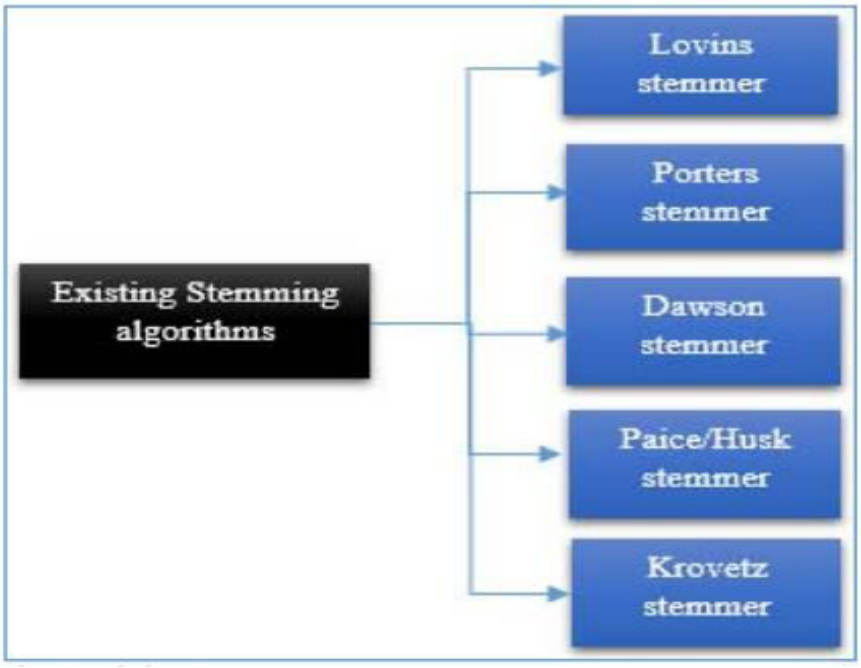

Figure 3.2 Stemming algorithms [**Ismailov *et al.*, 2016**]

The classification of stemming algorithms is illustrated in Figure 3.2. Paice concluded that the Porter stemmer has a lower error rate than the Lovins stemmer, based on the stemming errors. [**Paice**, **1990**]. The Lovins stemmer, on the other hand, is a heavier stemmer that yields superior data reduction [**Lovins**, **1968**]. The Lovins stemming method is significantly larger than the Porter algorithm due to its extremely long list of endings. However, it has the advantage of being faster. It only requires two major steps to remove a suffix with its vast collection of suffixes, as opposed to the Porter algorithm's five. This essentially exchanges space for time.

- **Token generation**

As the simulation work has been done on the Python development environment, the data is tokened using python library tokenizer. The tokenization process is demonstrated in Figure 3.3 as follows.

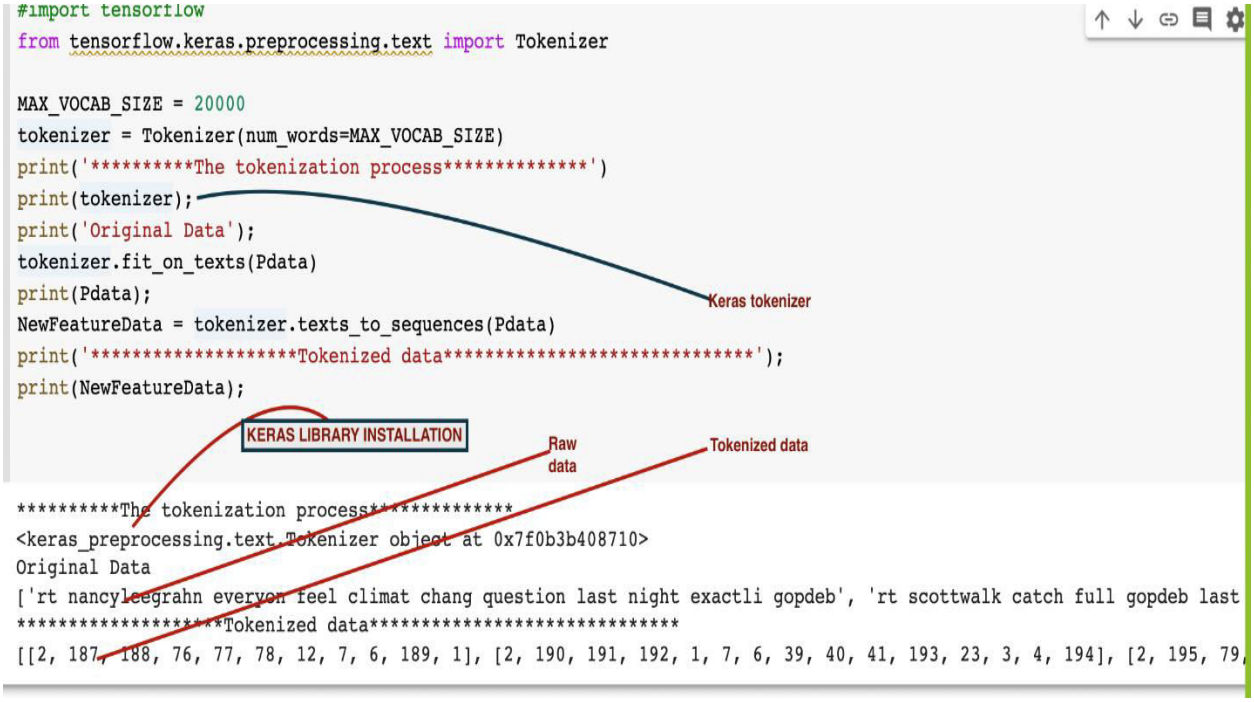

Figure 3.3 Data tokenization

In order to tokenize the data, the proposed work has utilized the Keres library which is a sub library set of TensorFlow. To utilize TensorFlow in the local system, its libraries must be installed on the local system, and for the same purpose, any GUI oriented platform can be utilized. Initially, the proposed work was done on the local host, and for the same purpose, the Anaconda system tool for Python that extends to spider notebook was utilized. As shown in Figure 3.3, the data is initialized with the tokenizer and a curve-fitting policy has been adopted by the tokenizer to adjust the data. Furthermore, each word is tokenized with a unique identification number. Based on the tokened data, a sample size reduction technique has been applied that selects the most significant attribute set from each token category.

Role of sample size selection or reduction

As per Darbin and Watson method, if the data contains more than 5% outlier, the overall prediction of the entire system can be affected up to 50% and hence more co-related data selection mechanism is applied for the selection of the most suitable tokens specified against each class category. The

work of selecting of most suitable token is also the problem of optimization of the current state of data that improves the overall classification accuracy of the data. The term optimization is a mathematical approach that is used to find the minima or maxima value of functions. The methods through which the optimization is carried out are generally acknowledged as optimization techniques. Metaheuristic algorithms are the algorithms that search the data for local minima and local maxima. In the case of a meta-heuristic algorithm, there may be one or more than one solution to a given set of problems. The most suitable optimization methods are illustrated in Table 3.2.

Table 3.2 Benefits of Optimization Techniques

S. No	Optimization Method	Advantage
1	Decision Rule	• It is easy to implement • Efficient
2	Interactive methods	• Easy to communicate • Flexible
3	Mathematical Programming	• Usually, Optimal
4	Heuristic/Meta-Heuristic	• It is easy to implement • It is easy to Program • Provides more than one optimal solution
5	Soft Computing	• Optimal or non-optimal solution • Compatible with other modules.

3.4 Metaheuristic Algorithms for Pre processing

A metaheuristic is an iterative optimization process that guides and modifies the operations of subordinate heuristics to effectively generate high-quality solutions. [**Smith-Miles *et al.***, **2013**]. It is an iterative master process that uses subordinate heuristics to efficiently produce high-quality solutions. These heuristics may manipulate a complete or incomplete single solution or a collection of solutions at each iteration. The subordinate heuristics can range from high to low-level

procedures or simple local searches to construction methods. The concept of a metaheuristic can be used to define heuristic methods for solving various problems. It is a general algorithmic framework that can be adapted to different optimization problems with relatively few modifications.

Metaheuristic characteristics:

- The metaheuristic method is used as part of a global procedure that guarantees to find the optimal or near to optimal solution to a problem by exploring the search space efficiently.
- Metaheuristics utilize heuristics that are guided by an overarching strategy, incorporating domain-specific knowledge.
- Metaheuristic methods are problem independent and more flexible as compared to exact methods.
- Traditional methods are not able to handle voluminous data efficiently. On the other hand, metaheuristics handles high dimensional data in a viable manner.

Metaheuristics are adequate for solving NP-hard problems. These are mainly used for Feature extraction and dimension reduction [**P. Agrawal *et al.***, **2021**]. Metaheuristic techniques have the potential to address multi-object problems. Population-based meta-heuristicis one of the most effective optimization algorithm architectures. Different types of behaviours are observed in nature and hence different algorithms are studied and presented. Some of the popular algorithmic architectures of the population-based meta-heuristic as follows.

a) Artificial Bee Colony (ABC): It is based on the behavior of honey bees. These bees mainly comprise of three types of bees namely employed bee, onlooker bee, and scout bee working collectively [**Dong *et al.***, **2019**].
b) Ant Colony Optimization (ACO): The behavior of ants has inspired a lot of researchers. Ants group to form pheromone solutions and in every iteration, the pheromone solution changes its selection or optimization threshold [**Dorigo *et al.***, **2006**], [**Dorigo & Blum**, **2005**].
c) Particle Swarm Algorithm (PSO): Based on the flying motion of the particles with a rational velocity, PSO was formed. PSO has been a popular pick for a lot of researchers. The fitness function is always dependent upon the processing particle velocity and the distance covered with that velocity in a certain time interval gap [**Freitas *et al.***, **2020**].

d) Cuckoo Search (CS): It is a behavior that is observed in Cuckoo birds for their eggs. There are different variations of CS. Some of them are about laying eggs of the cuckoo bird in another's nest to prevent the eggs. In other algorithm architecture, the cuckoo bird destroys all the eggs if one egg is identified as rotten [**Yang & Deb**, **2009**].

e) Firefly: The lighting of the fireflies has attracted many researchers resulting in a firefly algorithm. The firefly algorithm tries to settle down the far going fly by increasing the current light intensity and tries to keep the group as big as possible [**Fister *et al.***, **2014**].

f) Frog Search (FS): The FS algorithm is about the food search mechanism of the frogs. The frogs are not very choosy about food as they can eat flies, small fishes, etc., but when it comes to finding them, FS is one of the efficient algorithm architectures.

g) Genetic Algorithm (GA): GA is a meta-heuristic inspired by the process of natural selection where the fittest individuals are selected for reproduction to produce offspring of the next generation **[Anandan, 2022], [Othman *et al.*, 2022].**

Artificial Bee Colony (ABC) is a meta-heuristic based algorithm architecture that has been opted for by several researchers and quite of them have been illustrated by numerous researchers [**Bhadoriya & Dutta**, **2015**], [**Sahota & Verma**, **2016**], [**Sarker & Kayes**, **2020**].

3.5 Artificial Bee Colony Optimization

Karaboga invented the ABC algorithm in 2005, which is a global optimisation system that replicates honey bee foraging behaviour [**Ilango *et al.***, **2019**]. In nature, the hive has a division of labour, and forager bees operate together without a central control mechanism to maximise the amount of nectar loaded into the hive. ABC consists of three bees employed, onlooker, and scout bees. Researchers describe the behaviour of real bees and provide a thorough parallel.

- **Foraging Principles of Natural Honey Bees**

Two techniques are developed for the procedure of executing the compilation of honey bees' nectar [**Nguyen *et al.***, **2020**]:

- Recruitment
- Abandonment

Recruitment deals with the participation of bees in the execution and the leave of food sources after their usage is Abandonment. Bees are the same shape and size, but they can be classified according to the bees' mode of operation or their responsibilities in carrying them. The employer's

bees collect information about the source and often go to the hives, waiting in the hive with the former bee for information. Bees that provide information to employers are called onlookers. To crack this information, employers use a unique technique - swinging dance. Swinging is a bee's materialistic community that is essential for collectors because it shows some important information about the sources of food that resemble - the direction, distance, and value of nectar. ABC is also used to handle large datasets for clustering [**Gaikwad *et al.*, 2020**].

ABC is made up of three types of bees namely the employed bee the onlooker bee and the scout bee. The purpose of the employed bee is to collect the food from various food sources. As the employed bees have to collect the food from different data sources, it is not necessary that each food element or component that is brought by the employed bee, suited to the best category.

Summary

Pre-processing of the data refers to removing ambiguities from the input data to produce precise classification architecture. Further, to optimised the selected features, feature selection techniques are used. Feature selection is the process of selecting efficient features for the next analysis. There are various methods to select the features but for feature optimization, researchers are using evolutionary algorithms or soft computing algorithms. Pre-processing is implemented in terms of cleaning the data and preparing the input in a suitable form for the next analysis. Initially, if the dataset is clustered then K-Means clustering is used for clustering the data, then labelled the data. Finally, prepare the dataset in the required format. Researchers have used various techniques to optimized the selected features to get efficient outcome of the complete analysis. Many researchers have using PSO, Ant Colony Optimization(ACO)for feature selection procedures. Some researchers are using machine learning algorithms also for feature selection such as Naïve Baye or using both supervised or unsupervised algorithms.

CHAPTER 4:
A HYBRID APPROACH USING GROUPED ABC FOR FEATURE SELECTION AND MEAN-VARIANCE OPTIMIZATION FOR RULE MINING

4.1 Background

4.2 Rule Mining Architectures/Algorithms

4.3 Propagation-Based Rule Mining Architectures

Summary

4.1 Background

Rule mining architecture is dependent upon the input data value, the processing rule sets, and the way the rules are formed in the system. The ruleset itself is of two types viz. rule base system and propagation-based system. The rule-based system is completely dependent on the set of rules that are formed for the processing but they consume a lot of time if the number of rules is more. In order to understand the concept of latency, consider a situation where “John” a normal human being, has to provide a tip to a waiter “Ali” based on the type of service, food, and ambiance of the restaurant. Now, John has three input variables as shown in Figure 4.1 namely service, food, and ambiance.

Each input variable can have two or more than two membership functions. As in the case of the illustrated example, each input variable has three membership functions. For each query, each membership function has to be analysed and in the current case, at least 7 rules will be formed. In addition to this, there is no re-usability of the generated tip method. Every time, the engine will have to surf all the rules, and for sure that is going to consume a lot of time. Propagation-based learning methods are useful when it comes to latency reduction.

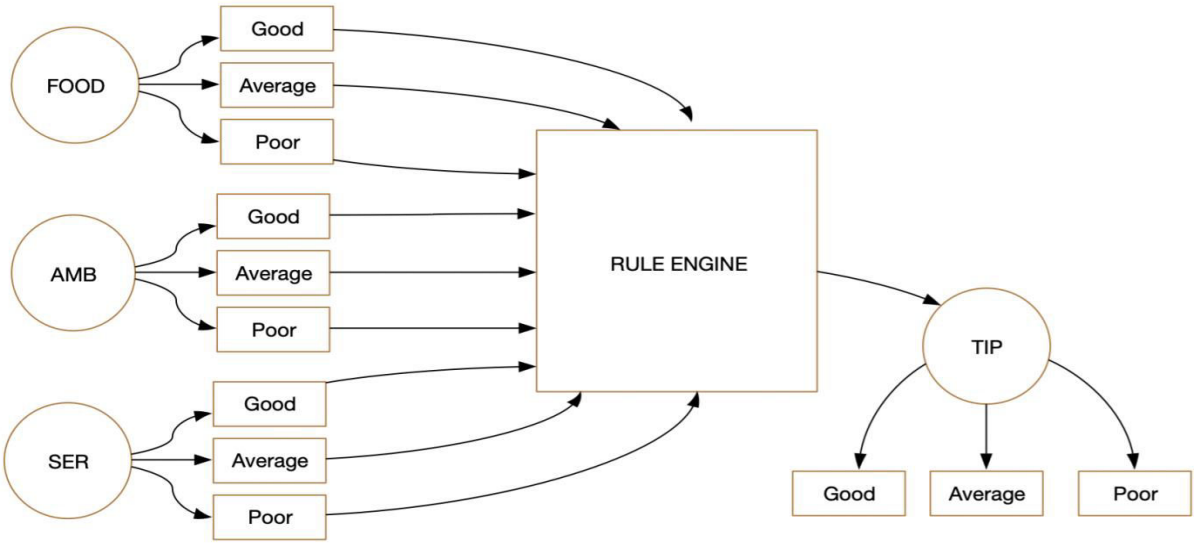

Figure 4.1 Fuzzy rule engine

A propagation-based rule mining architecture contains four essential components as follows.

a. Data
b. Feature extraction method or extracted features
c. Mechanism of propagation engine
d. Validation of the outcome

As the rules in the case of propagation-based mining architecture are incorporated through propagation functions, the prediction time is quite low as compared to straight rule-based architecture as shown in Figure 4.1. The propagation engine converts the input variable's membership function into a property vector using a feature extraction mechanism or algorithm. These features are propagated through a propagation engine rather than getting propagated through a rule engine. The propagation engine uses a propagation function to circulate the data against its Ground Truth value. The propagation engine also has a stopping criterion that decides when the propagation engine has to stop the training. The user data is classified against the GT which in the case of the illustrated example is Tip-Good, Tip-Average, and Tip-Bad. Prior to the illustration of the proposed framework, there are algorithm architectures that are completely based on static rule-based mechanisms and require to be illustrated as follows.

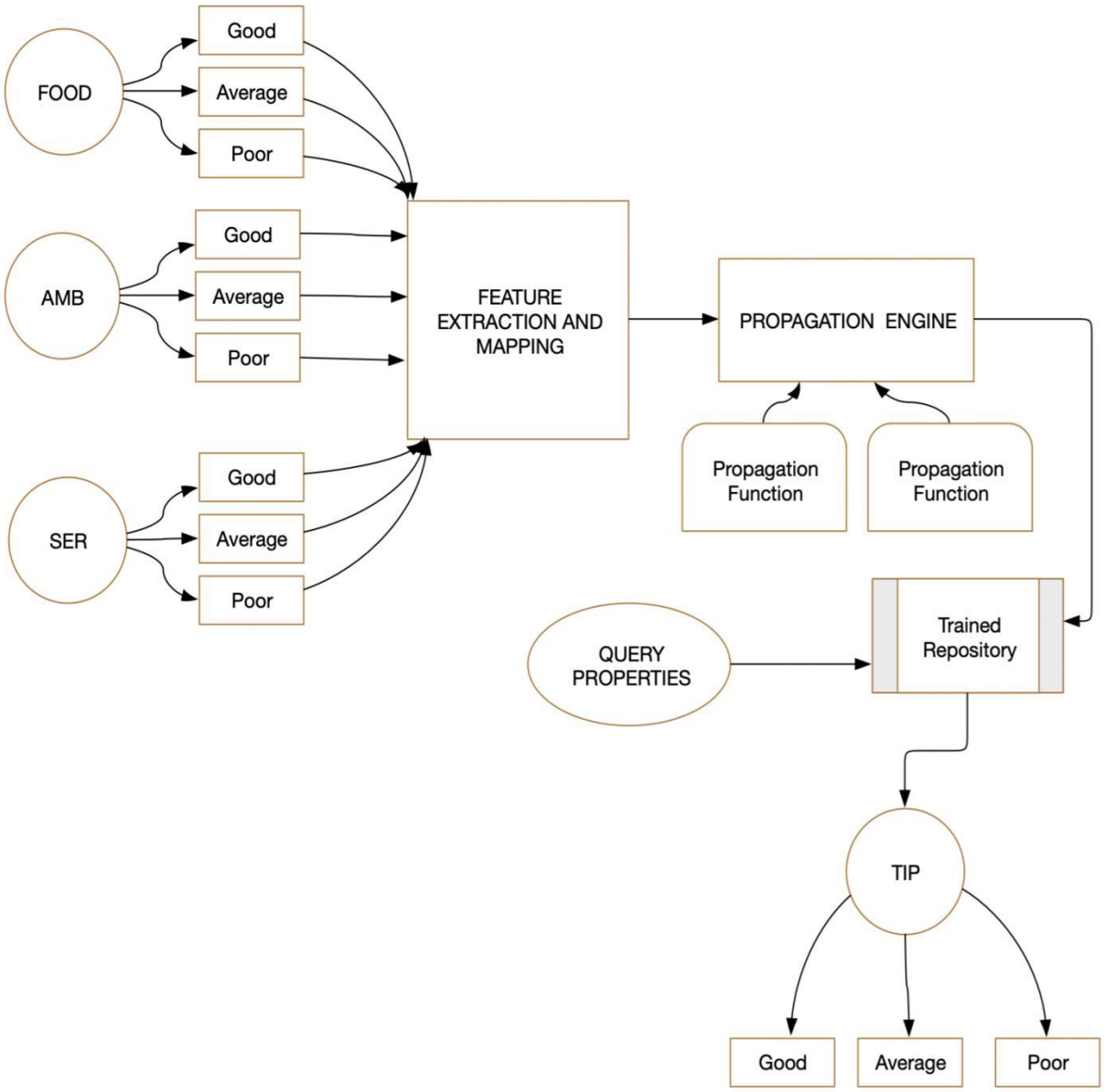

Figure 4.2 Propagation based rule mining architecture

4.2 Rule Mining Architectures/Algorithms

As it is now clear from the studied architecture, a rule mining engine has to be defined with a set of rules against its input variables to produce an outcome. This section briefs the state-of-the-art methods of pure rule mining mechanisms but they are not used in the modern world computation in a very vast level.

- **Fuzzy Logic**

A fuzzy logic (FL) system can manage numerical data and handle linguistic information at the same time [**Serrano-Guerrero *et al.***, **2021**]. A FL is a nonlinear mapping of a scalar output (the vector output) from an input data (feature). The importance of FL is that it is so diverse. There is a plethora of scenarios that can lead to realization of different mappings. This richness necessitates a thorough understanding of FL and the components that make it up a Fuzzy Logic System (FLS) [**Zheng *et al.***, **2022**]. Anonymously, this is comparable to solving the problems in engineering, engineers are always confronted with the challenge of representation. The nonlinear mapping's specifics are established by fuzzy set theory (FST) and FL. It accomplishes this by illustrating how crisp set theory and dual logic may be extended to their fuzzy counterparts. FL imposed causality as a constraint on the development of the FLS because engineering systems are, for the most part, causal. Its purposes and goals, on the other hand, are entirely different. As a result, fuzzy logic is concerned with Modes of communication that are approximate. In general, logic implies that Fuzzy logic reasoning chains are brief in length, and rigor is less crucial than it is in traditional logical systems. In a word, fuzzy logic is a type of reasoning that is based on uncertainty. Fuzzy has a larger expressive power. The fact that it has logic is what gives it its name. Fuzzy logic is the formal foundation of approximate reasoning, with exact reasoning being considered a limiting instance [**Gupta *et al.***, **2019**].

The FLS is in which imprecise data and vague statements are fed as input and decisions on that statement are considered as output. Fuzzy logic is unique in that it attempts to emulate imprecise forms of reasoning, which are essential to human decision-making in situations where ambiguity and imprecision exist. Unlike traditional logical systems, it can draw an approximate response to a question from a body of knowledge that is incomplete, vague, or not entirely reliable.

- **Apriori Algorithm**

The Apriori algorithm was developed by Agrawal and Srikant in 1994 and is a popular algorithm widely useful for data mining applications [**Agrawal *et al.***, **1994**]. The algorithm used the candidate generation to mine frequent item sets. Apriori is the basic algorithm of Association Rule Mining and it is used to boost the applications in data mining [**Liu**, **2010**]]. Apriori is one of the top data mining algorithms and the various characteristics of data such as volume, velocity, and variety. The conventional data mining algorithms and techniques are efficient in mining data which is not scalable and efficient to manage the big data. Different architectures and technologies such

as MapReduce and Hadoop are adopted for the analysis of data [**AlZu'bi *et al.*, 2018],** [**Singh *et al.*, 2018**].

Apriori algorithm is used for an iterative process useful to alternate the important tasks. The first one is used to generate the candidate solution from frequent item sets having previous iterations and the second one is used for the scanning of databases that support candidates against each simulation round. For K^{th} iteration ($[K \geq 2$), candidate k itemsets has been generated S_K from frequent itemsets (K-1) (F_K) and then k itemsets for each iteration has been checked against the candidate solution in S_K which is used to support the counting. Candidate itemsets (S_K) has been obtained by conditionally joined (F_{K-1}) which is used for pruning the itemsets which is not used to satisfy the Apriori property. As per this property, all the itemsets of candidate solution has been used to remove from candidate itemsets if anyone subset (K-1) that are not present in (F_{K-1}). Apriori algorithm is also used for association rules in mining applications [**Al-Maolegi & Arkok**, **2014**].

- **Decision Tree**

Data mining is the process of removing data from a collection of data and translating it into a comprehensible structure. It is a statistical process that combines techniques from artificial intelligence, computer vision, statistics, and distributed databases to find patterns in massive data sets. You can sort through all the disorderly and repeated noise in your data with data mining. Understanding the pertinent data and effectively utilising it are also helpful in determining the likelihood of results. Thus, data mining quickens the process of making wise choices. There are five different processes of data mining such as Anomaly Detection, Association Rule Mining, Clustering the data, Regression analysis, and data classification [**Sharma & Kumar**, **2016**].

A data mining function called classification places objects in a collection into specific groups or classes. Determining the class label for each occurrence in the data is the goal of classification. A classification model, for instance, can assist in classifying bank loan applications as safe or dangerous. Decision tree induction, rule-based methodology, memory-based learning, Bayesian networks, neural networks, and support vector machines are some of the different categorization techniques utilised in the field of data mining [**Gupta *et al.*, 2017**].

Decision Tree is the most widely used supervised classification technique which is comprised of learning process and classification. The classification process using Decision Tree is simple, convenient, and fast applicable to any domain. Decision Tree is the decision support tool which is used to support the decisions using a tree like graph and models. It is generally a classifier in the

shape of tree having different nodes such as leaf node and decision node. Researchers used the decision tree due to following reasons:

- Decision Trees are simple to interpret and understand and can be visualized in any form.
- Decision trees require very little data preparation compared to other procedures that often necessitate data normalization, the creation of dummy variables, and the handling of missing values.
- The computational cost of using decision trees for data prediction increases logarithmically with the amount of training data.
- Compared to other approaches, decision trees have the advantage of being able to handle both categorical and numerical data.
- Multi-output subjects can be handled through decision trees.
- Decision trees use a white box model, which means that the output is often binary, making it easy for Boolean logic to explain the outcome as either yes or no.

- **Types of Decision Tree**

There are two types of Decision Tree such as Classification tree and Regression Tree which are illustrated as follows:

- Classification Tree
- Regression Tree

CART is a combination of classification and regression tree which was proposed by Brieman in 1984 [**Gupta *et al.***, **2017**]. The classification tree was built using the attributes that are splitted in a binary form. However, CART is also used for the analysis of regression using the regression tree. The regression feature of the CART is considered in predicting a dependent variable for the given set of time. To process and support the nominal attribute data in a continuous form, there is an average speed of the CART. The advantages of classification and regression tree are given as follows:

- The missing values can be handles automatically using the surrogate splits.
- The combinations of continuous and discrete variables are used.
- The variables are selected automatically using the CART.
- The interaction between the different variables can be established using the regression analysis.

- The variation of CART as per monotonic transformation is almost negligible.

Apart from the advantages, there are also some disadvantages of using the classification and regression mechanism.

- The use of classification and regression may impact due to instability in decision trees.
- There is only one variable used for splitting.
- The classification is non-parametric.

➢ **Genetic Programming**

A genetic programming is a predictive strategy that solves optimization and forecasting issues by choosing, aggregating, and modifying the intended variables progressively utilizing mechanisms similar to biological evolution[**Kumar *et al.***, **2007**]. It's an example of stochastic gradient descent. The focus on the employment of the "crossover" controller, which execute the process of blending optimal solutions, similar to the role of crossing in wildlife, is a distinguishing aspect of the genetic algorithm.

To search for the optimal feature set among the available population, an accurate representation of features must be performed, and all candidate features must be encoded within a chromosome. A total of "Q" features is selected from the "P" dimensional dataset, and the precision value of each candidate feature among one of the "N" chromosomes is determined. A series of steps are followed to minimize the error in fitness value and determine an optimized value. The offspring obtained from selection, crossover, and mutation are considered parents and are responsible for the next generation. This process of generating the best offspring continues until the desired feature subset is obtained. If the selection criteria, such as reaching a maximum number of iterations or having the chromosome value the same as the population size, are met, the genetic algorithm process is terminated. The decision to stop the process is made based on the criteria that are set for the algorithm. The steps followed for the genetic algorithm are provided below;

1. [Begin] Create a randomly generated population of n chromosomes (suitable solutions for the problem).
2. [Fitness] Determine each chromosome y's fitness f(y) in the search space
3. [Increased population] Repeat the steps above to create a new population.
4. [Selection] 4 Choose two parent chromosomes from a population based on how fit they are (the better fitness, the bigger the chance to be selected).

5. [Crossover] crosses the parents to produce new offspring with a crossover probability [children]. If no crossover occurs, the offspring is a carbon duplicate of the parents.
6. [Mutation] generates new offspring with a mutation chance (position in chromosome).
7. [Accepting] Place new offspring in the new population.
8. [Replace] Algorithm further initiated using a new population generated in the search space.
9. [Test] Best solution has been returned if the termination condition has been satisfied.
10. [Loop] Repeat the above steps by following the step 2.

Workflow of GA

- **Initial population**

Randomly, the initial population has been generated and the only criterion is a sufficient diversity of individuals so that the population does not fall into the nearest extreme.

- **Fitness Assignment**

This process evaluates the fitness measure or metrics for each chromosome in the population through fitness function.

Selection

This is the process of evolution in the population, where our main aim is to have best fitness value of the offspring. The best fitness value will give us more chances of survival of the offspring. Thus, this method focuses on selection of parents, i.e., a pair of chromosomes is selected to breed. The result of the breeding is expected to be an offspring with maximum fitness. Thus, the chances of selection of a chromosome as a parent are higher if its fitness measure is higher.

- **Crossover**

In this procedure, a pair of chromosomes, chosen as parents is operated upon by crossover strategies with crossover probabilities as an important metric for making members of the new population. The general workflow of GA is illustrated in Figure 4.3.

- **Mutation**

In this step, each child is applied a mutation operation at each locus with the deciding metric as mutation probability. At this point there is an exit condition, if the population generated till this step satisfies the end condition, then the whole algorithm is stopped and the corresponding population is presented as the desired solution [**Katoch *et al.***, **2021**]**.**

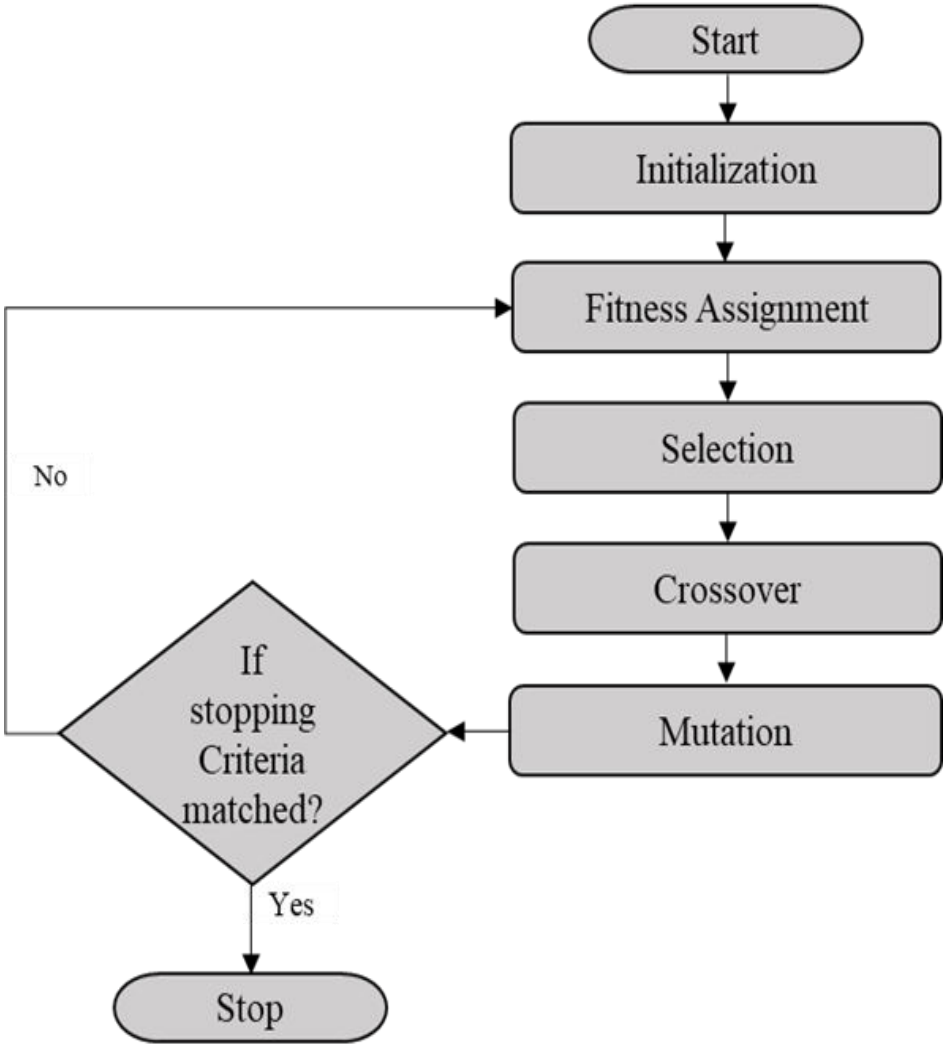

Figure 4.3 Workflow of GA

4.3 Propagation Based Rule Mining Architectures

Due to advancements in the propagation-based architecture and reduction in the computation complexity, the proposed architecture uses propagation-based rule mining architecture. Recent times had seen tremendous revolutionised work in the field of Artificial Intelligence (AI) [**Cioffi *et al.***, **2020**], [**Holzinger *et al.***, **2018**], [**Makridakis**, **2017**]. The proposed work is also inspired form the achievements integrating AI in various research works [**Mohanapriya & Lekha**, **2018**]. In the proposed mobile malware detection system four machine learning classifiers are used namely, Naïve Bayes, k- Nearest neighbour(k-NN), Support Vector Machine(SVM), and Artificial

Neural Networks [**Amrani *et al.***, **2018**], [**Dey *et al.***, **2020**], [**Hammad & Al-Awadi**, **2016**; **Kim *et al.***, **2019**], [**Madasu & Elango**, **2019**].

- **Naive Bayes (NB)**

  The Naive Bayes classification approach is based on the probabilistic method inspired by Bayes' Theorem, which assumes that the predictors used in the model are independent of each other. This model is straightforward to construct and is effective in handling high-dimensional datasets. [**Singh & Kumar**, **2017**]. It is a probabilistic classification method based on Bayes' Theorem, which assumes that the predictors used are independent of each other. It is known for its ease of construction and efficiency in dealing with high-dimensional datasets, as well as its simplicity compared to other complex classification methods. [**Chen *et al.***, **2020**]. In the next step, the classifier model is created while considering the inputs for all the possible values of x and extracting the output that exhibits the maximum probability. The expression can be represented using the following mathematical equation:

- **Parameter estimation and vivid Naïve Bayes models**

For the estimation of the parameters for a defined distribution one needs to assume non-parametric models for the features extracted from the given training set [**Soria *et al.***, **2011**]. The assumptions made on the feature distributions are studied under event models.

- **Gaussian Naïve Bayes**

It is the Naïve Bayes to deal with the real valued attributes on the assumption of normal or Gaussian distribution. It works by estimating the values of the standard deviation and mean of the training data set. In other words, the means, standard deviation, and probability of each class are used in this approach.

Let's consider a variable 'a'. The mean for each class value can be represented as follows:

$$\boldsymbol{mean_a = \frac{1}{n} * sum_a} \qquad \textbf{[4.7]}$$

Where, the number of instances is represented as 'n' and values for input variable in the training data set is 'a'. Standard deviation is calculated as the square root of the difference between each value and the mean as follows:

$$\boldsymbol{sd_a = \sqrt{\frac{1}{n} * sum[\, a_i - \, mean_a{}^2]}} \qquad \textbf{[4.8]}$$

Where, a_i is the value of 'a' for i[th] instance, and the mean value is taken from the earlier equation and inquired. Under ideal conditions normal distribution results in bell shaped curve as shown in Figure 4.4 with maximum density distribution around zero mean value.

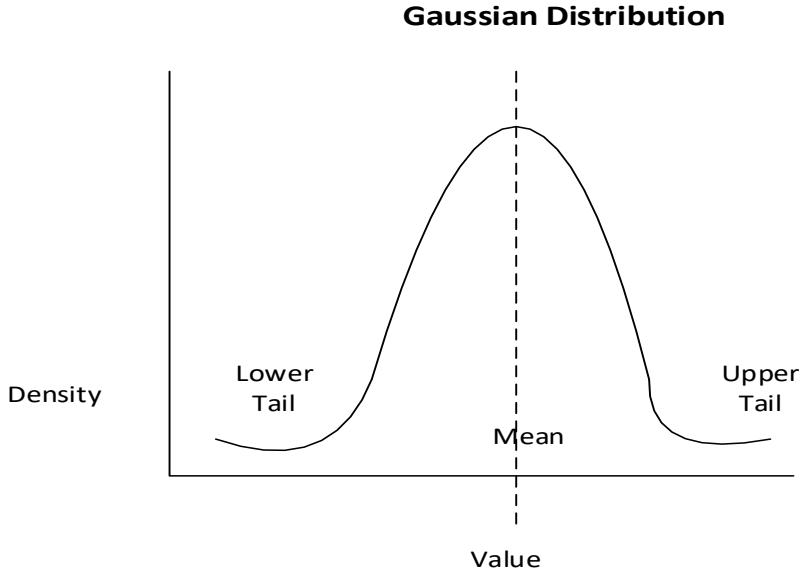

Figure 4.4 Gaussian distribution for zero standard deviation

Predictions are made using Gaussian Probability Density Function using the following relation:

$$\boldsymbol{pdf}_{[x,\ \boldsymbol{mean_a}, \boldsymbol{sd_a}]} = [\mathbf{1}/\sqrt{\mathbf{2} * \boldsymbol{C} * \ \boldsymbol{sd_a}}\ ] \ * \mathbf{exp}\ [-[(\boldsymbol{a_i} - \ \boldsymbol{mean_a}^{\mathbf{2}}) \ / \ [\mathbf{2} * \boldsymbol{sd_a}^{\mathbf{2}}]] \quad \textbf{[4.9]}$$

Where, Gaussian PDF is represented by $pdf_{[x]}$, means is represented by $mean_a$, standard deviation is represented by sd_a, a numerical constant is represented by 'C', Euler's number also a numerical constant is represented by exp [4.9].

Application of Naïve Bayes Algorithm:

- **Real time predictions**

Naïve Bayes results in enhanced speed of predictions making it possible for real time prediction and analysis.

- **Multi class prediction**

The algorithm works by giving due importance to the posterior probability of multiple classes available for the considered variable. As such, it offers application in prediction that involves multiple classes.

- **Text classification**

Naïve Bayes could successfully deal with multiclass problems and hence exhibit potential applications for text classification in terms of success rates of classification in comparison to similar algorithms. It is widely used and well-known applications are sentiment analysis to predict positive and negative customer responses and spam filtering to detect email spam.

- **Recommendation system**

Naïve Bayes has been successfully employed in collaboration with other filtering algorithms for the designing of recommendation systems. Such systems are based on the data mining strategies and machine learning approaches to make estimations related to the likeness of dis-likeness of the considered resource.

Merits of Naïve Bayes approach

- Naïve Bayes offers speedy predictions even for multiple class-based features.
- When features are assumed to be independent of each other, they performs better than the logistic regression models while requiring smaller training datasets.
- It exhibits enhanced performance with categorical variables rather than numerical variables. A normal distribution is used as a generalized assumption to deal with numerical variables.

Limitations of the Naïve Bayes approach

- Zero Frequency: It can be understood while considering a case of a variable that is present in test data and missing in training data. The employed prediction model will be unable to perform predictions for this situation and will be assigned zero probability. This situation can be successfully dealt with with the application of smoothing approaches like Laplace estimation.
- The assumption of the existence of independent predictors is another limitation of this approach because in reality, it is impossible to have such an ideal condition.
- The probabilities estimations at times may be unreliable.

➢ **Support Vector Machine and Multiclass SVM**

SVM is an efficient technique that has been designed to solve complicated issues. SVM is quite good at distinguishing between the two clusters. It is employed for the classification and cross-validation of the clustered made using k-means. The next section is a general description of SVM

[**Sriram *et al.*, 2015**]. SVM is a type of supervised machine learning that uses a collection of information supplied as training data that corresponds to one of several groups, the Training and testing method creates a framework for predicting the classification of a current instance. SVM excels at summarizing issues, which is the goal of learning algorithms. Statistical learning theory is used to investigate the difficulty of getting information, generating forecasts, and forming decisions based on a set of data. In computational learning theory, the problem solution for this is as follows.

The main aim is to compute a function 'f' that reduces the error formulated as follows as

$$\int \mathbf{V}[\mathbf{b}, \mathbf{f}(\mathbf{a})\mathbf{p}(\mathbf{a}, \mathbf{b})\mathbf{dadb} \qquad \textbf{[4.10]}$$

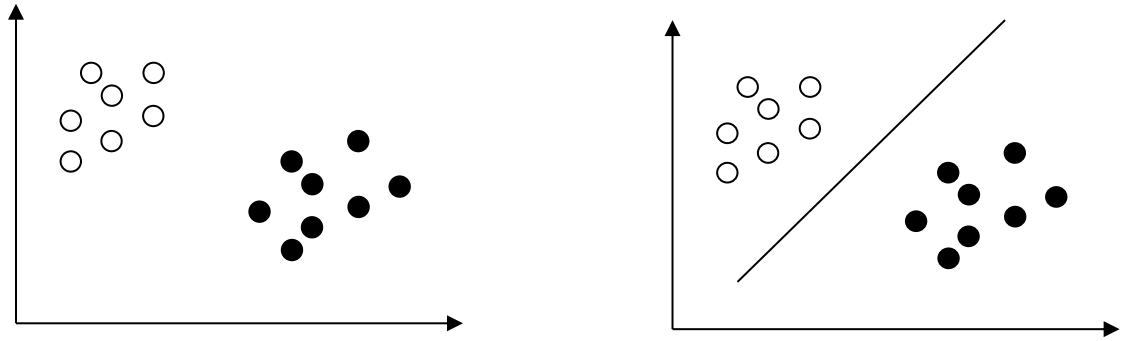

Figure 4.5 (a) Two Types of Dataset **(b) Classification**

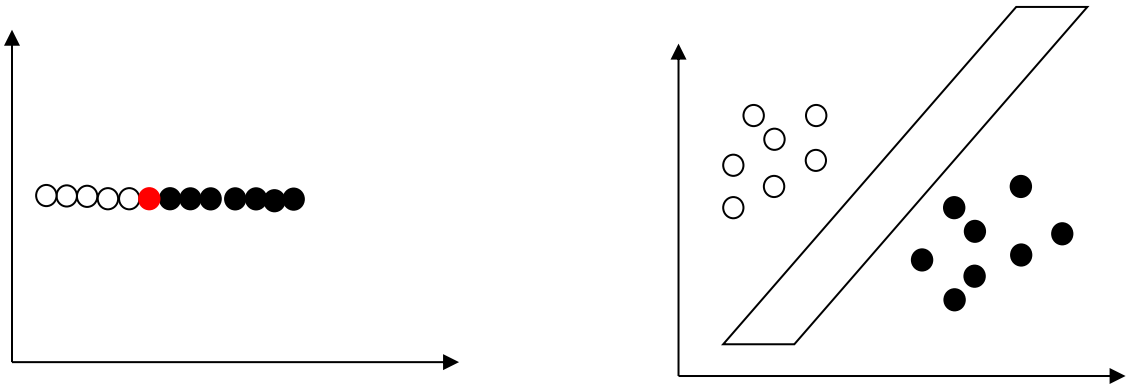

Figure 4.6 (a) Linear Classification **(b) 3D representation**

ML techniques were employed to determine the depiction of uncomplicated variables. As a result, the primary goal of learning is to generate a premise that accurately classifies the classification model, and the early learning algorithm is designed to discover this specific representation of the

data. Its generalization relates to the capability to accurately classify data that isn't part of the training set. The four major points can be used to describe the core notion of SVM.

People can quickly discriminate among various data kinds as presented for each test, but it is much harder for a system to discriminate and display. Figure 4.4 contains two unique data categories, which the researcher intends to classify. Although it could be in the field of view, it is quite straightforward to visually categorize with the naked eye in this scenario. A KF that distinguishes these material kinds, on the other hand, can be used to identify these two distinct classes. For the categorization of 2D data, a vertical line is added among different datasets and represented in figure 4.5 (b) and linear classification in Figure 4.6 (a) and (b).

- **K-Nearest Neighbour**

KNN is an unsupervised learning technique and the classification using the KNN is very slow. The decision about the neighbours is a slow process that accepts only numerals. The speed of KNN is faster in comparison to the decision tree. KNN, also known as k-nearest neighbor classification, is a non-deterministic algorithm that doesn't always return the same results and doesn't perform well with noisy data. It has been widely researched for over 40 years in the field of pattern recognition. KNN uses the K most similar instances from historical data to classify new records. To accomplish this, the algorithm first calculates the distance between the new instance and the training samples, then finds the K nearest neighbors. It then assigns the category to which each neighbor belongs and compares it to the category of the new sample. If all neighbors belong to the same category, the new sample will be assigned to that category as well.

The K-nearest neighbors (KNN) algorithm assigns a category to a test sample X based on the category of its K nearest neighbors among the training instances. In other words, K neighbors of X are found among the K observations of X, and X is classified into the category that appears most frequently among the most recent K training instances. The KNN algorithm gradually enlarges the area surrounding the test sample X until it includes K training instances, and then applies the decision rule. For instance, in Figure 4.8, when K=6, the decision rule assigns the test sample X to the black category.

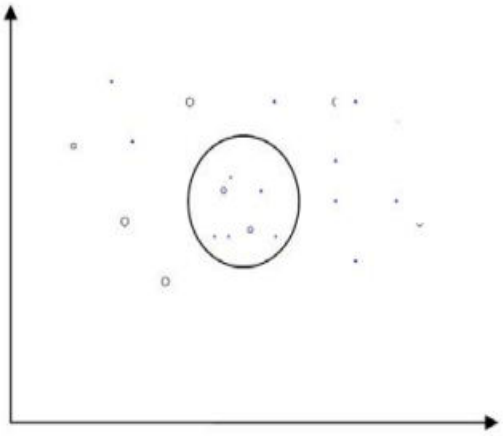

Figure 4.7 KNN

KNN is a slow learning technique based on the eyeball, the neighbourhood classification that saves all the training samples and is aware that the new samples must also be classified to create the classification process. Decision numbers and backpropagation algorithms, on the other hand, must first construct a general model prior to embracing samples for classification. Since all calculations are put off until then, lazy learning is slower in categorization but faster in training than eager learning. The comparison between KNN and the decision tree is categorized as follows:

Table 4.1 Comparison between KNN and DT

Decision Tree (DT)	**KNN**
Eager classification	Lazy classification
Supervised Learning process	Unsupervised learning process
The records are classified using some rules	The records are classified by deciding the neighbors
Both numerical and categorical attributes are accepted	Only numeral attributes are accepted.
Speed is slow for large databases.	Speed is faster for all types of data

The white box in which classified text is in readable form	Black box in which classified text is not in readable form
Deterministic	Non-Deterministic
Effective results on a small dataset	Effective results on a large dataset
Depends upon other algorithms like the Hunt algorithm for best results.	KNN has its own algorithm to perform different tasks

- **Artificial Neural Network**

The artificial neural network gets its name because it refers to the work of neurons in the brain. A neural network is a system composed of computing units-artificial neurons function similar to neurons in a biological brain [**Shanmuganathan**, **2016**]. Like biology, artificial neurons receive and process information, and then transmit it further. By interacting with each other, neurons can solve complex problems [**Sharma *et al.***, **2020**], including:

- Object class definition,
- Identify dependencies and aggregate data,
- Divide the received data into several groups according to the specified characteristics,
- Forecast etc.

Neurons are special biological cells that process information as shown in Figure 4.8. It consists of cell bodies or somatic cells and two types of external dendritic branches: axons and dendrites. The cell body includes a nucleus and plasma. The nucleus contains information about genetic characteristics, while plasma contains the molecular information to produce materials needed for neurons. A neuron receives signals in the form of pulses from other neurons through the dendrites [receivers], and transmits signals generated by the cell body along the axons [transmitters], which pass into strands at the ends known as synapses.

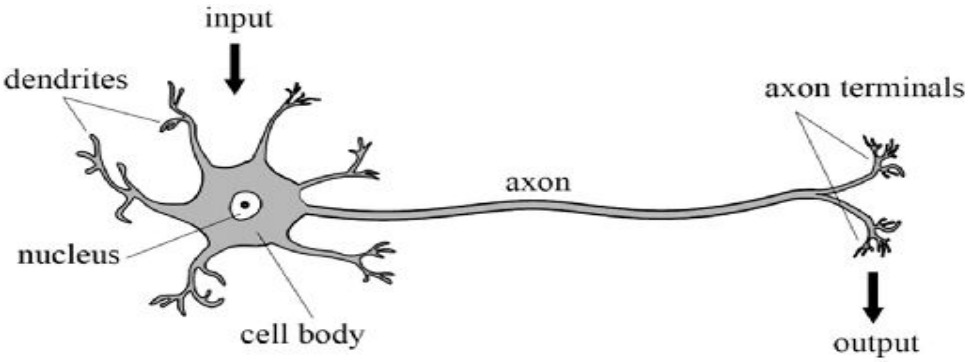

Figure 4.8 Biological Neuron

- **Working of ANN**

Learning ability is a fundamental characteristic of the brain, and it is also a key component of artificial neural networks. In the realm of neural networks, learning involves creating an effective network architecture and adjusting link weights to perform specific tasks. Typically, this involves adjusting the link weights of the network based on the available training samples. As the weights are fine-tuned, the performance of the network improves.

Artificial Neural Networks (ANNs) are biologically inspired computer programs designed to process information in the same manner as the human brain. ANN detects patterns and data relationships and learns through experience rather than collecting knowledge from programming. ANN consists of hundreds of units. These artificial neurons are also called processing elements, which are related to weights and create neural structures and arrange them in layers as shown in Figure 4.9.

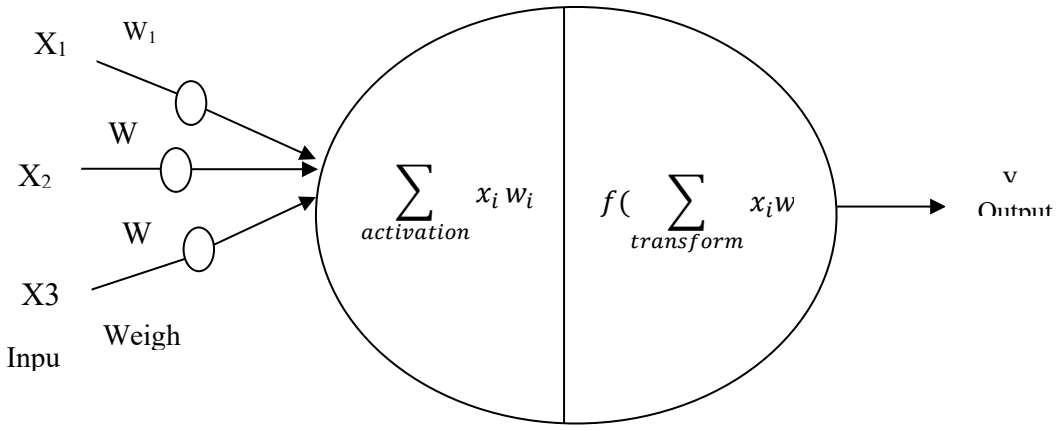

Figure 4.9 Structure of ANN

Summary

The chapter illustrates the proposed work model in detail. The proposed work has been performed algorithmically to both the kind of the data namely the data with the available GT value and the data where there is no GT value. In case no GT is available, the proposed work has divided data into three categories namely Positive, Negative, and Neutral to label the created clusters, a statistical approach inspired by Machine Learning has been applied to the data. To extract the relevant features from the set, the ABC algorithm has been applied with a novelization in the grouping and evaluation behaviour. To train the system with the supplied GT value and the selected feature set, different training algorithms viz. Neural Networks, Naïve Bayes Classifier, etc has been applied. Out of all the applied classifiers, the result of Neural Networks has been identified as the best possible solution for utilized data. A rule mining engine based on mean and variance is applied for the final recognition of data elements.

CHAPTER 5:
PROPOSED METHODOLOGY FOR RULE MINING

5.1 Proposed work

5.2 Feature Selection

5.3 Rule Mining

Summary

5.1 Proposed work

Intelligent software development that includes machine learning and big data has become a critical part of larger businesses. Companies are using soft computing algorithms to increase their efficiency, and the goal is to look for approaches and rules with the most optimization support. Threshold values for determining optimized solutions are entirely optional, but all options are carefully considered.

The proposed framework is represented in Algorithm 1, in which first the dataset is loaded, and then preprocessing is performed. After performing the preprocessing, features are selected by using the proposed method Grouped ABC(G-ABC) then data is divided into 70% to 30% ratio to validate the procedure by using four classifiers K-NN, NB, SVM, and NN. After getting selected features, association rule mining is performed using no minimum support and no minimum confidence, and again, results are validated by using four classifiers K-NN, NB, SVM, and NN.

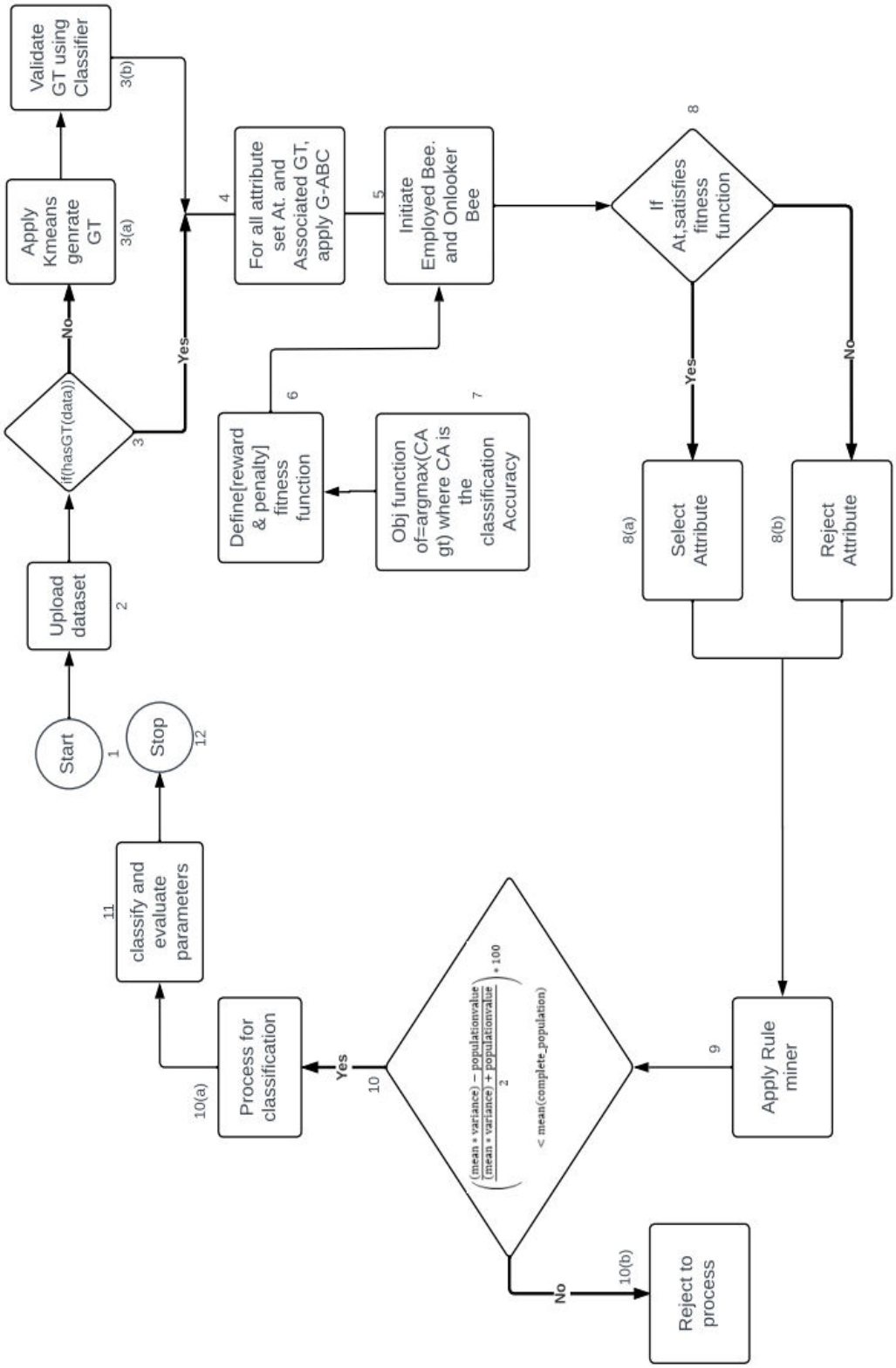

Figure 5.1 The process of the proposed algorithm (R-Miner Using soft computing)

As shown in Figure 5.1, the proposed work aims to perform prediction for both labelled and unlabelled data. In the case of the labelled data, there is no need for ground truth(GT) generation whereas, in case of unlabelled data, the GT is generated via two subsequent processes. In the first Process, the data is divided into three groups considering the variance of the data and label them as Good, Moderate, and Bad based on their co-relation present in the data. The proposed work applies an improved Artificial Bee Colony(ABC) with improved fitness function. Once the data attribute set is selected via the proposed Grouped ABC, an association rule mining is applied to check whether the data is suitable for classification or not. To do so, a mean and variance method is applied. If the variance in the selected data is greater than the complete population mean, then it can be processed for classification. Furthermore, a neural-based deep learning method is applied.

Algorithm 1 Proposed Algorithm: R-Miner Using Soft Computing

Require: ds as data set gt as Ground Truth

Ads = ds. sort(gt)Arrange data as per gt

IF $isempty(gt)$

k =

2: 5 initate Cluster size from 2 and ending upto5, represents the possibility of the clusters

$[kindex, kcent] = kmeans\ (A - ds, k) Divide\ the\ data\ into\ k\ number\ of\ clusters$

$Initiate\ Nb = Naive\ Bayes(Ads, kindex)$

$Initate\ Naive\ Bayes\ Classifier\ for\ the\ validation\ of\ the\ k - clusters$

$Classification\ A = Classify.NaiveBayes(Nb, Ads, randomsample())$

$Ma = Find\ Max(ClassificationA);$

$Find\ maximum\ classification\ acuracy\ from\ all\ clusters$

$gt = kindex.Ma;$

end IF

Require: $AST\ as\ Attribute\ Set, K\ as\ total\ number\ of\ GT\ Classes$

$k = 1;$

$initialize\ a\ state\ variable\ where$

$k \parallel = \{1, \dots, k\}$

$While\ k \leq K$

$Eb = Find(Ast.index -$

$of(k))Extracting\ the\ feature\ vectors\ of\ the\ specified\ class$

$lf\ initialize\ levy\ flight\ variable\ where\ lf = \{1, \dots, l\}\ AND\ L = 10$

$while\ lf \leq L$

$Bp = Generate\ Population\ (AST.Random())$

$Obl = means(Bp);$

$if\ Bp\ satifies\ Obl$

$reward++$

$else$

$plenty++$

$endif$

$if\ reward \geq plenty$

$accept\ attribute$

$else$

$reject\ attribute$

$End\ While$

Initiate Rule Mining Engine

$f = Choose\ Mining\ Engine$

$for\ m = 1: gt$

$m = 1: gt$

$rn = Generate\ Random - pop;$

$generate\ a\ random\ population$

$Popmean = Calculate\ Mean(Rn);$

$PopVar = Calculate\ Variance(Rn);$

$MeanA = Calculate\ Mean(gt.record)$

Variance=Calculate Variance (gt.record);

$f = Choose\ Mining\ Engine$

$\boldsymbol{for}\ m = 1: gt$

$rn = Generate\ random - pop;\ generate\ a\ random\ population$

$PopMean = Calculate\ Mean\{Rn);$

$PopVar = Calculate\ Variance(Rn);$

$MeanA = Calculate\ Mean(gt.record)$

Variance=Calculate Variance (gt.record);

$\boldsymbol{if}\ \left(\frac{(m.Mean * m.variance) - (PopMean * PopVar)}{(PopMean * PopVar) + (Mean * Variance)}\right) * 100$

$< mean(complete\_population)$

$Accept\ for\ processing,$

Else

$Reject\ for\ processing$

$\boldsymbol{END\ FOR}$

$a.NearestNeighbor$

$b.SVM$

$c.Naive\ Bayes$

$d.Neural\ Propagation$

$Engine\ [xtrain, xtext.ytrain, ytest] = SplitTrainTest(Ads, 70{:}30)$

$xtrain{:}Training\ Dataset$

$xtest{:}Test\ Dataset$

$ytrain{:}Test\ Label$

$\boldsymbol{if}\ f == a$

$Nc = 3; Neighbor\ Count$

$tR = InitiateTraining(xtrain, ytrain, Nc);$

$Classifiedresult = Simulate(tR, xtest);$

$Classificationscore(ClassificationResult, ytest)$

$\boldsymbol{Elseif}\ f == b$

$KernalFunction = '\ linear', 'polynomial', 'rbf'$

$tR = InitiateTraining(xtrain, ytrain, KernalFunction);$

$ClassifiedResult = Simulate(tR, xtest);$

$Classificationscore(ClassificationResult, ytest)$

$\boldsymbol{Elseif}\ f == c$

$BCI = '\ Gaussian';$

$tR = InitiateTraining(xtrain, ytrain, BCI);$

$ClassifiedResult = Simulate(tR, xtest);$

$Classificationscore(ClassificationResult, ytest)$

Else

$Neural\ Layer = 5{:}\ 20$

$InitiateTraining(xtrain, ytrain, BCI);$

$ClassifiedResult = Simulate(tR, xtest);$

$Classificationscore(ClassificationResult, ytest)$

This algorithm is divided into 2 parts. The first step indicates preprocessing. Words like "the" and "and" are commonly used in documents and make data analysis more easily understandable, but these words are not required for analysis. These words can be ignored as they are called stop words. Some other words are used frequently in the English language such as proper nouns, numbers, conjunctions, and prepositions. So, it may be a good idea to remove such common terms when converting data into text format. It is also possible to use stemmer which removes any suffixes like –ed, –ing, etc before processing them as a sentence or token based on their frequency of occurrence (English corpus). After pre-processing the textual data it was observed that the number of features to be used for the next phase remains the same. So to mine optimized features further feature selection is implemented. For this, in the proposed model, we used G-ABC (proposed model). This algorithm combines five randomly collected active bees to reduce the time of execution. Furthermore, this algorithm reduces the selected features for rule generation in step 2. Step 2 is used for rule mining where no threshold value is allocated. The fitness function includes the calculation of the mean and variance for each element in the population. For both steps, three classes of ground truth values are used: Positive, Negative, and Neutral. For each element, mean, variance, and mean*variance are calculated. Additionally, these values are advancing for calculating the fitness value for each element. If the condition is true, then the value is added to the final population. Otherwise, the element is rejected.

- **Dataset**

  - Twitter Kaggle:

Twitter Dataset consists of tweets of community and their emotions [**Jamal *et al.*, 2019**], [**K. Khan & Ramsahai, 2020**], [**Mohana *et al.*, 2021**]. There are three different columns used for twitter

such as twitter id, sentiments, and last column for messages. Out of these columns, twitter id is a unique identifier which is used for each tweet, sentiment can be positive (1) or negative (0). Tweet in the Kaggle set is represented in columns "." Further, the dataset is made up of words, emotions, references and URLs. In the training datasets, there are more than 700000 tweets and 200000 tweets for test datasets. URLS are provided for frequent access to the data and there are also some Hashtag available used for unstructured sentences accompanied by # symbol. For instance, the phrase for suitable hash tag is #(S+).

- BASEBALL

The dataset contains two elementary information viz. the data and the GT. As illustrated earlier, the data contains a lot of ambiguous information and hence a pre-processing is required to remove ambiguous information from the data and is termed as pre-processing [**Jang *et al.***, **2014**].

5.2 Pre-processing

The main steps of pre-processing are Disintegration of records and Removing "Stop Words" . Common English terms such as "the", "of", "to" and other miscellaneous terms are not necessary for analysis; these words are often known as Stop-Words. Without producing any useful results, these words obnoxiously take up extra time to execute. Therefore, eliminating these words is necessary to obtain more optimal results. Furthermore, stemming the words involves accurately portraying word regularity. Stemming words also refers to the process of simplifying words by dropping consonants like "ed" and "ty." This process indicates that when a sentence is tokenized into terms for each individual, each tuple value and word can function as tokens. Before the tokenization Porter Stemmer algorithm is used to remove the stemming particles from the words., such as hopping reduces to hop, computing reduces to computer, and so on.

Pre-processing is one of the major architectural steps in a simulation design against text and other multimedia data. In the case of the proposed work, the pre-processing architecture involves two pre-processing is required to remove ambiguous information from the data steps namely stop word removal and tokenization. Stop word refers to the words that do not contribute to any decision-making process. For example, "The movie has sufficient frames to declare it a good movie" and "The movie has sufficient frames to declare it a bad movie" belong to two different contexts but if a system has to understand both categories, the first 8 words are not contributing to decision as

they are common in both the context. In such a situation, these words are called stop words and there is a full list of stop words for most possible languages on this planet Earth that contribute to any technical advancement. Stop words may include all the punctuations, nouns, verb, etc.

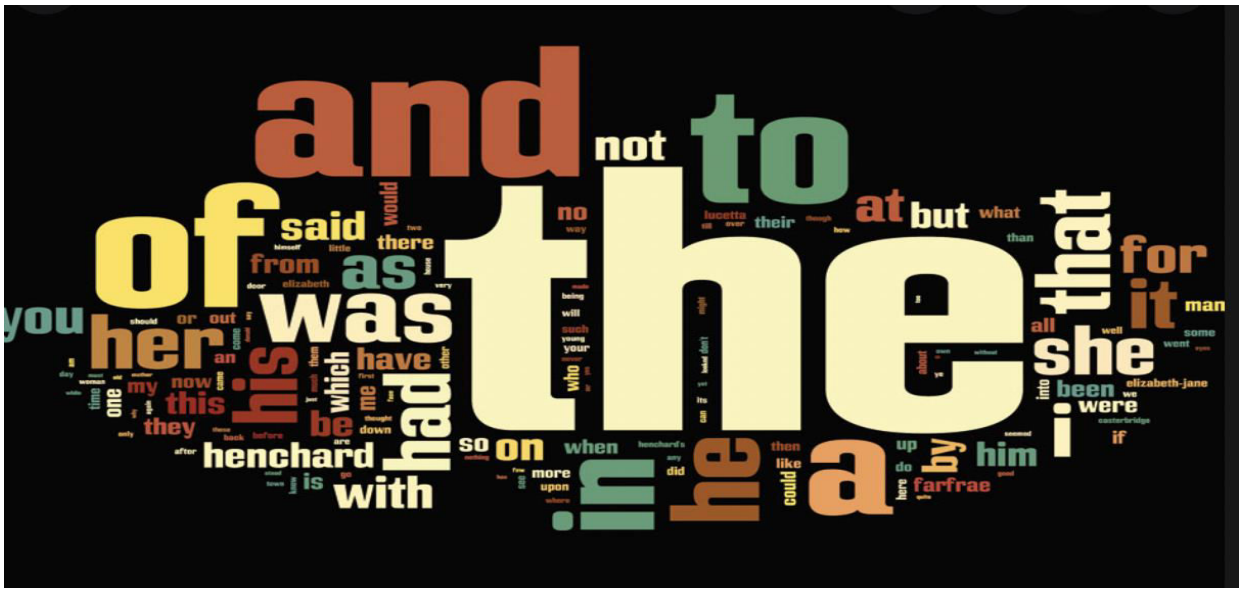

Figure 5.2 Stop word sample

The proposed work uses Porter Stemming Algorithm (PSA) for the filtration of stop words. The proposed implementation architecture is designed and developed under Python development pattern. . First of all, the data is separated against its specified emotion, and the three emotions considered here are used to represent positive, negative, and neural sentiments. For example, a sample space of 100 files is demonstrated in Figure 5.3.

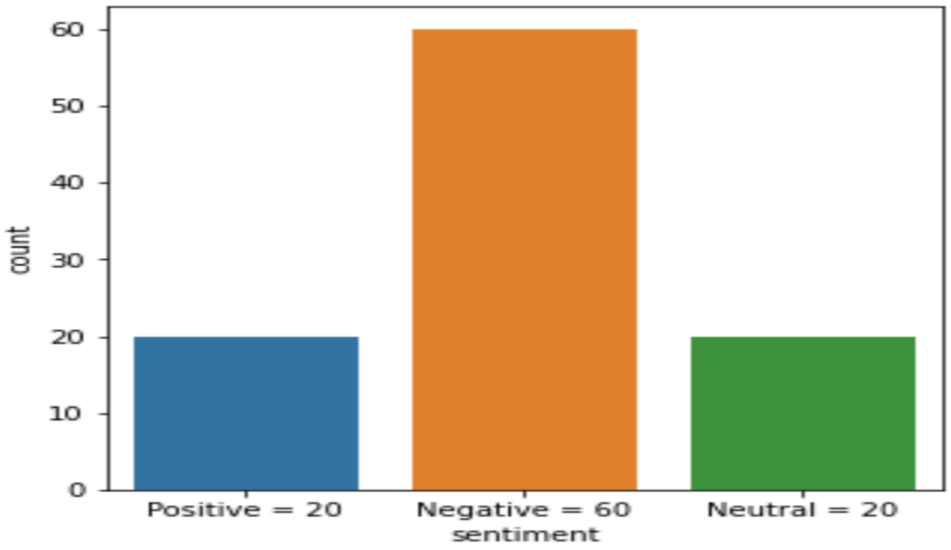

Figure 5.3 Data Separation against its ground truth value

As shown in Figure 5.3, the data is separated into 3 ground truth classes namely positive, negative, and neutral. The dataset of baseball contains 1300 samples whereas in case of twitter dataset, there are 4343434 no of samples against different ground truth values.

5.3 Feature Selection

The basic principle of ARM is dependent upon the maximum confidence generated within a given class interval. The highest confidence value will represent the closest co-relation among the data attributes and can be recommended on top of any other made recommendation. Keeping the architecture in mind, the proposed ABC algorithm is also based on central co-relation among the data attributes and selects the best attribute set that represents the best co-relation. The objective is to increase the overall classification accuracy to maximize the association architecture among the data attributes and its ground truth value. The proposed ABC algorithm can be illustrated using the following flow diagram.

The proposed ABC algorithm is divided into 12 consecutive steps as shown in Algorithm Improved ABC. The proposed algorithm utilizes the principle of co-relation and considers the feature vector of each class, denoted by K. For example, for the baseball dataset, and for the Twitter dataset, the value for K is 3 based on three emotions namely positive, negative, and neutral. Each feature is considered as one employed bee from the respective class. The proposed algorithm demonstrates a levy distribution-based behaviour in which the employed bee is paired with 5 other bees to form an employed bee group. The global food of the group will be considered as the harmonic mean of every attribute set in the entire group data denoted as Attribute Set (ATS) in Figure 3.6. The target is to maximize the overall classification accuracy of the class to show a strong association among the data attributes respective to their ground truth value. In other words, it means that the main objective is to achieve maximum class accuracy for each of the three classes under study each used to represent a specific type of sentiment, namely, positive, negative, and neutral sentiments. In such a scenario, a global best and a local best is passed to the ABC fitness function. If the local best is not far from the global best, it is considered to be the food that can be served or selected.

The proposed work has been implemented on Python simulation architecture. To support the further programming architecture that also contains Keras libraries, the development platform was switched to Google Collab. The proposed ABC algorithm architecture is based on the grouping behaviour of the bees in the hive and hence the proposed algorithm has been named grouped ABC.

The algorithm architecture is as follows:

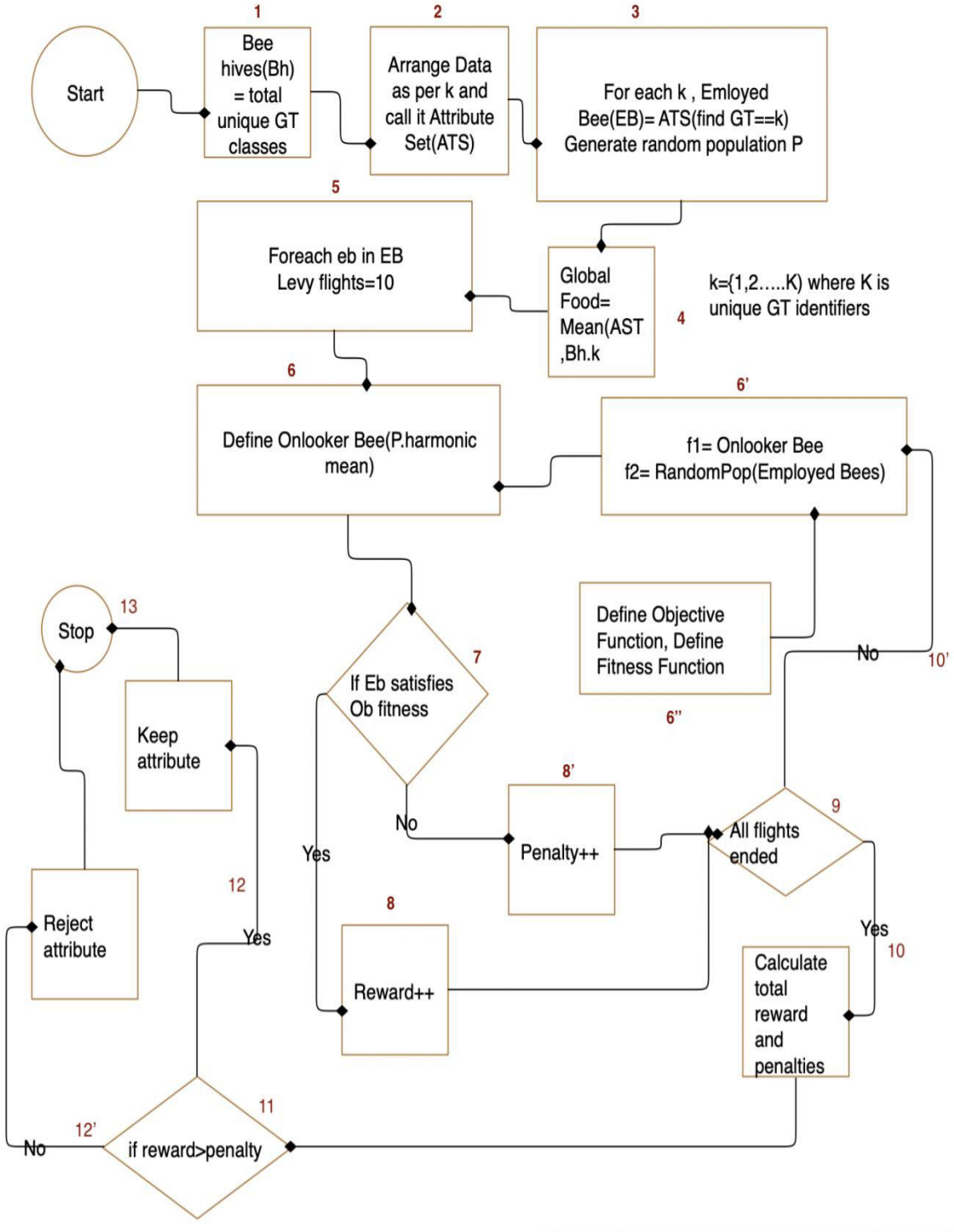

Figure 5.4 The process of proposed G-ABC algorithm

- **Implementation design of Grouped-ABC**

The proposed ABC algorithm is made of up two bee components namely the employed bee and the onlooker bee whereas the proposed algorithm architecture is not comprised on the scout bee. As clear from the ABC definition architecture, a scout bee is a bee that takes rest once as an employed bee, it has worked so hard that it cannot fly anymore to collect good quality juice which is later to be formed as honey. In the case of the proposed algorithm architecture, every bee is getting evaluated for levy flights and not for a lifetime, this results in ignorance of the scout bee in the case of the proposed solution. As shown in Figure 5.5 the formation of the employed and the onlooker bee for every phase is done on the Feature data that is passed for every ground truth value.

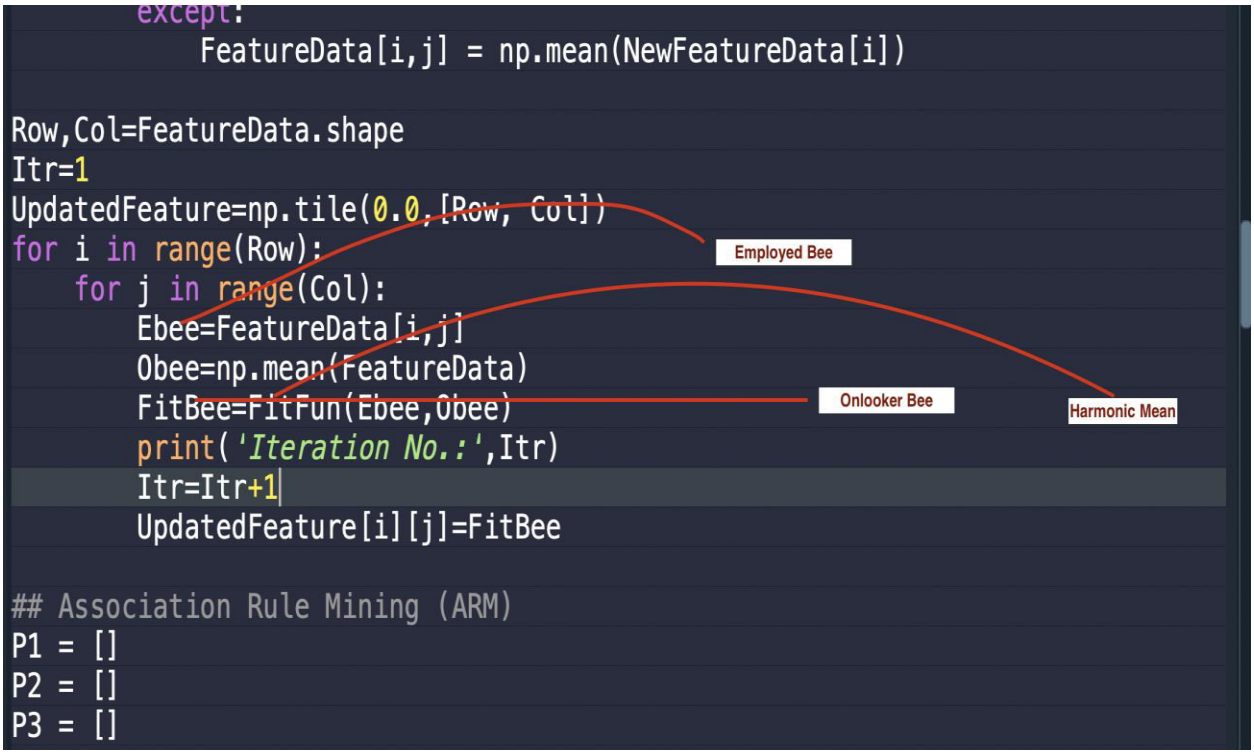

Figure 5.5 ABC Architecture

ABC is formed with the feature vector where the employed bee and the onlooker bee are formed from the feature data itself. Both the employed bee and the onlooker bee are passed to a fitness function. In the case of the proposed implementation architecture, it has been named and termed as FitFun.

```
    Target[1,0] = 2

#%% Apply ABC for Selection
## Fitness function of ABC
def FitFun(Ebee,Obee):
    if Ebee > Obee:
        FitBee=Ebee
    else:
        FitBee=Obee
    return FitBee
```

Figure 5.6 Fitness function

As for example, the sampled dataset is rectified or optimized by ABC and the results are provided in table 5.1 as follows. The table provides the information of the size and the list representing the feature vectors for each index.

Table 5.1 Optimized feature set

Index	Type	Size	List
0	List	9	[[2, 195, 79, 196, 197, 1, 198, 199, 200],
1	List	8	[16, 215, 94, 216, 217, 218, 219, 1],
2	List	12	[220, 221, 2, 222, 223, 45, 95, 224, 225, 46, 226, 1],
3	List	13	[2, 248, 249, 39, 23, 250, 251, 252, 253, 254, 255, 53, 1],
4	List	9	[99, 100, 256, 257, 27, 258, 259, 18, 1],
5	List	14	[2, 276, 108, 109, 277, 278, 279, 280, 281, 282, 110, 283, 284, 1],
6	List	9	[2, 111, 28, 285, 15, 286, 29, 112, 1],
7	List	8	[2, 291, 1, 292, 293, 3, 4, 294],
8	List	14	[1, 301, 59, 46, 123, 60, 124, 32, 5, 28, 23, 125, 108, 45],
9	List	12	[322, 16, 133, 323, 324, 325, 134, 1, 62, 326, 327, 328],
10	List	14	[2, 141, 345, 346, 347, 348, 349, 350, 27, 351, 352, 17, 142, 353],

11	List	10	[2, 374, 375, 376, 377, 378, 147, 57, 1, 379],
12	List	10	[2, 387, 388, 389, 390, 1, 3, 4, 391, 392],
13	List	10	[2, 396, 397, 2, 398, 1, 3, 4, 399, 400],
14	List	14	[81, 1, 82, 10, 83, 84, 85, 86, 87, 8, 5, 31, 24, 425],
15	List	14	[441, 57, 29, 442, 443, 17, 58, 444, 10, 163, 445, 446, 447, 1],
16	List	9	[2, 478, 479, 1, 480, 481, 3, 4, 482],
17	List	12	[2, 489, 1, 490, 491, 36, 492, 493, 5, 71, 494, 3],
18	List	14	[121, 517, 518, 519, 520, 73, 521, 522, 523, 524, 37, 1, 95, 525],
19	List	15	[2, 526, 74, 5, 527, 528, 529, 18, 73, 530, 531, 20, 112, 532, 533],
20	List	16	[2, 542, 543, 180, 18, 165, 544, 545, 546, 547, 166, 107, 98, 7, 6, 1],
21	List	17	[2,559,560,175,1,561,562,563,164,566,567,568,3,104]
22	List	8	[2, 577, 59, 578, 579, 130, 580, 1],
23	List	15	[589, 590, 29, 35, 591, 109, 592, 593, 594, 595, 1, 148, 149, 7, 6],
24	List	13	[2, 607, 608, 146, 609, 136, 610, 12, 77, 78, 611, 1, 612],
25	List	15	[2, 619, 99, 9, 183, 35, 620, 1, 127, 621, 622, 623, 19, 624, 3],
26	List	13	[631, 73, 167, 1, 7, 6, 3, 4, 632, 3, 4, 633, 634],
27	List	12	[2, 641, 1, 8, 642, 643, 644, 645, 186, 12, 180, 38],
28	List	18	[2, 653, 654, 655, 3, 4, 656, 657, 658, 7, 6, 1, 659, 660, 661, 34, 144, 8],
29	List	15	[2, 169, 668, 669, 62, 670, 671, 672, 673, 10, 674, 1, 3, 4, 675]]

5.4 Rule Mining

Rule mining is a method where we search for patterns in the data. In the case of medical data, it could be used to predict something like age at the time of diagnosis and other personal attributes that can have a large impact on human life. The mean-variance optimization algorithm is nature inspired algorithm where two parameters are optimised at the cost of other parameters and the output will be a ruleset that is not just based on some assumptions but also based on real data occurrences of the objects in our dataset. This can come in very handy when we cannot identify a pattern immediately in exploratory analysis but we still want to know if there is any relationship

between two objects in our dataset as that can help us with future predictions and/or classification such as patient mortality prediction etc.

After selecting the features rule mining is implemented with a mean, variance optimization method. The proposed framework performs rule mining without a threshold for confidence and support requirements. For evaluating the rule's accuracy, a fitness function is used where the mean and variance of each element or population are calculated. Initially, data after feature selection is divided into three populations or classes. "Mean * variance optimization" is used for each population's element. Further, mean, variance is calculated for each element and if the condition is true then the value is accepted otherwise the value is rejected. After calculating the final rules again four classifiers KNN, SVM, NB, and NN are used. 70% and 30% of data are used for training and testing respectively. Finally, the complete framework is evaluated based on performance measures.

The proposed work uses mean and variance to perform the rule mining engine. The mean is referred to as the arithmetic mean of the supplied elements and the variance is evaluated based on the mean itself. In order to be intact, the mean and variance is aimed to be on the lower side to be precise on the classification score. The mean and variance can be mathematically defined as follows.

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{n} \boldsymbol{A_i}}{\boldsymbol{n}} \quad \textbf{[5.1]}$$

$$\boldsymbol{Variance} = \frac{\sum_{i=1}^{n} \boldsymbol{A_i - \mu}}{\boldsymbol{n}} \quad \textbf{[5.2]}$$

Fitness Function:

The performance of each population element is evaluated by following formula:

$$\left(\frac{(mean*variance)-population\_value}{((mean*variance)+population\_value)/2} * 100 < mean(\ complete\_population)\right) \quad \textbf{[5.3]}$$

Where mean is calculating average of the given set of values and variance is it gauges how widely apart a group of numbers are from one another.

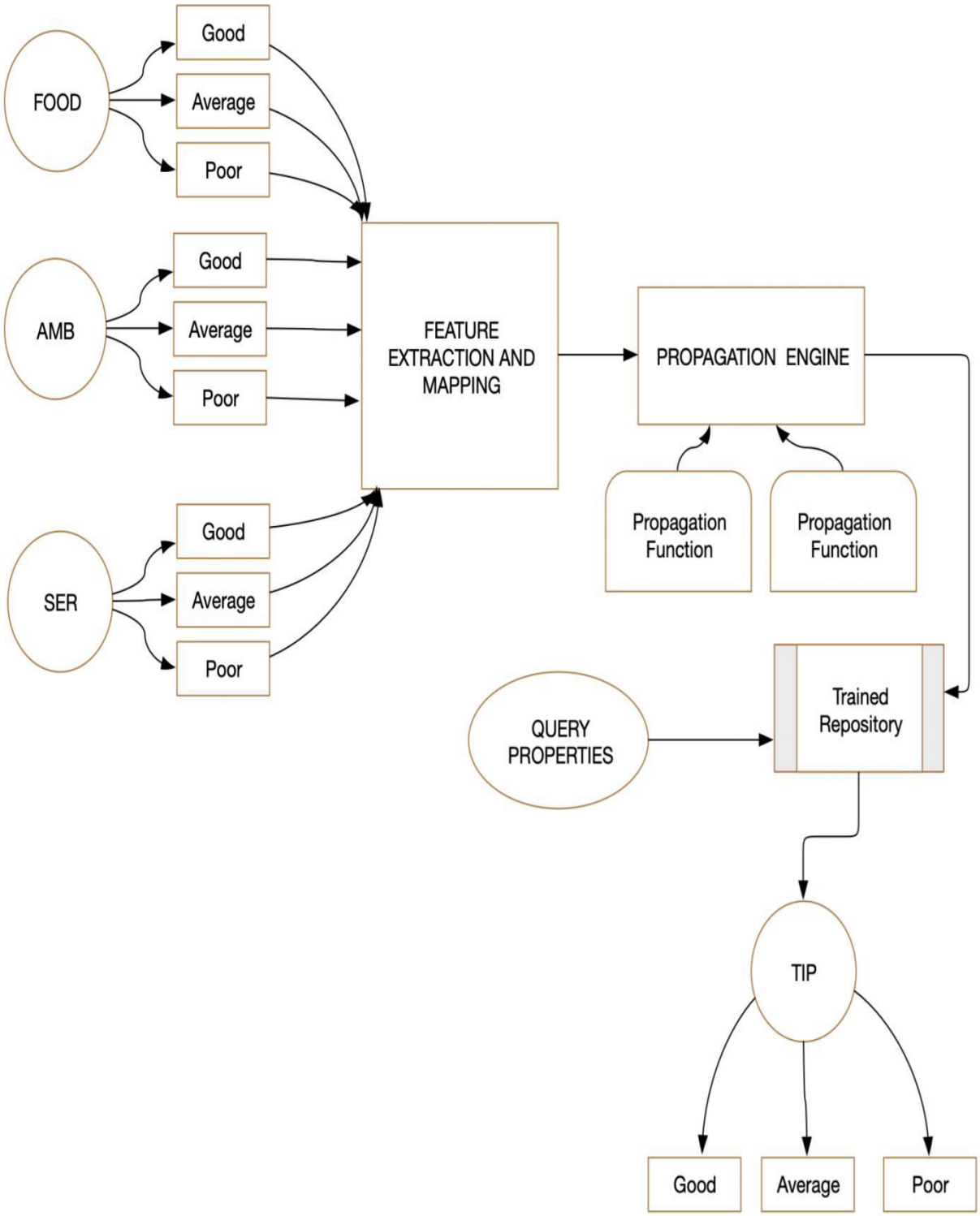

Figure 5.7 Propagation-based rule mining architecture

Propagation-based learning methods are useful when it comes to latency reduction. A propagation-based rule mining architecture contains four essential components as follows.

a. Data
b. Feature extraction method or extracted features
c. Mechanism of propagation engine
d. Validation of the outcome

As the rules in the case of propagation based mining architecture are incorporated through propagation functions, the prediction time is quite low as compared to straight rule-based architecture as shown in Figure 4.1. The propagation engine converts the input variable's membership function into a property vector using a feature extraction mechanism or algorithm. These features are propagated through a propagation engine rather than getting propagated through a rule engine. The propagation engine uses a propagation function to circulate the data against its Ground Truth (GT) value. The propagation engine also has a stopping criteria that decides when the propagation engine has to stop the training. The user data is classified against the GT which in the case of the illustrated example is Tip-Good, Tip-Average, and Tip-Bad.

The neural engine uses the weighted method to predict the emotion of the provided input. The rules for the detection can be illustrated using the following table. The rules have been formed in such a way that Ground_Truth(2) is encoded with emotion value 2, Ground_Truth is encoded as 1 and Ground_Truth is encoded as 3.

Table 5.2 Represents Rule Set

Rule	**Antecedent**	**Consequent**
Rule 1	high_positive_valence, high_arousal	Ground_Truth(2)
Rule 2	low_positive_valence, low_arousal	Ground_Truth (1)
Rule 3	moderate_positive_valence, moderate_arousal	Ground_Truth (3)
Rule 4	high_positive_valence, low_arousal	Ground_Truth(2)

Rule 5	low_positive_valence, high_arousal	Ground_Truth (3)
Rule 6	high_positive_valence, moderate_arousal	Ground_Truth(2)
Rule 7	low_positive_valence, moderate_arousal	Ground_Truth (3)
Rule 8	moderate_positive_valence, low_arousal	Ground_Truth (1)
Rule 9	moderate_positive_valence, high_arousal	Ground_Truth (3)
Rule 10	moderate_positive_valence, low_arousal	Ground_Truth(2)

In the above table, the antecedent represents the combination of input features related to sentiment scores (e.g., positive sentiment score, negative sentiment score) and arousal scores. The consequent represents the predicted sentiment class as Ground_Truth(1), Ground_Truth(2), and Ground_Truth(3).

Please note that in sentiment analysis with Neural Networks, weights are learned during the training process, and associations between features and sentiment classes are determined based on the network's learned parameters. The rulesets can be encoded as follows.

1. Rule 1 1, 0 2
2. Rule 2 0, 1 1
3. Rule 3 2, 2 3
4. Rule 4 1, 2 2
5. Rule 5 2, 1 1
6. Rule 6 1, 3 2
7. Rule 7 0, 4 1
8. Rule 8 2, 3 3
9. Rule 9 0, 3 2
10. Rule 10 1, 4 2

Rule 1:

Antecedent: high_positive_sentiment_score, low_negative_sentiment_score

Consequent: Ground_Truth(2)

Explanation: Rule 1 states that an input with a high positive sentiment score and a low negative sentiment score is classified as Ground_Truth(2). This rule assumes that a high positive sentiment score indicates a positive sentiment and a low negative sentiment score implies a lack of negative sentiment. Therefore, based on these criteria, the sentiment is classified as Ground_Truth(2).

Rule 2:

Antecedent: low_positive_sentiment_score, high_negative_sentiment_score

Consequent: Ground_Truth (1)

Explanation: Rule 2 suggests that an input with a low positive sentiment score and a high negative sentiment score is classified as Ground_Truth (1). This rule assumes that a low positive sentiment score indicates a lack of positive sentiment, and a high negative sentiment score implies a strong presence of negative sentiment. Therefore, based on these criteria, the sentiment is classified as Ground_Truth.

Rule 3:

Antecedent: moderate_positive_sentiment_score, moderate_negative_sentiment_score

Consequent: Ground_Truth (3)

Explanation: Rule 3 states that if an input has both a moderate positive sentiment score and a moderate negative sentiment score, it is classified as Ground_Truth (3). This rule assumes that moderate values for both positive and negative sentiment scores imply a balance between positive and negative sentiments, resulting in a Ground_Truth sentiment.

Rule 4:

Antecedent: high_positive_sentiment_score, moderate_negative_sentiment_score

Consequent: Ground_Truth(2)

Explanation: Rule 4 suggests that an input with a high positive sentiment score and a moderate negative sentiment score is classified as Ground_Truth(2). This rule assumes that a high positive sentiment score indicates a positive sentiment and a moderate negative sentiment score implies a relatively low presence of negative sentiment. Therefore, based on these criteria, the sentiment is classified as Ground_Truth(2).

Rule 5:

Antecedent: moderate_positive_sentiment_score, high_negative_sentiment_score

Consequent: Ground_Truth (1)

Explanation: Rule 5 states that an input with a moderate positive sentiment score and a high negative sentiment score is classified as Ground_Truth (1). This rule assumes that a moderate positive sentiment score indicates some positive sentiment and a high negative sentiment score implies a strong presence of negative sentiment. Therefore, based on these criteria, the sentiment is classified as Ground_Truth.

Rule 6:

Antecedent: high_positive_sentiment_score, high_arousal_score

Consequent: Ground_Truth(2)

Explanation: Rule 6 suggests that if an input has both a high positive sentiment score and a high arousal score, it is classified as Ground_Truth(2). This rule assumes that both high positive sentiment and high arousal indicate a positive and energetic sentiment, aligning with the classification of happiness.

Rule 7:

Antecedent: high_negative_sentiment_score, low_ arousal_score

Consequent: Ground_Truth (1)

Explanation: Rule 7 states that an input with a high negative sentiment score and a low arousal score is classified as Ground_Truth (1). This rule assumes that a high negative sentiment score indicates a strong presence of negative sentiment, and a low arousal score suggests a lack of energy or excitement. Therefore, based on these criteria, the sentiment is classified as Ground_Truth.

Rule 8:

Antecedent: moderate_positive_sentiment_score, moderate_arousal_score

Consequent: Ground_Truth (3)

Explanation: Rule 8 suggests that if an input has both a moderate positive sentiment score and moderate arousal score, it is classified as Ground_Truth (3). This rule assumes that moderate values for both positive sentiment and arousal imply a balanced and Ground_Truth sentiment.

Rule 9:

Antecedent: low_positive_sentiment_score, high_arousal_score

Consequent: Ground_Truth(2)

Explanation: Rule 9 states that an input with a low positive sentiment score and a high arousal score is classified as Ground_Truth(2). This rule assumes that a low positive sentiment score indicates a lack of positive sentiment, but a high arousal score suggests a high level of energy or excitement. Therefore, based on these criteria, the sentiment is classified as Ground_Truth(2).

Rule 10:

Antecedent: high_positive_sentiment_score, low_arousal_score

Consequent: Ground_Truth(2)

Explanation: Rule 10 suggests that an input with a high positive sentiment score and a low arousal score is classified as Ground_Truth(2). This rule assumes that a high positive sentiment score indicates a positive sentiment and a low arousal score suggests a lack of energy or excitement. Therefore, based on these criteria, the sentiment is classified as Ground_Truth(2).

These rules are designed to capture relationships between different sentiment features and sentiment classes. Each rule specifies the conditions (antecedent) under which a particular sentiment class (consequent) is assigned. However, it's important to note that these rules are hypothetical and may not accurately reflect the complexity of sentiment analysis. In practice, sentiment analysis using neural networks involves training the model on labelled data, learning the patterns and relationships between features and sentiment classes, and making predictions based on the learned model.

Summary

The chapter illustrates the proposed work model in detail. The proposed work has been performed in an algorithmic way to both the kind of the data namely the data with the available GT value and the data where there is no GT value. In case no GT is available, the proposed work has divided data into three categories namely Positive, Negative, and Neutral to label the created clusters, a statistical approach inspired by Machine Learning has been applied to the data. To extract the relevant features from the set, the ABC algorithm has been applied with a novelization in the grouping and evaluation behaviour. To train the system with the supplied GT value and the selected feature set, different training algorithms viz. Neural Networks, Naïve Bayes Classifier, etc have been applied. Out of all the applied classifiers, the result of Neural Networks has been identified as the best possible solution for utilized data. A rule mining engine based on mean and variance is applied for the final recognition of data element.

CHAPTER 6:
RESULTS AND DISCUSSION

6.1 Introduction

6.2 Evaluation for Feature Selection Approaches

6.3 Evaluation of Classification Approaches

6.4 Evaluation using Multiple Simulations

6.5 Evaluation Using More Data Sample

6.1 Introduction

In the last few decades, data mining has played a vital role in decision-making and is considered, an essential tool to perform different operations. Data mining is essential to discovering unknown patterns from a large database **[Aggarwal *et al.*, 1993], [Cios *et al.*, 1998]**. Contains various functionalities, algorithms, models, and techniques used to discover and extract the relevant patterns from the large database repository. Association rule mining is a well-known technique of data mining. This technique is used to extract the correlation between the data points, frequent patterns, and associations between the structures **[Rehman *et al.*, 2021], [Ben *et al.*, 2022]**. Association rule mining will be used for the analysis of association among datasets for performance and correctness. The motive of this paper is to describe this new method by associating it with the original problem.

In machine learning and artificial intelligence, association rule mining is one of the techniques and ways of using big data for information extraction **[Batool *et al.*, 2023].** As the process is an unsupervised learner, it depends on the similarity or correlations of the dataset members that are used for training. Association rule mining, also known as soft clustering or dependency mapping, helps represent the data visualizing map generated from different types of representations such as diagrams, etc**[Batool *et al.*, 2022], [Boulila *et al.*,2010]**.

Several mining algorithms are used to mine rules. The Apriori algorithm uses a nested data structure to narrow the search space and achieve better results than other algorithms. FP tree is also another efficient data structure for finding rules from large datasets **[Aggarwal *et al.,* 1993]**. In this approach, we need to build one single node for each customer and its relation to their budget limit. The file created in this way can be fed into the FP tree algorithm which will compute the branch of interest and return the value of that branch. This value should fall within the restricted range determined by the user's budget limits. Let's say we want to find customers who have below $100 available for subscription; then we would check only one or two branches at the motor to get higher accuracy, we need more nodes in the branch representing different conditions of above $100 availability and so on.

Soft Computing is a branch of artificial intelligence and has been used to create algorithms that are adaptive to different strengths, weaknesses, and drawbacks in their input and response to the environment. The concept was developed in the 1970s by researchers at the University of Toronto's School of Computer Science, Canada. Another name for soft computing is fuzzy logic. Soft computing deals with modelling and analysis which use extra information that has been considered beforehand in difficult scenarios or problems. One of the main concepts behind soft computing is that it does not require precise mathematical modelling or statistical analysis to process data for solving problems. Repetitive patterns, learning rules, PSO, ABC, and using neural networks are some of the main examples **[Maulik & Sanghamitra, 2000], [Langdon *et al.*, 2005], [Mata *et al.*, 2002]**.

Swarm optimization is a type of metaheuristics that uses algorithms and Particle swarm optimization to solve optimization problems **[Jemmali *et al.*, 2022]**. It is a kind of population-based algorithm in which individual particles search for the optimal solution to a given problem. These particles are known as "particles" and are created by a well-defined set. In the beginning, all particles have an equal probability of moving toward their destination. The selected particles from this pool represent group members, who search for the optimal solution to their given task. Swarm optimization is a population-based method for designing optimal solutions to a given problem **[Beiranvand *et al.*, 2014]**. A swarm consists of particles that define the "minimal unit" through which the swarm should travel to find a solution. The average velocity, denoted by ρ, describes how fast each particle can go in its current state **[Moslehi *et al.*, 2020], [Pu *et al.*, 2021], [Agrawal *et al.*, 2015], [Karunyalakshmi *et al.*, 2017]**.

Artificial Bee Colony Optimization (ABCO) is a diagnostic and predictive algorithm for the optimal management of hives using artificial bees. The algorithm was inspired by honey bee daily activities **[Mata *et al.*, 2002]**. ABCO is divided into three bees, an employee bee, and two scout bees. The first job of an employee bee is to search for food; otherwise, the task is passed to another scout bee. In case there are multiple sources of food, chances of approval increase if the final location has more nectar than other sources. In other words, it becomes more likely that the food will be accepted if the hive already contained some of that type of nectar earlier in its life cycle, giving this location a higher probability score than others with fewer available resources **[Ishibuchi *et al.*, 2004]**.

The work presents a multi-faced evaluation performed at each level to justify the integration of individual techniques at various stages. The simulation ordinals used in the detailed results and analysis are given in Table 6.1.

Table 6.1 Simulation Ordinals

Parameters	Description
System Description	11th Gen Intel[R] Core [TM] i5-1135G7 @ 2.40GHz
Optimization Approaches	PSO, ABC, PSO with ABC, and Enhanced Group-ABC
Other Techniques	Associated Rule Mining, Mean-Variance Optimization
Classifiers Evaluated	Naïve Bayes (NB), K –Nearest Neighbor ([KNN), Support Vector Machine (SVM), Neural Network (NN)
Datasets	Baseball, and Twitter Dataset
Number of Records used for experimentation	100 out of 180 records from the Baseball dataset

	100 out of 1,000 records from the Twitter Dataset
Evaluation Parameters	Precision, Sensitivity, F-measure, Accuracy, Execution Time

6.2 Evaluation for Feature Selection Approaches

At this stage two optimization approaches namely, PSO and ABC are implemented. These are further combined and the overall evaluation is performed to find out the best feature selection approach among, PSO, ABC, PSO+ABC, and G-ABC with associated rule mining and Mean-Variance Optimization. The performance analysis has proceeded using both the baseball and Twitter datasets in terms of the number of relevant feature selections and the time required to complete the feature selection process.

- **Number of Selected Features**

At feature selection level G-ABC is introduced and evaluated in addition to other optimization approaches to compare its effectiveness in the selection of relevant features.

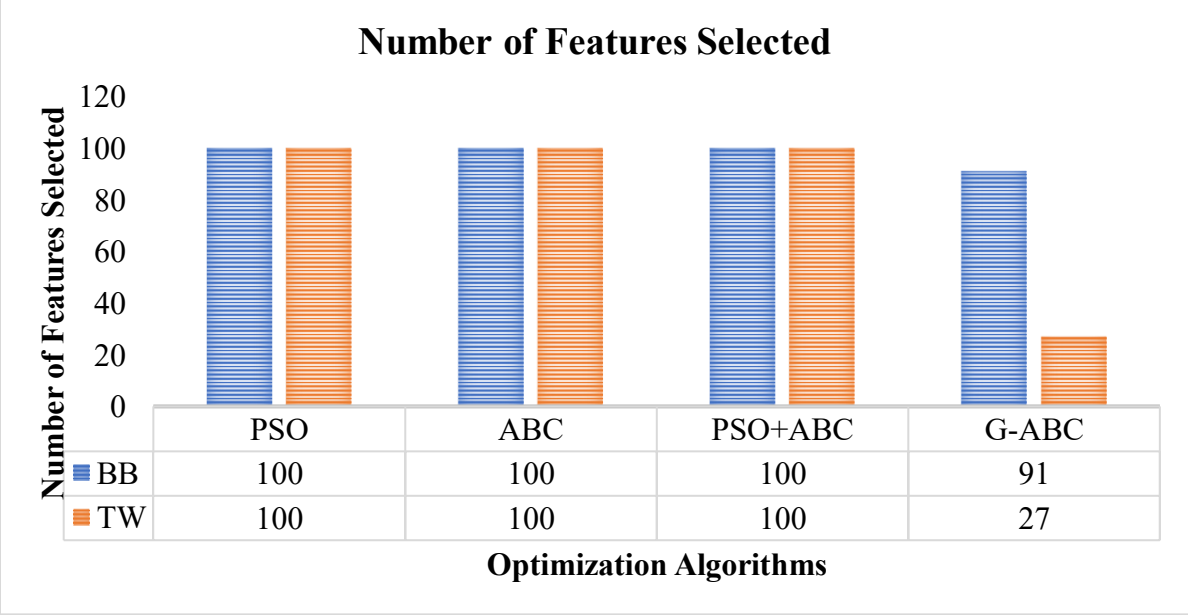

Figure 6.1 Comparison of Number of Selected Features using Different Optimization Approaches

The developed G-ABC has produced diverse sets of rules in a short period. Thus, it leads to the selection of the most relevant and important features among the available features. The number of features selected by G-ABC in comparison to the other algorithms namely PSO, ABC, and PSO+ABC are depicted in Figure 6.1. It is observed that only G-ABC is able to differentially select the most relevant features with 91 features selected from 100 records for the baseball dataset and 27 features selected for the Twitter dataset.

- **Execution Time of Feature Selection**

The time required to complete the feature selection process for each of the optimization approaches is shown in Table 6.2 and Figure 6.2. The observed variation in the execution time with an increase in the number of records for each of the optimization approaches is illustrated here.

Table 6.2 Variation in the Execution Time using Different Optimization Approaches

Number of Records	PSO	ABC	PSO+ABC	G-ABC
10	2.1425	2.145519	2.163373	2.190548
20	2.166921	2.159427	2.172766	2.208418
30	2.188156	2.207825	2.180084	2.217148
40	2.19558	2.265807	2.235953	2.281967
50	2.239146	2.295983	2.282111	2.317775
60	2.317756	2.383397	2.388566	2.426708
70	2.449061	2.467211	2.442774	2.513513
80	2.600345	2.520916	2.512178	2.610149
90	2.606163	2.671789	2.625618	2.800987
100	2.76315	2.801924	2.635788	2.87513
Average	**2.366878**	**2.39198**	**2.363921**	**2.444234**

Figure 6.2 Comparison of Execution Time Required by Different Optimization Approaches

It is generalized that with the increase in the number of records the execution time required by each of the optimisation approach increase. However, even though the proposed G-ABC is associated with slightly higher execution time, it overall performed much better than other optimization approaches when relevant feature selection is taken into consideration. Now, based on the best performance of the G-ABC, it is further evaluated using a combination of classifiers in the next stage.

6.3 Evaluation of Classification Approaches

In this section, G-ABC along with other optimization approaches is evaluated in combination with various classification approaches namely, NB, KNN, SVM, and NN for performance analysis in terms of precision, sensitivity, f-measure, accuracy, and execution time.

- **Precision Analysis**

It is the ratio of the true positive rate to the additive value of true positive and false positive. Table 6.3 shows the precision value computed using the optimization techniques namely, PSO, ABC, PSO+ABC, and the G-ABC for the feature selection using 100 records. The table shows the

precision results that depict the rule mining using the Baseball dataset with Naïve Bayes classifier, KNN, SVM, and NN.

Table 6.3 Precision Analysis using Baseball Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	0.823745	0.838557	0.855422	0.87332
20	0.830477	0.85296	0.867968	0.892204
30	0.837511	0.863847	0.877598	0.899279
40	0.85091	0.872443	0.884315	0.912732
50	0.855037	0.884131	0.887119	0.925617
60	0.857359	0.891833	0.894958	0.932553
70	0.865822	0.899011	0.895881	0.934344
80	0.872147	0.901698	0.903784	0.939309
90	0.874513	0.903283	0.90639	0.940821
100	0.877709	0.904034	0.909136	0.943652
Average	**0.854523**	**0.88118**	**0.888257**	**0.919383**
Using KNN				
10	0.85415	0.866944	0.885576	0.89166
20	0.863511	0.883058	0.886967	0.898448
30	0.876115	0.896171	0.892236	0.906983
40	0.890606	0.907058	0.906914	0.924038
50	0.89357	0.915633	0.918884	0.92582
60	0.903475	0.92034	0.924064	0.93863

70	0.904957	0.924252	0.92971	0.943693
80	0.905024	0.9268	0.932083	0.944641
90	0.906039	0.930871	0.93257	0.94789
100	0.907582	0.931101	0.93391	0.949396
Average	**0.890503**	**0.910223**	**0.914291**	**0.92712**
Using SVM				
10	0.878824	0.880844	0.905835	0.924194
20	0.888359	0.900849	0.922926	0.924414
30	0.905421	0.911332	0.93539	0.931639
40	0.910263	0.922465	0.947717	0.941762
50	0.91117	0.92846	0.960114	0.950894
60	0.913422	0.936502	0.964589	0.956329
70	0.91526	0.943421	0.965884	0.957875
80	0.918656	0.952598	0.970992	0.967213
90	0.920363	0.954641	0.97215	0.967397
100	0.922501	0.956859	0.974188	0.968351
Average	**0.908424**	**0.928797**	**0.951979**	**0.949007**
Using NN				
10	0.89147	0.914565	0.938255	0.96357
20	0.907459	0.931565	0.958327	0.97276
30	0.907518	0.943448	0.961919	0.976144
40	0.908512	0.959075	0.964106	0.979033
50	0.922057	0.97108	0.968741	0.980887
60	0.923208	0.972698	0.973033	0.984556
70	0.926701	0.983565	0.973642	0.98798

80	0.928918	0.989468	0.974022	0.981079
90	0.931168	0.990962	0.977192	0.991125
100	0.931478	0.994519	0.977258	0.992487
Average	**0.917849**	**0.965094**	**0.96665**	**0.980962**

Feature extraction has been done using different optimization techniques such as PSO, ABC, PSO+ABC, and the Group-ABC. The analysis has been done using the baseball dataset and different classifiers have been applied for better results. Table 6.3 shows that for the Naïve Bayes classifier, the precision using the baseball dataset with PSO optimization techniques for 10 records is 0.82, 0.83 using the ABC, 0.85 using the PSO+ABC, and 0.87 using the G-ABC. The increase in many records also influences the precision for rule mining. For 50 records, about 0.88 precision is obtained using ABC and using PSO+ABC while 0.85 is obtained using PSO and 0.92 using G-ABC. When the number of records doubles and approaches to 100 then the least precision value shown by using PSO and 0.90 was shown using ABC and PSO+ABC while proposed technique shows 0.95. The average value using the proposed technique approaches to 0.91 while 0.88 acquired using the ABC and PSO +ABC.

For KNN classifier, the precision using the baseball dataset with PSO optimization techniques for 30 records is 0.87 while around 0.89 is obtained using the ABC and PSO+ABC. The precision using the G-ABC proposed technique is 0.90. The increase in number of records also influences the precision for rule mining. For 80 records, about 0.90 and 0.926 precision is obtained using ABC and using ABC+PSO respectively while 0.93 is obtained using PSO and 0.94 using G-ABC. When the number of records approaches to 100 then least precision value shown by using PSO and 0.93 is acquired using ABC and PSO+ABC technique while proposed technique shows about 0.94. The average value using the proposed technique approaches to 0.92 while 0.91 acquired using the ABC and PSO +ABC.

Features extraction using the different optimization techniques for SVM classifier, the precision with PSO optimization techniques for 40 records is 0.93, 0.90 and 0.91 using the ABC and using the PSO respectively, 0.93 using the PSO+ABC and G-ABC. The increase in number of records also influences the precision for the extraction of features. For 50 records, about 0.91 and 0.92 precision is obtained using the ABC and using the PSO respectively. The precision using

PSO+ABC is 0.96 while 0.95 is obtained using G-ABC. The least average precision value shown using PSO which is 0.90 and 0.92 is shown using ABC and 0.95 for PSO+ABC while proposed technique shows about 0.95.

For NN classifier, the precision using the PSO optimization technique for 10 records is 0.89 while around 0.91 and 0.93 is obtained using the ABC and PSO+ABC respectively. The precision using the G-ABC proposed technique is 0.96. For 70 records, about 0.92 and 0.98 precision is obtained using PSO and using the ABC respectively. When the number of records approaches to 100 then least precision value shown by using PSO and maximum obtained using the G-ABC. However, the average value using the G-ABC is 0.98 and 0.96 acquired using ABC and PSO+ABC technique.

Thus, analysis results shown that G-ABC optimization technique perform well in comparison to other optimization techniques for different classifiers. The feature extraction using the different optimizers has been done and the extraction results using the PSO technique are least in comparison to other techniques.

Table 6.4 Precision Analysis using Twitter Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	0.822761	0.832808	0.847359	0.869329
20	0.8276	0.843089	0.863059	0.881348
30	0.840951	0.844391	0.866672	0.898141
40	0.852984	0.85133	0.877494	0.899264
50	0.859218	0.861236	0.881052	0.910374
60	0.863704	0.867865	0.89064	0.922705
70	0.864078	0.871336	0.890985	0.927281
80	0.871862	0.871743	0.899256	0.930141
90	0.874929	0.873402	0.901029	0.934229

100	0.875546	0.875494	0.903017	0.937333
Average	**0.855363**	**0.859269**	**0.882056**	**0.911014**
Using KNN				
10	0.85415	0.865466	0.867375	0.880912
20	0.868935	0.869877	0.876635	0.886691
30	0.874029	0.880003	0.888017	0.900757
40	0.884822	0.888665	0.902852	0.902252
50	0.884824	0.898539	0.915408	0.914442
60	0.8909	0.899216	0.915408	0.922253
70	0.897315	0.90403	0.919321	0.923946
80	0.898145	0.905151	0.926281	0.928669
90	0.898879	0.905262	0.928953	0.930422
100	0.900269	0.907232	0.929132	0.93317
Average	**0.885227**	**0.892344**	**0.906938**	**0.912351**
Using SVM				
10	0.878473	0.88195	0.886669	0.912244
20	0.894835	0.887866	0.889126	0.922124
30	0.895264	0.888729	0.89834	0.93109
40	0.899979	0.899356	0.903152	0.937398
50	0.901416	0.90291	0.911985	0.9456
60	0.90384	0.912656	0.919607	0.953069
70	0.909074	0.922099	0.926929	0.958387
80	0.915418	0.922378	0.92811	0.96086
90	0.918034	0.924152	0.932566	0.964945
100	0.918321	0.925797	0.932962	0.966663

Average	**0.903465**	**0.906789**	**0.912945**	**0.945238**
Using NN				
10	0.89147	0.913405	0.922831	0.948144
20	0.906748	0.919976	0.936324	0.958325
30	0.915531	0.93493	0.949395	0.96476
40	0.924504	0.936127	0.95965	0.964971
50	0.932085	0.94933	0.961727	0.972806
60	0.939651	0.957388	0.96844	0.975248
70	0.946752	0.965828	0.979047	0.982641
80	0.953382	0.970044	0.986428	0.98533
90	0.957508	0.971032	0.990255	0.986101
100	0.959051	0.974485	0.992948	0.988117
Average	**0.932668**	**0.949254**	**0.964704**	**0.972644**

Twitter Dataset has been considered for feature extraction using the different optimization techniques such as PSO, ABC, PSO+ABC and using the Group-ABC. The analysis has been done using the different optimization techniques, and classifiers have been implemented for better results. Table 6.4 shows that for Naïve Bayes classifier, the precision using the twitter dataset with PSO optimization techniques for 10 records is 0.82, 0.83 using the ABC, 0.847 using the PSO+ABC, and 0.869 using the G-ABC. The increase in number of records also impacts the precision value for feature extraction. For 50 records, about 0.86 precision is obtained using PSO and using ABC while 0.88 is obtained using PSO+ABC and 0.91 using G-ABC. When the number of records increased to 100 then 0.87 precision value obtained using PSO and ABC respectively while PSO+ABC shows 0.90. The average value using the proposed technique approaches to 0.91 while 0.88 acquired using the PSO +ABC and 0.85 using the PSO and ABC respectively. Thus, G-ABC shows better results for different number of records in comparison to other techniques.

For KNN classifier, the precision using the twitter dataset with PSO and ABC optimization techniques for 20 records is 0.86 while around 0.87 is obtained using the PSO+ABC. The precision

using the G-ABC proposed technique is 0.88. The increase in number of records to 80, about 0.89 precision is obtained and 0.90 is obtained using ABC while 0.92 is obtained ABC+PSO and using G-ABC. The average value using the G-ABC is 0.91 while 0.90 acquired using the PSO +ABC. SVM classifier is used after the features extraction using the different optimization techniques. The precision with PSO and ABC optimization techniques for 40 records is 0.89, and 0.90 obtained using the PSO+ABC and 0.93 for G-ABC. The increase in number of records also influences the precision for the extraction of features. For 80 records, about 0.92 precision is obtained using the ABC and using PSO+ABC while 0.96 is obtained using G-ABC. The average precision value using PSO and ABC is 0.90 and 0.91 for PSO+ABC while proposed technique shows about 0.94. For NN classifier, the precision using the PSO optimization technique for 40 records is 0.92 while around 0.93 and 0.95 is obtained using the ABC and PSO+ABC respectively. The precision using the G-ABC proposed technique is 0.96. For 80 records, about 0.95 and 0.97 precision is obtained using PSO and using the ABC respectively. The least average precision value shown by using PSO and maximum obtained using the G-ABC. However, the average value using the G-ABC is 0.97 and 0.96 acquired using the PSO+ABC technique.

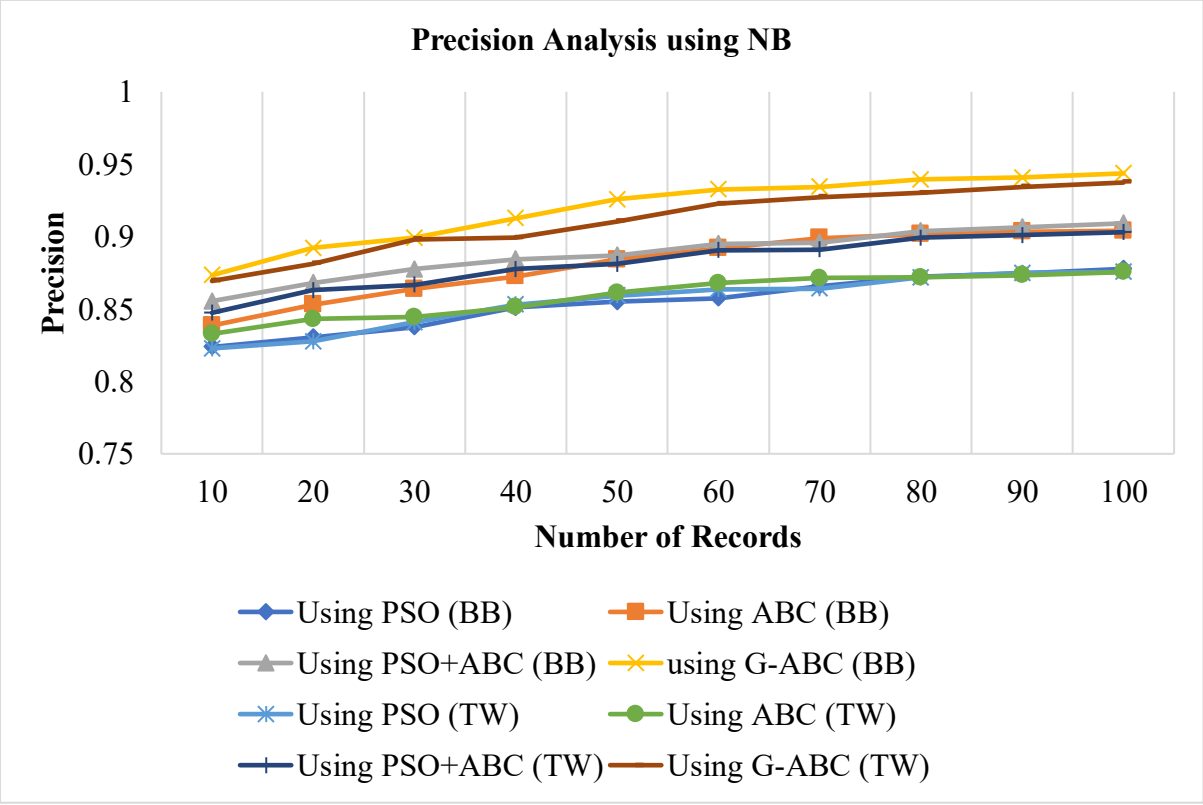

Figure 6.3 Precision Analysis using NB

Thus, analysis results shown that G-ABC optimization technique perform well in comparison to other optimization techniques for different classifiers. The feature extraction using the different optimizers has been done and the extraction results using the G-ABC technique are better for rule mining when compared with other optimization techniques.

Figure 6.3 shows the precision analysis using NB classifier for different optimization techniques using the baseball and twitter dataset. The analysis results shown when different optimization techniques has been compared then G-ABC perform well for both twitter and baseball dataset. The rising trend shown by the optimization techniques but results using G-ABC are better than other techniques.

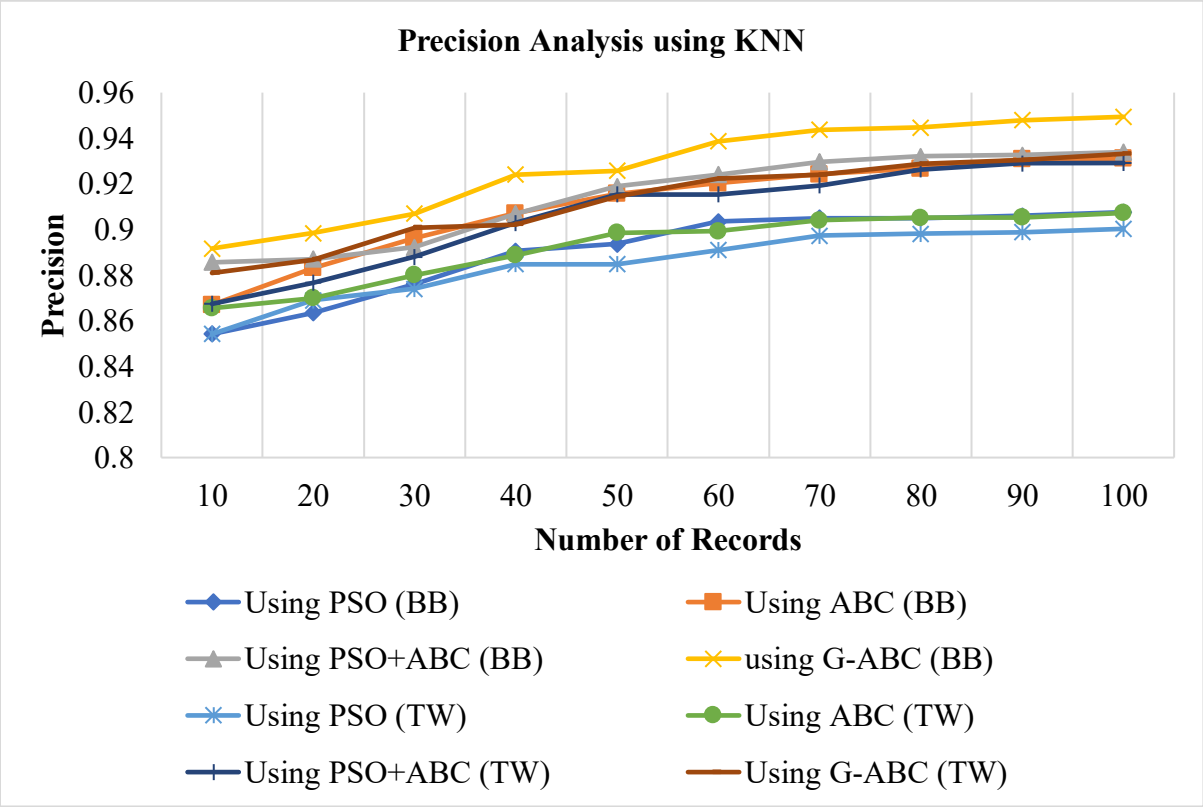

Figure 6.4 Precision Analysis using KNN

Figure 6.4 shows the precision analysis using KNN classifier for different optimization techniques using the baseball and twitter dataset. The analysis results shown when different optimization techniques has been compared then G-ABC perform well for baseball dataset followed by G-ABC for twitter dataset. Thus, results using the G-ABC are better than other techniques.

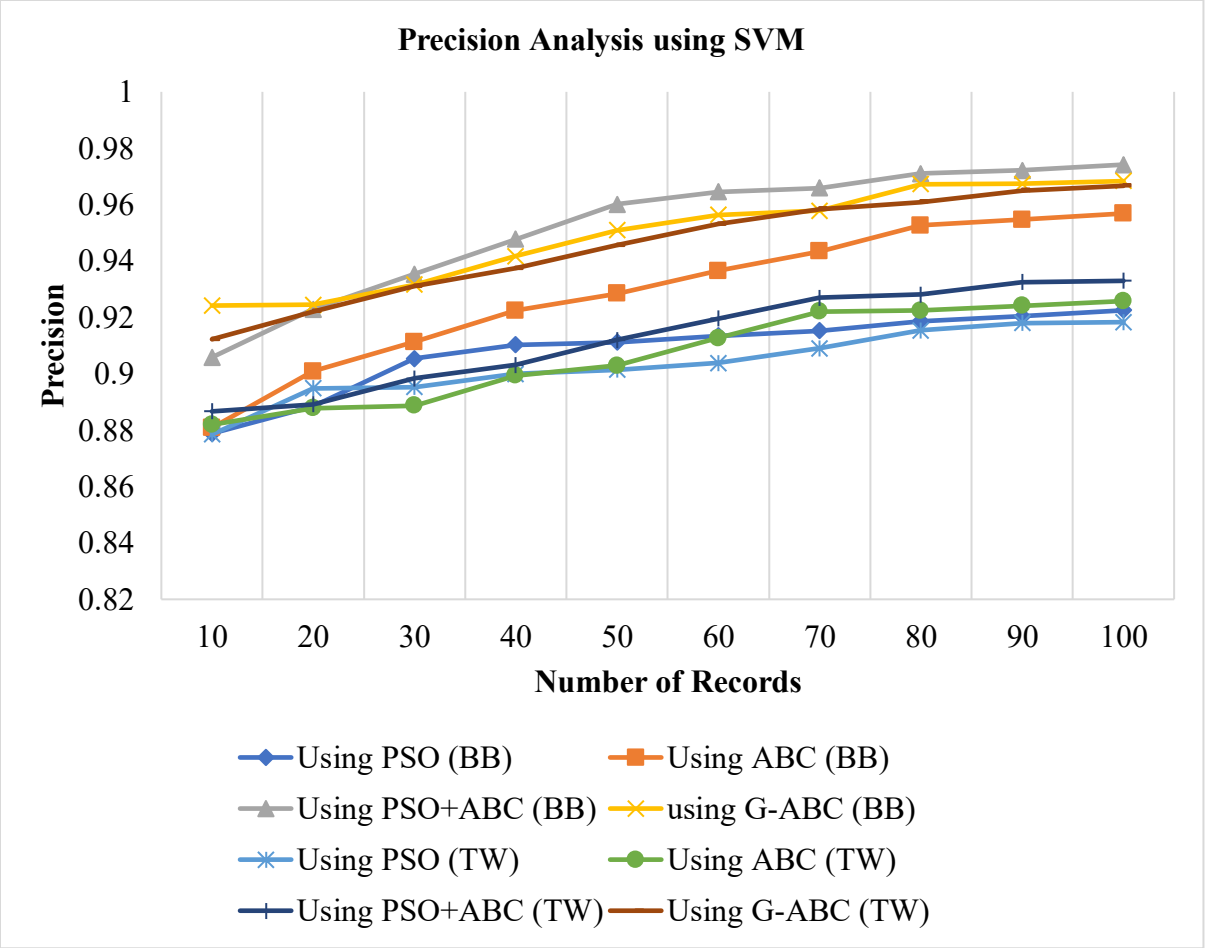

Figure 6.5 Precision analysis using SVM

Figure 6.5 shows the precision analysis using SVM classifier for different optimization techniques using the baseball and twitter dataset. The analysis results shown different optimization techniques follows the same rising trend. The techniques such as PSO+ABC perform better for 45 to 90 records using the baseball dataset. Thus, results using the G-ABC are better than other techniques using SVM classifier.

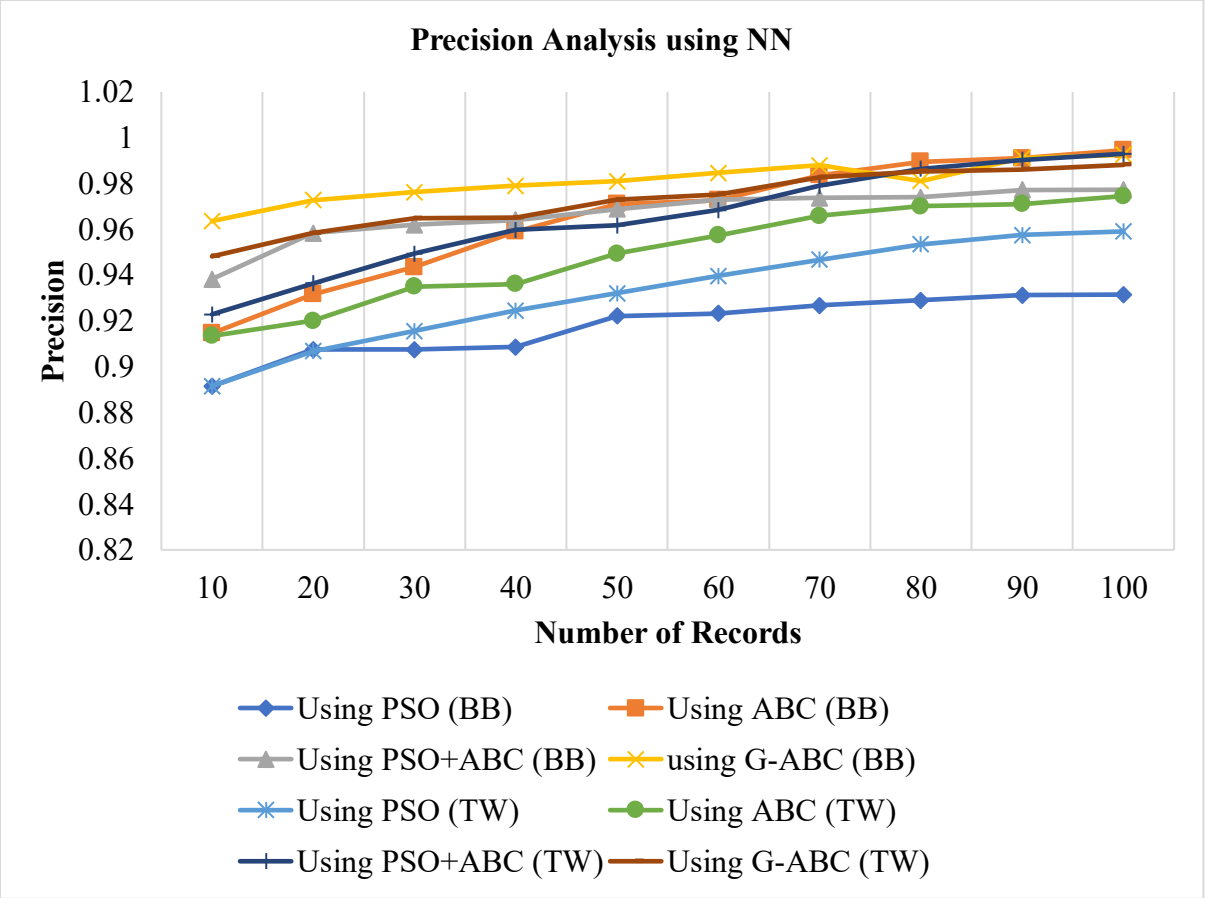

Figure 6.6 Precision analysis using NN

Figure 6.6 shows the precision analysis using NN classifier for different optimization techniques using the baseball and twitter dataset. The analysis results shown that least performance shown by the PSO technique using the baseball dataset and G-ABC using the baseball dataset perform better using NN classifier compared to other optimization techniques for feature extraction.

- **Sensitivity Analysis**

It is the ratio of the true positive rate to the additive value of true positive and false negative. Table 6.5 shows the sensitivity analysis computed using the optimization techniques namely, PSO, ABC, PSO+ABC and the G-ABC for the feature selection using 100 records. The table shows the sensitivity results that depict the rule mining using the Baseball dataset and Twitter dataset with Naïve Bayes classifier, KNN, SVM, and NN.

Table 6.5 Sensitivity Analysis using Baseball Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	0.809853	0.831718	0.852026	0.87469
20	0.823093	0.838953	0.85298	0.877498
30	0.828238	0.839532	0.856735	0.891079
40	0.830864	0.846933	0.86536	0.905017
50	0.831742	0.855636	0.867317	0.906888
60	0.843053	0.860721	0.874362	0.910734
70	0.847451	0.868602	0.878076	0.911174
80	0.848094	0.870036	0.882782	0.916949
90	0.852155	0.870638	0.883744	0.918917
100	0.852463	0.872605	0.88468	0.922271
Average	**0.836701**	**0.855537**	**0.86980677**	**0.903522**
Using KNN				
10	0.810341	0.828655	0.843783	0.865384
20	0.822619	0.84173	0.857393	0.865668
30	0.822828	0.847069	0.868862	0.881227

40	0.836303	0.851819	0.873249	0.887542
50	0.840771	0.860174	0.880666	0.897372
60	0.847155	0.868705	0.880869	0.908541
70	0.850812	0.875685	0.887568	0.916512
80	0.853752	0.876135	0.888647	0.917269
90	0.855804	0.878395	0.889122	0.919003
100	0.857557	0.880095	0.891923	0.919182
Average	**0.839794**	**0.860846**	**0.876208**	**0.89777**
Using SVM				
10	0.853752	0.867268	0.869494	0.878908
20	0.859095	0.879649	0.889491	0.899663
30	0.874061	0.884771	0.891003	0.906507
40	0.874522	0.900031	0.900638	0.918076
50	0.885176	0.906719	0.912527	0.923154
60	0.888394	0.909969	0.913605	0.935441
70	0.89016	0.916651	0.919928	0.936547
80	0.897329	0.916823	0.927781	0.940659
90	0.900952	0.921222	0.930756	0.944689
100	0.904225	0.923438	0.932319	0.946705
Average	**0.882767**	**0.902654**	**0.908754**	**0.923035**
Using NN				
10	0.877746	0.897356	0.920751	0.924608
20	0.896887	0.905066	0.943695	0.940089
30	0.901385	0.916638	0.948557	0.948237
40	0.90586	0.918337	0.959814	0.96239

50	0.915062	0.923746	0.962622	0.965609
60	0.919494	0.927906	0.964886	0.976657
70	0.930308	0.928601	0.969026	0.977726
80	0.935048	0.929686	0.969451	0.98389
90	0.938434	0.933404	0.969834	0.984636
100	0.941238	0.937032	0.970536	0.986351
Average	**0.916146**	**0.921777**	**0.957917**	**0.965019**

Table 6.5 shows that Feature extraction has been done using the different optimization techniques such as PSO, ABC, PSO+ABC and using the Group-ABC. The analysis has been done using the baseball dataset and different classifiers have been applied for better results. Table 6.5 shows that for Naïve Bayes classifier, the sensitivity using the baseball dataset with PSO optimization techniques for 10 records is 0.80, 0.83 using the ABC, 0.85 using the PSO+ABC, and 0.87 using the G-ABC. For 50 records, about 0.83 and 0.86 sensitivity is obtained using ABC and using PSO+ABC while 0.85 is obtained using PSO and 0.90 using G-ABC. When the number of records approaches to 100 then least sensitivity value shown by using PSO and 0.88 is shown using ABC and PSO+ABC while proposed technique G-ABC shows 0.92. The average value using the proposed technique approaches to 0.90 while 0.86 acquired using the PSO +ABC.

For KNN classifier, the sensitivity using the baseball dataset with PSO optimization techniques for 50 records is 0.84 while around 0.88 is obtained using the PSO+ABC. The sensitivity using the G-ABC proposed technique is 0.89. The increase in number of records also influences the sensitivity for rule mining. For 80 records, about 0.85 and 0.88 sensitivity is obtained using PSO and using ABC+PSO respectively while 0.87 is obtained using PSO and 0.91 using G-ABC. The average value using the proposed technique approaches to 0.89 while 0.86 and 0.87 acquired using the ABC and PSO +ABC respectively.

Using the SVM classifier, the sensitivity analysis with ABC optimization techniques for 80 records is 0.91 while around 0.92 is obtained using the PSO+ABC. The sensitivity using the G-ABC proposed technique is 0.94. The increase in number of records also impacts the sensitivity for rule mining. For 100 records, about 0.90 and 0.93 sensitivity is obtained using PSO and using

ABC+PSO respectively while 0.90 is obtained using ABC and 0.94 using G-ABC. The average value using the proposed technique approaches to 0.92 while 0.90 using the ABC and PSO +ABC. However, the average value using the NN classifier using the proposed technique is 0.96, 0.95 for ABC+PSO, 0.92 for ABC, and 0.91 for PSO.

Table 6.6 Sensitivity Analysis using Twitter Dataset

Using Naïve Bayes				
Number of Records	Using PSO	Using ABC	Using PSO+ABC	Using G-ABC
10	0.810	0.834	0.850	0.861
20	0.812	0.840	0.870	0.880
30	0.813	0.848	0.874	0.896
40	0.818	0.850	0.887	0.905
50	0.825	0.858	0.887	0.918
60	0.833	0.865	0.899	0.919
70	0.834	0.874	0.904	0.929
80	0.837	0.878	0.904	0.938
90	0.838	0.882	0.908	0.941
100	0.840	0.882	0.909	0.944
Average	**0.826021**	**0.861099**	**0.889147**	**0.913056**
Using KNN				
10	0.811	0.827	0.850	0.868
20	0.811	0.848	0.851	0.881
30	0.815	0.854	0.867	0.886
40	0.819	0.869	0.878	0.890
50	0.831	0.870	0.887	0.900

60	0.833	0.872	0.892	0.900
70	0.842	0.876	0.900	0.909
80	0.844	0.883	0.907	0.911
90	0.847	0.884	0.912	0.915
100	0.848	0.885	0.912	0.918
Average	**0.830116**	**0.866748**	**0.885623**	**0.897912**
Using SVM				
10	0.854	0.876	0.894	0.909
20	0.859	0.897	0.907	0.913
30	0.872	0.900	0.915	0.920
40	0.880	0.911	0.930	0.924
50	0.892	0.920	0.935	0.925
60	0.899	0.924	0.943	0.931
70	0.905	0.924	0.948	0.935
80	0.909	0.929	0.955	0.944
90	0.913	0.931	0.958	0.946
100	0.915	0.931	0.958	0.947
Average	**0.889635**	**0.914286**	**0.934347**	**0.929517**
Using NN				
10	0.878	0.895	0.903	0.930
20	0.885	0.912	0.907	0.933
30	0.900	0.924	0.926	0.952
40	0.906	0.925	0.929	0.966
50	0.911	0.936	0.938	0.979
60	0.921	0.938	0.945	0.989

70	0.924	0.940	0.946	0.991
80	0.930	0.949	0.954	0.995
90	0.931	0.951	0.958	0.998
100	0.934	0.953	0.961	0.999
Average	**0.911796**	**0.932248**	**0.93678**	**0.97324**

Table 6.6 shows that using the Naïve Bayes classifier, the average sensitivity value using the G-ABC technique approaches to 0.91 while 0.86 and 0.88 acquired using the ABC and PSO +ABC respectively.

Using the KNN classifier, the average sensitivity value using the G-ABC technique is 0.89, 0.88 acquired using the ABC+PSO, 0.86 obtained using the ABC technique, and 0.83 using the PSO technique.

The similar trend is seen using the SVM classifier with improved sensitivity value for G-ABC which is 0.93 while 0.91 and 0.88 obtained using the PSO and ABC technique. Thus, improved results obtained by implementing the G-ABC technique.

The sensitivity analysis using the NN classifier shown that 0.91 obtained using the PSO, 0.93 using the ABC and PSO+ABC while results using the G-ABC technique is 0.97. Thus, there is an improvement in sensitivity analysis using the G-ABC technique.

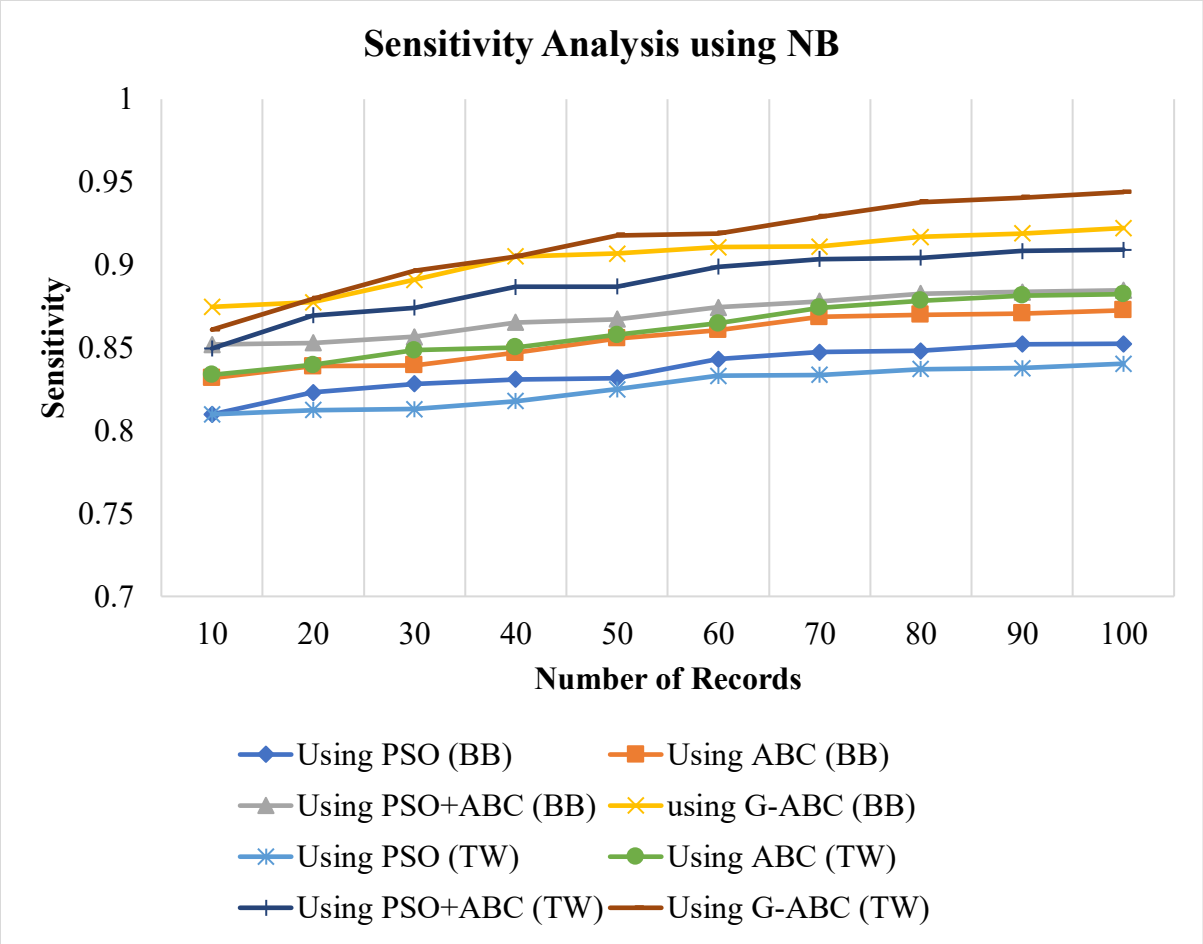

Figure 6.7 Sensitivity Analysis using NB

Figure 6.7 shows the sensitivity analysis using NB classifier for different optimization techniques using the baseball and twitter dataset. The sensitivity analysis results show that least performance shown using the PSO technique for twitter dataset and G-ABC using the twitter dataset perform better in comparison to other optimization techniques for feature extraction.

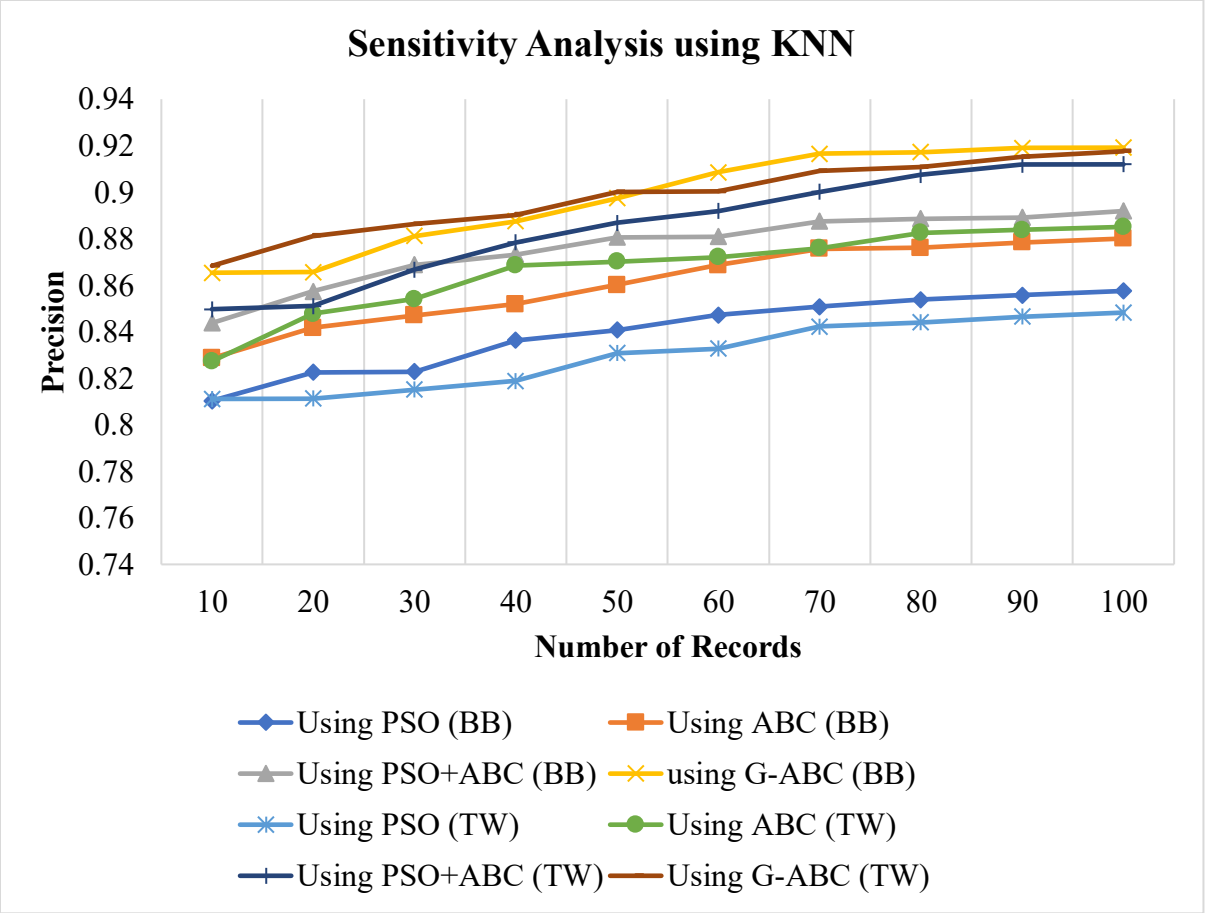

Figure 6.8 Sensitivity Analysis using KNN

Figure 6.8 shows the sensitivity analysis using the KNN classifier for different optimization techniques using the baseball and twitter dataset. The sensitivity analysis result shown that least performance is obtained using the PSO technique for twitter dataset and better performance is shown using the G-ABC for baseball dataset.

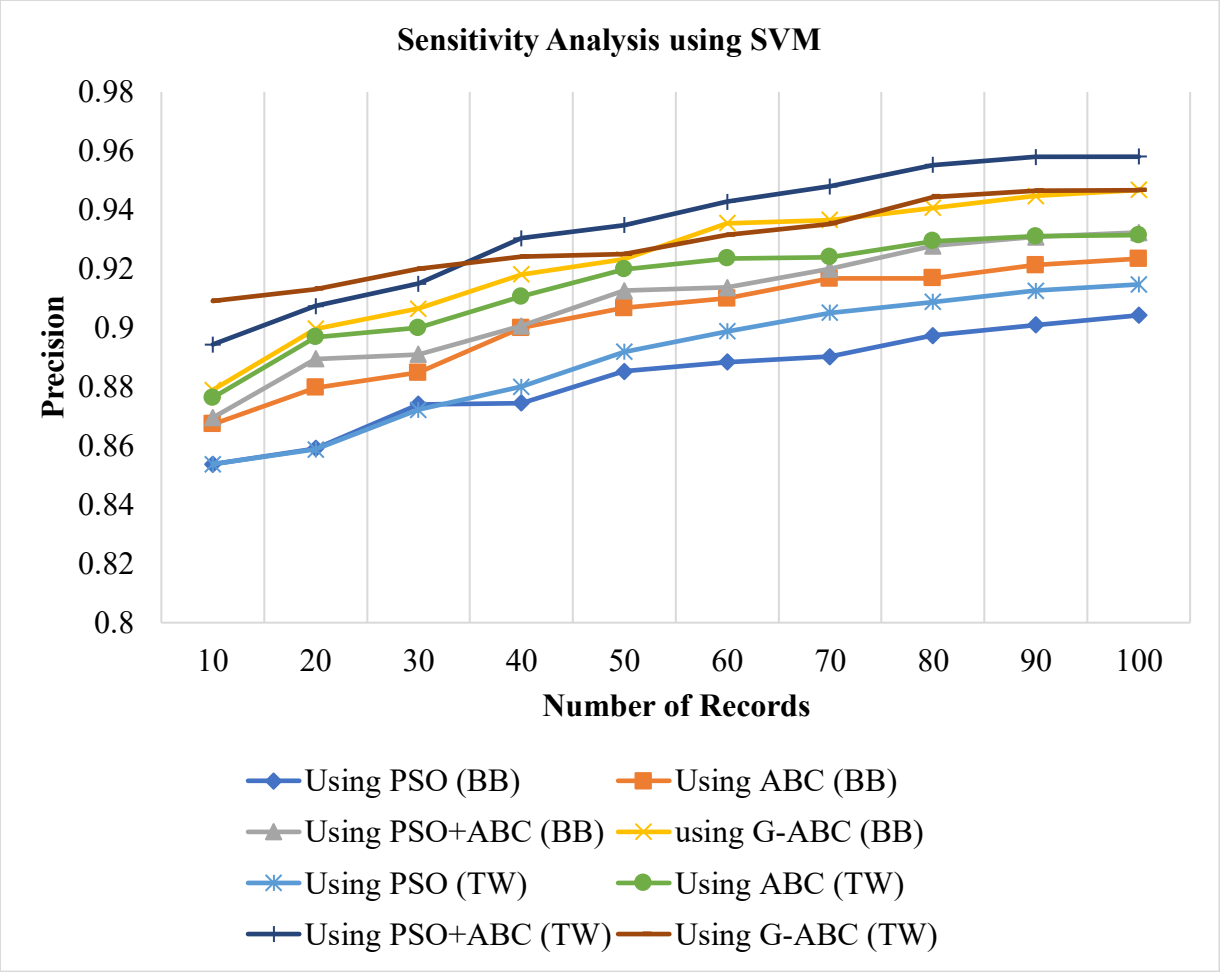

Figure 6.9 Sensitivity Analysis using SVM

Figure 6.9 shows the sensitivity analysis using the SVM classifier for different optimization techniques using the baseball and twitter dataset. The sensitivity analysis result shown that least performance is obtained using the PSO technique for baseball dataset and better performance shown using the PSO+ABC for twitter dataset.

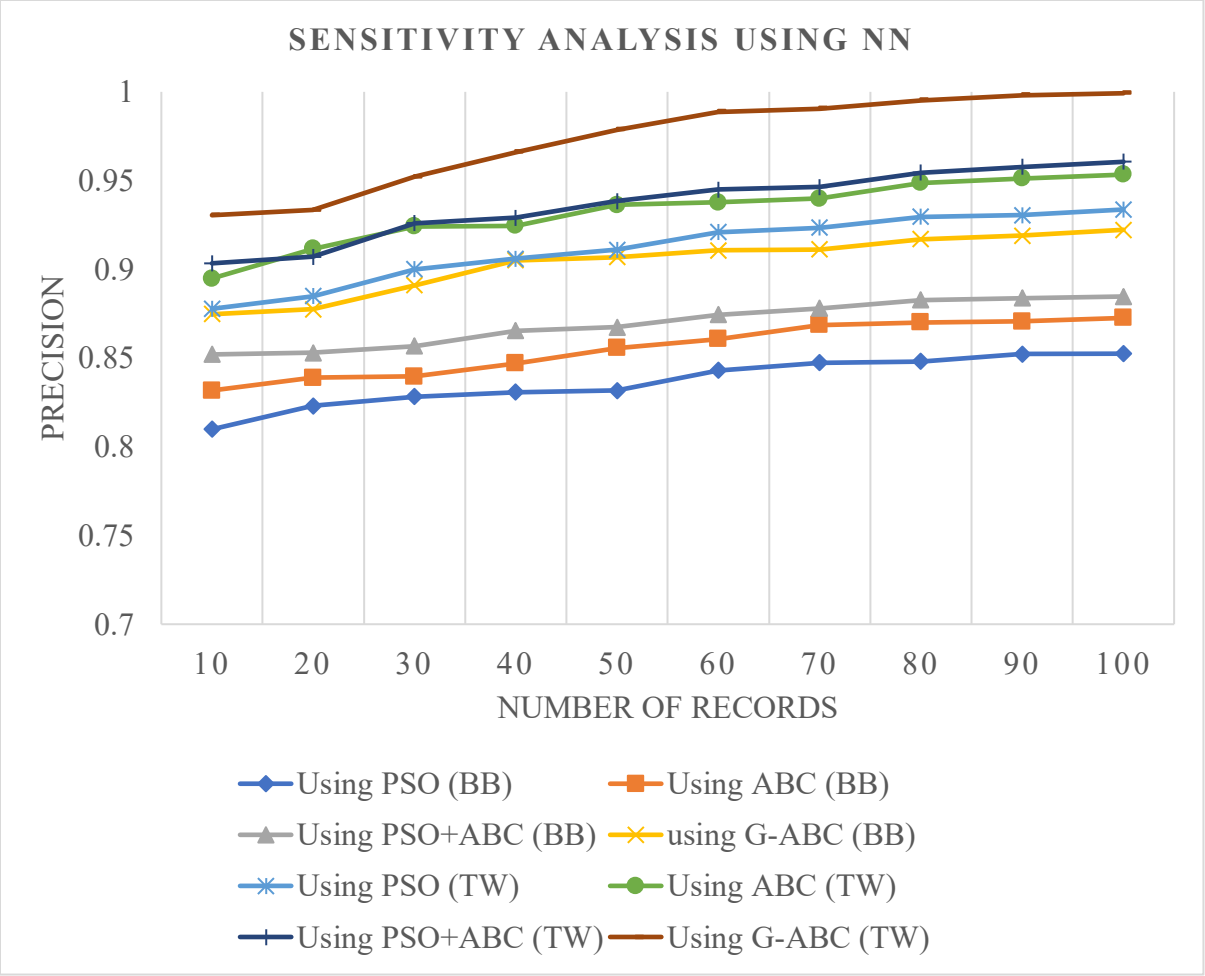

Figure 6.10 Sensitivity Analysis using NN

Figure 6.10 shows the sensitivity analysis using the NN classifier for different optimization techniques using the baseball and twitter dataset. The sensitivity analysis result shown that least performance is obtained using the PSO technique for baseball dataset and better performance is obtained using the G-ABC technique for twitter dataset.

- **F-measure Analysis**

It is the twice of ratio of product of recall and precision to the sum of recall and precision. Table 6.5 shows the F-measure analysis computed using the optimization techniques namely, PSO, ABC, PSO+ABC and the G-ABC for the feature selection using 100 records. The table shows the F-measure results that depict the feature extraction using the Baseball dataset and Twitter dataset with Naïve Bayes classifier, KNN, SVM, and NN.

Table 6.7 F-measure Analysis using Baseball Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	0.81674	0.835123	0.85372	0.874004
20	0.826768	0.845899	0.860409	0.88479
30	0.832848	0.851516	0.867041	0.89516
40	0.840768	0.859499	0.874735	0.908858
50	0.843228	0.86965	0.877106	0.916157
60	0.850146	0.876001	0.88454	0.921514
70	0.856538	0.883545	0.886889	0.922613
80	0.859952	0.885584	0.893159	0.927994
90	0.863189	0.88666	0.894924	0.92974
100	0.864901	0.888042	0.896742	0.932839
Average	**0.845508**	**0.868152**	**0.878927**	**0.911367**
Using KNN				
10	0.831669	0.847367	0.864175	0.878325
20	0.842569	0.861899	0.87193	0.881753
30	0.848636	0.870928	0.880394	0.893919
40	0.862601	0.878571	0.889763	0.905422
50	0.866367	0.887037	0.899369	0.911374
60	0.874409	0.893777	0.90195	0.92334
70	0.87705	0.899313	0.908151	0.929904
80	0.878641	0.900755	0.909847	0.930754
90	0.880206	0.903872	0.910328	0.933223

100	0.881861	0.90488	0.912434	0.934045
Average	**0.864401**	**0.88484**	**0.894834**	**0.912206**
Using SVM				
10	0.866107	0.874003	0.887293	0.900982
20	0.873482	0.890123	0.9059	0.911871
30	0.889465	0.897856	0.912657	0.918901
40	0.892035	0.91111	0.923578	0.929768
50	0.897985	0.917461	0.935716	0.936819
60	0.900734	0.923045	0.938405	0.94577
70	0.902535	0.929844	0.942346	0.947091
80	0.907868	0.934368	0.948895	0.953751
90	0.910554	0.937634	0.951003	0.955908
100	0.913271	0.939852	0.952794	0.957406
Average	**0.895404**	**0.915529**	**0.929859**	**0.935827**
Using NN				
10	0.884555	0.905879	0.929421	0.943687
20	0.902142	0.918124	0.950955	0.956145
30	0.904441	0.92985	0.955192	0.961988
40	0.907184	0.938264	0.961955	0.97064
50	0.918546	0.946822	0.965672	0.973188
60	0.921347	0.949774	0.968943	0.980591
70	0.928501	0.955293	0.971329	0.982826
80	0.931973	0.958646	0.971731	0.982482
90	0.934787	0.961323	0.973499	0.98787
100	0.936332	0.96492	0.973886	0.98941

Average	**0.916981**	**0.942889**	**0.962258**	**0.972883**

Table 6.7 shows that results using the Naïve Bayes classifier using the baseball dataset, the average F-measure value using the G-ABC technique approaches to 0.91 while 0.86 and 0.84 acquired using the ABC and PSO technique while 0.87 is obtained using the PSO+ABC.

The results using the KNN classifier, the average F-measure value using the G-ABC technique is 0.91, 0.89 acquired using the PSO+ABC, 0.86 obtained using the PSO technique, and 0.88 obtained using the ABC technique.

The similar trend is seen using the SVM classifier with improved F-measure value for G-ABC which is 0.93 while 0.89 and 0.91 is obtained using the PSO and ABC technique. Thus, improved results obtained by implementing the G-ABC technique.

The F-measure analysis using the NN technique shown that 0.91 obtained using the PSO, 0.94 and 0.96 is obtained using the ABC and PSO+ABC respectively while results using the G-ABC technique is 0.97. Thus, G-ABC technique provides better results.

Table 6.8 F-measure Analysis using Twitter Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	0.816256	0.83324	0.848474	0.865106
20	0.819972	0.841402	0.866274	0.880479
30	0.826706	0.846417	0.870391	0.897297
40	0.835049	0.850757	0.88212	0.902137
50	0.841794	0.859546	0.883996	0.914109
60	0.848114	0.866255	0.894739	0.920806
70	0.848598	0.872741	0.897218	0.928198
80	0.854177	0.875058	0.901811	0.934049
90	0.855893	0.877452	0.904721	0.937451

100	0.857624	0.87891	0.906111	0.940631
Average	**0.840418**	**0.860178**	**0.885586**	**0.912026**
Using KNN				
10	0.832082	0.845971	0.858454	0.874592
20	0.839097	0.85866	0.863682	0.883943
30	0.843582	0.866875	0.877271	0.893482
40	0.85057	0.878509	0.890475	0.896207
50	0.856984	0.884135	0.900914	0.907148
60	0.860822	0.88545	0.903491	0.911168
70	0.868936	0.889791	0.909643	0.916449
80	0.870289	0.893692	0.916757	0.919661
90	0.871916	0.894406	0.920334	0.92273
100	0.873492	0.896027	0.920475	0.925319
Average	**0.856777**	**0.879352**	**0.89615**	**0.90507**
Using SVM				
10	0.865936	0.879117	0.890484	0.910655
20	0.876407	0.892327	0.898151	0.917613
30	0.883555	0.894301	0.906545	0.925495
40	0.889873	0.90497	0.916548	0.930703
50	0.896623	0.911313	0.923242	0.935156
60	0.901289	0.918082	0.931015	0.94211
70	0.907093	0.923013	0.937319	0.946655
80	0.912077	0.925856	0.94145	0.95251
90	0.915271	0.927548	0.945067	0.955578
100	0.916505	0.928602	0.945317	0.956543

Average	**0.896463**	**0.910513**	**0.923514**	**0.937302**
Using NN				
10	0.884555	0.90403	0.913001	0.939229
20	0.8957	0.915743	0.921464	0.945653
30	0.907748	0.92957	0.937523	0.958421
40	0.915103	0.930318	0.944063	0.965502
50	0.92141	0.942709	0.949948	0.975741
60	0.930206	0.947481	0.956551	0.98192
70	0.935	0.952697	0.962383	0.98657
80	0.941357	0.959287	0.970192	0.990267
90	0.943817	0.961031	0.973705	0.99203
100	0.946207	0.963807	0.976492	0.993637
Average	**0.92211**	**0.940667**	**0.950532**	**0.972897**

Table 6.7 shows that results using the Naïve Bayes classifier using the twitter dataset, the average F-measure value using the G-ABC technique approaches to 0.91 while 0.86 and 0.84 acquired using the ABC and PSO technique, and 0.88 is obtained using the PSO+ABC.

The F-measure results using the KNN classifier, the average value using the G-ABC technique is 0.90, 0.89 acquired using the PSO+ABC, 0.85 obtained using the PSO technique, and 0.87 is obtained using the ABC technique. The similar trend is seen using the SVM classifier with improved F-measure value for G-ABC which is 0.93 while 0.89 is obtained using the ABC technique. Thus, improved results obtained by implementing the G-ABC technique.

Similarly, the F-measure analysis using the NN technique shown that 0.97 is obtained using the G-ABC, and 0.94 and 0.95 is obtained using the ABC and PSO+ABC respectively. Thus, G-ABC technique provides better results in terms of F-measure.

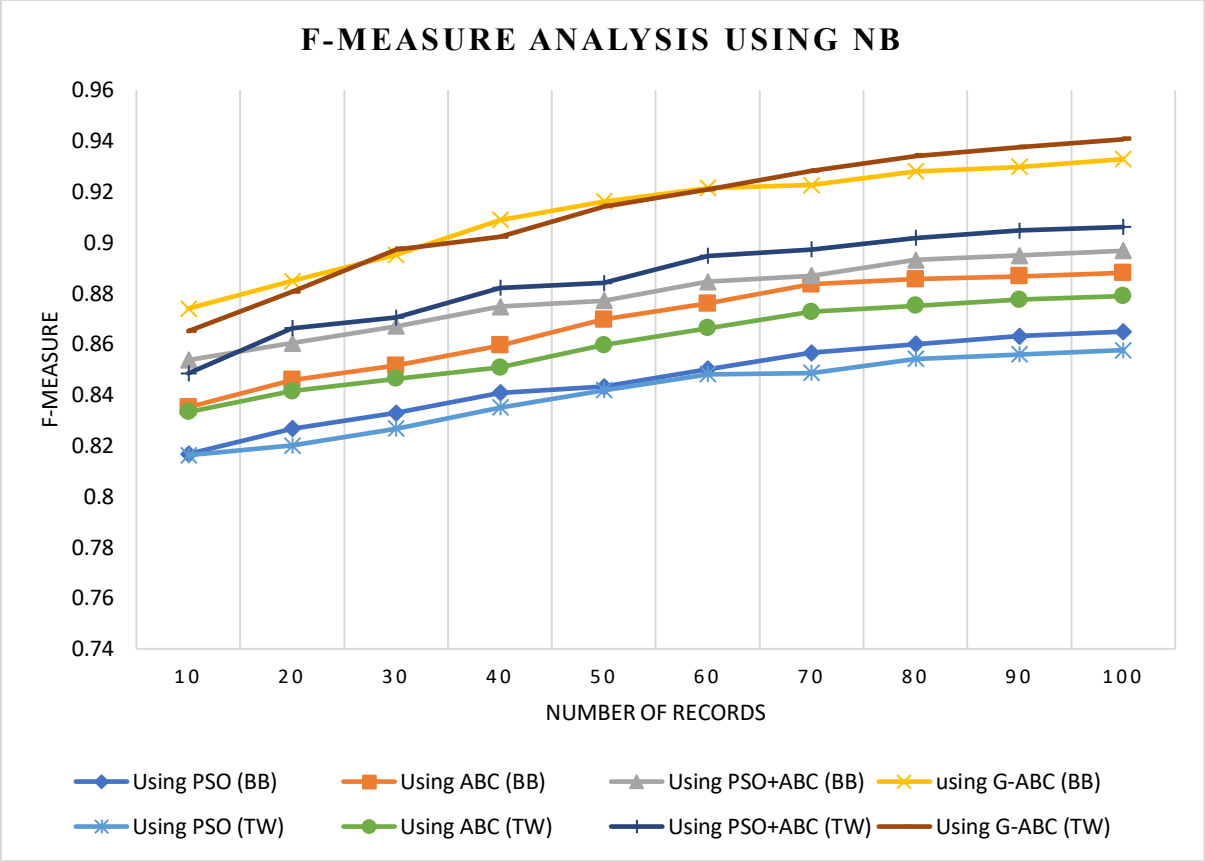

Figure 6.11 F-measure Analysis using NB

Figure 6.11 shows the F-measure analysis using the NB classifier for different optimization techniques using the baseball and twitter dataset. The F-measure analysis result shown that least performance is obtained using the PSO technique for baseball dataset and better performance is obtained using the G-ABC technique for twitter dataset.

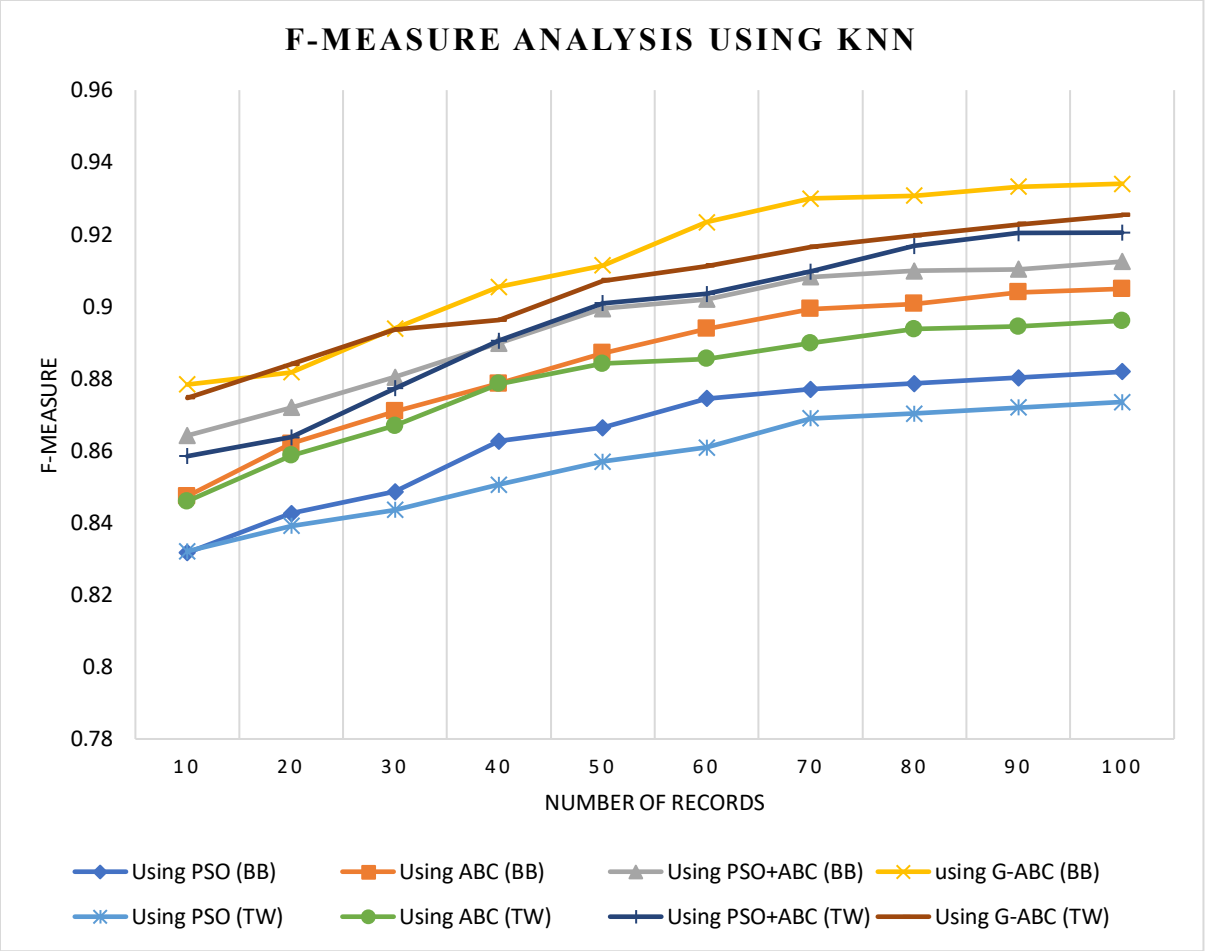

Figure 6.12 F-measure Analysis using KNN

Figure 6.12 shows the F-measure analysis using the KNN classifier for different optimization techniques using the baseball and twitter dataset. The F-measure analysis result shown that least performance is shown by using the PSO technique for twitter dataset and better performance is shown using the G-ABC technique for baseball dataset.

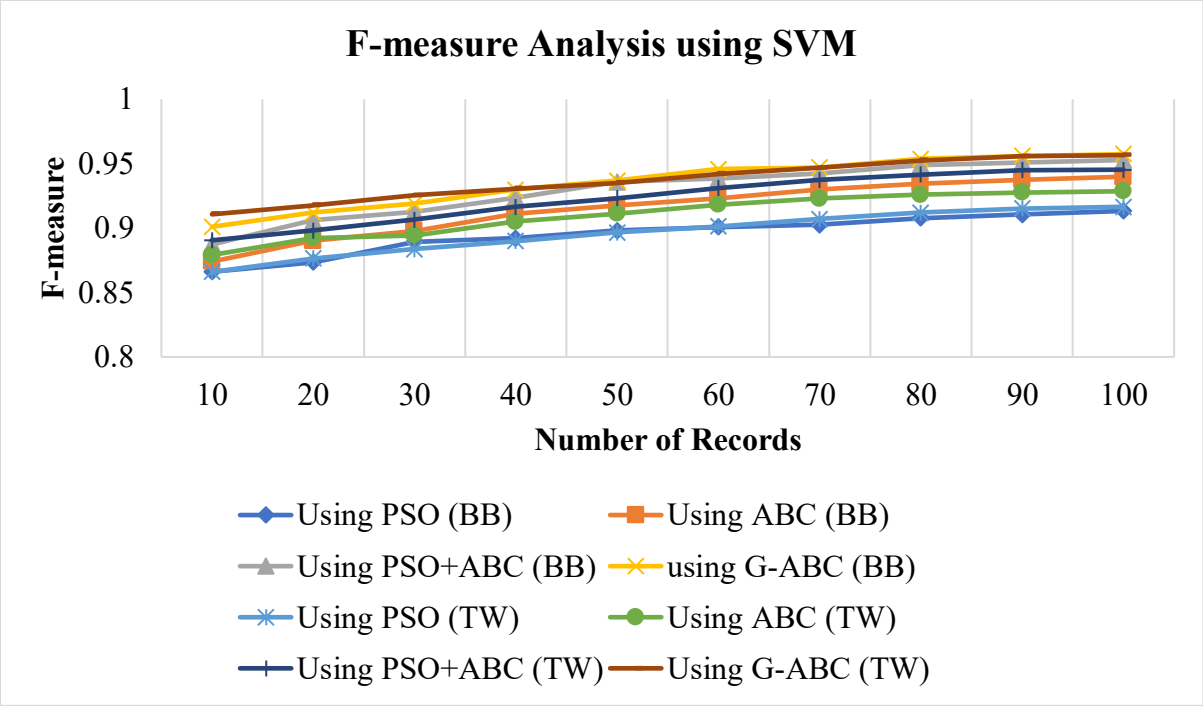

Figure 6.13 F-measure Analysis using SVM

Figure 6.13 shows the F-measure analysis using the SVM classifier for different optimization techniques using the baseball and twitter dataset. The F-measure analysis result shown that least performance is shown by using the PSO technique and ABC technique for twitter dataset and better performance is shown using the G-ABC technique for both datasets.

Figure 6.14 shows the F-measure analysis using the NN classifier for different optimization techniques using the baseball and twitter dataset. The F-measure analysis result shown that least performance is shown by using the PSO technique for baseball dataset and better performance is shown using the G-ABC technique for twitter dataset.

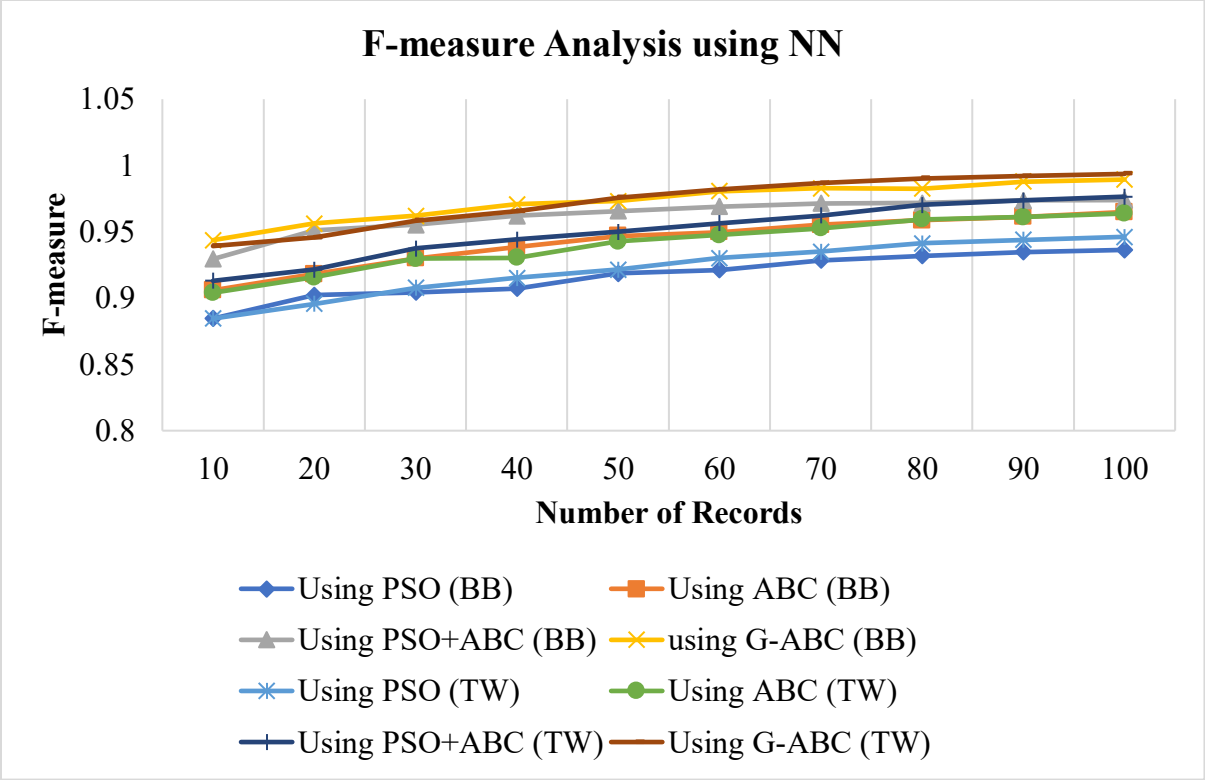

Figure 6.14 F-measure Analysis using NN

- **Accuracy Analysis**

Accuracy is defined the probability of true positive rates for different number of records. Table 6.9 shows the Accuracy analysis computed using the optimization techniques namely, PSO, ABC, PSO+ABC and the G-ABC for the feature selection using 100 records. The table shows the F-measure results that depict the feature extraction using the Baseball dataset and Twitter dataset with Naïve Bayes classifier, KNN, SVM, and NN.

Table 6.9 Accuracy Analysis using Baseball Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	84.2247	85.1923	87.62382	89.28796
20	86.13563	85.42997	88.12569	89.55157

30	87.58586	86.48356	88.25582	89.64192
40	87.61614	87.90417	88.78433	91.05439
50	88.2091	88.72209	89.65062	91.52012
60	89.0752	89.38447	90.20528	91.88144
70	89.6685	89.48034	91.15097	92.39542
80	89.72911	90.02522	91.64957	92.91863
90	89.76063	90.14744	91.99284	92.97195
100	89.89588	90.20004	92.22606	93.14151
Average	**88.19008**	**88.29696**	**89.9665**	**91.43649**
Using KNN				
10	87.1384	89.91836	89.97421	92.17586
20	87.84224	90.83008	91.71874	93.06898
30	89.46136	91.03868	92.72111	93.26042
40	90.33741	91.79074	93.43857	94.01556
50	91.44344	93.04017	93.96355	94.27992
60	92.53243	93.83694	95.02385	95.40958
70	93.21362	94.05143	95.06481	96.32888
80	93.76608	94.92248	95.49345	96.5046
90	93.90228	95.01122	95.92385	96.60539
100	93.93443	95.37042	96.12076	96.87673
Average	**91.35717**	**92.98105**	**93.94429**	**94.85259**
Using SVM				
10	88.317	90.59888	91.61803	93.01948
20	90.78011	92.60389	92.05347	94.91345
30	92.46066	92.68478	93.92095	96.06904

40	93.129	94.22225	94.15198	97.02441
50	93.83168	95.16283	94.22955	97.99198
60	94.90878	96.1617	94.53426	98.61951
70	95.85349	96.2	94.73752	98.97671
80	95.89612	96.25974	95.24242	99.96044
90	96.22433	96.26673	95.66608	100.2353
100	96.40235	96.54329	95.95044	100.375
Average	**93.78035**	**94.67041**	**94.21047**	**97.71853**
Using NN				
10	89.2145	90.68314	91.84231	94.18141
20	89.55096	91.85599	92.07305	95.98927
30	89.92157	92.7711	93.3536	96.76791
40	90.89646	93.99328	93.58424	98.52984
50	91.75805	95.01344	94.62896	99.28529
60	92.18892	95.07063	95.09501	99.39578
70	92.6035	96.0418	95.14925	99.39985
80	92.93373	96.20106	95.68815	99.45846
90	92.97308	96.23833	95.91031	99.4875
100	93.26101	96.463	96.12101	99.49265
Average	**91.53018**	**94.43318**	**94.34459**	**98.1988**

Table 6.9 shows that results using the Naïve Bayes classifier using the baseball dataset, the average F-measure value accuracy using the ABC and PSO technique is 88.19 while 89.9 is obtained using the PSO+ABC. The G-ABC technique shows more prominent results in comparison to other techniques.

The results using the KNN classifier, the average accuracy value using the G-ABC technique is 94.85% 93.9% is acquired using the PSO+ABC, 92.98 is obtained using the ABC technique, and 91.35 is obtained using the ABC technique.

The similar trend is seen using the SVM classifier with improved accuracy value for G-ABC which is 97.71 while 94.67 and 93.78 is obtained using the ABC and PSO technique respectively. Thus, improved results obtained by implementing the G-ABC technique with SVM classifier.

The accuracy analysis using the NN technique shown that 91.5% accuracy is obtained using the PSO, 94% accuracy is obtained using the ABC and PSO+ABC respectively while results using the G-ABC technique is 98%. Thus, G-ABC technique performs better in comparison to other techniques.

Table 6.10 Accuracy Analysis using Twitter Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	84.2247	86.60822	87.16086	89.19464
20	85.83663	86.87842	88.42455	89.35538
30	85.92312	86.8789	89.46208	90.02647
40	86.15758	88.36173	89.71763	90.62457
50	86.96031	88.63135	90.45303	91.38295
60	87.10424	89.68456	91.60777	92.36969
70	87.5856	90.52322	92.36509	93.3246
80	88.44264	90.81338	93.2688	93.56649
90	88.80332	90.9107	93.64936	93.56854
100	88.90724	91.12337	93.72176	93.63128
Average	**86.99454**	**89.04138**	**90.98309**	**91.70446**
Using KNN				
10	87.657	88.17207	90.4826	92.36148

20	88.40116	88.62171	91.16707	93.43336
30	89.35825	89.45983	92.05222	93.80362
40	89.95276	90.19569	92.88689	95.27012
50	90.12094	90.75184	93.10714	95.49272
60	91.12458	91.55334	93.20341	95.64905
70	92.08084	91.98866	94.15492	95.67352
80	92.42254	92.73888	94.99645	95.96956
90	92.49671	92.94922	95.31748	96.27091
100	92.68572	93.16578	95.354	96.45123
Average	**90.63005**	**90.9597**	**93.27222**	**95.03756**
Using SVM				
10	88.657	89.82322	91.76841	93.67312
20	89.84077	91.79266	93.32765	94.61962
30	90.4658	93.5022	93.64474	95.59494
40	91.96885	93.69384	95.13479	95.85002
50	93.10229	94.36681	95.64308	96.45411
60	93.41677	94.73048	96.22336	96.47795
70	93.53668	94.8253	97.217	97.49542
80	93.98253	95.17728	97.22927	97.82886
90	94.04336	95.18494	97.31732	98.1356
100	94.21586	95.23472	97.36859	98.27149
Average	**92.32299**	**93.83315**	**95.48742**	**96.44012**
Using NN				
10	89.883	92.34625	93.90708	94.99266
20	91.28864	94.53702	95.59192	95.34881

30	92.66935	96.32538	96.3905	96.95189
40	93.96749	96.38955	97.50792	98.07594
50	95.10257	96.96589	97.80704	98.99142
60	96.30141	97.09245	98.74675	99.346
70	96.78259	97.88619	98.98287	99.42878
80	97.26718	98.11573	99.20933	99.69879
90	97.62529	98.57671	99.42271	99.68884
100	97.91163	98.66335	99.52163	99.654
Average	**94.87991**	**96.68985**	**97.70877**	**98.21771**

Table 6.10 shows that the accuracy analysis using the NB, KNN, SVM, and NN classifier for twitter dataset. The results shown that using the NB classifier, 86.99% accuracy is obtained using the PSO, 89% and 90% accuracy is obtained using the ABC and PSO+ABC respectively while results using the G-ABC technique is 91.7%. Furthermore, 98.21% and 96.4% accuracy is obtained using the G-ABC technique for NN and SVM classifier. The performance of the G-ABC technique is superior in comparison to other optimization techniques.

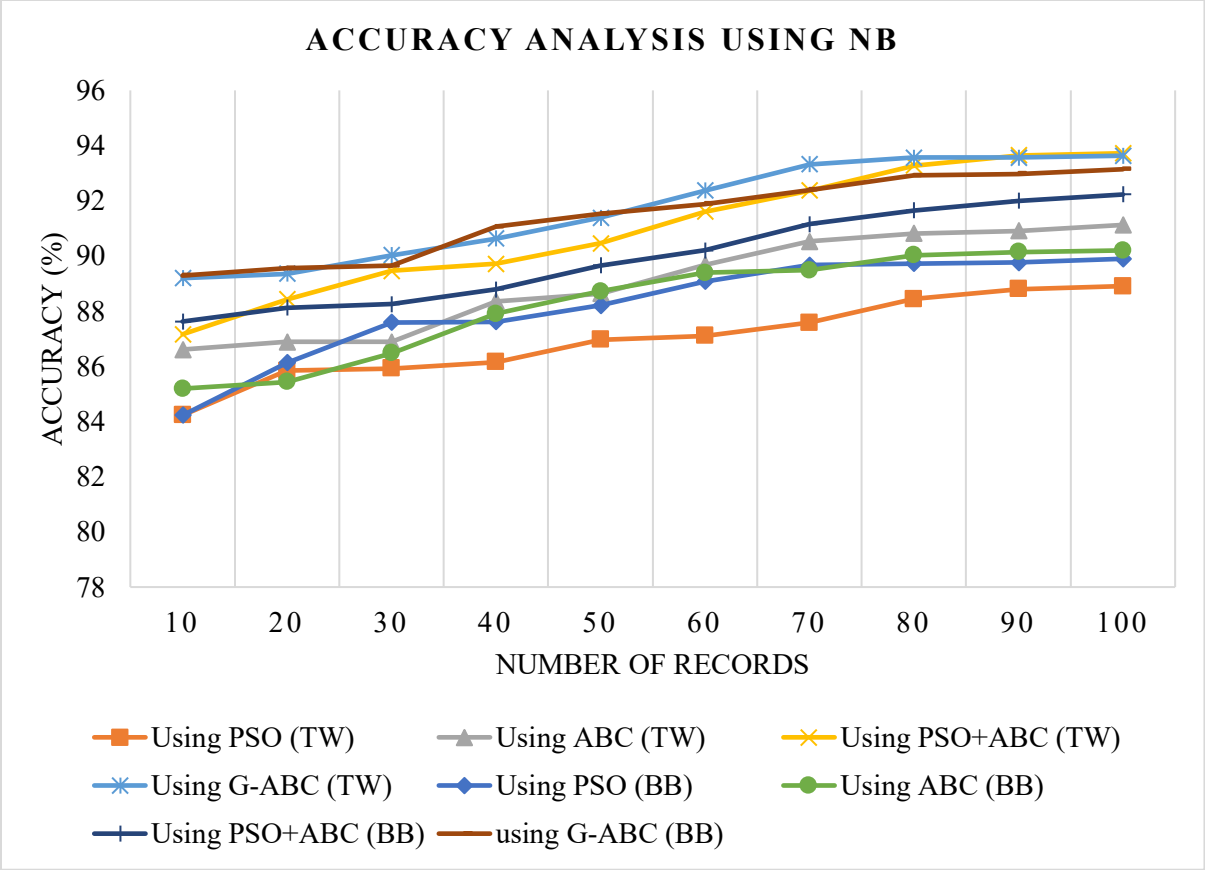

Figure 6.15 Accuracy Analysis using NB

Figure 6.15 shows the accuracy analysis using the NB technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for twitter dataset and least performance is shown using the PSO technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques.

Figure 6.16 shows the accuracy analysis using the KNN technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for twitter dataset and least performance is shown using the PSO technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques

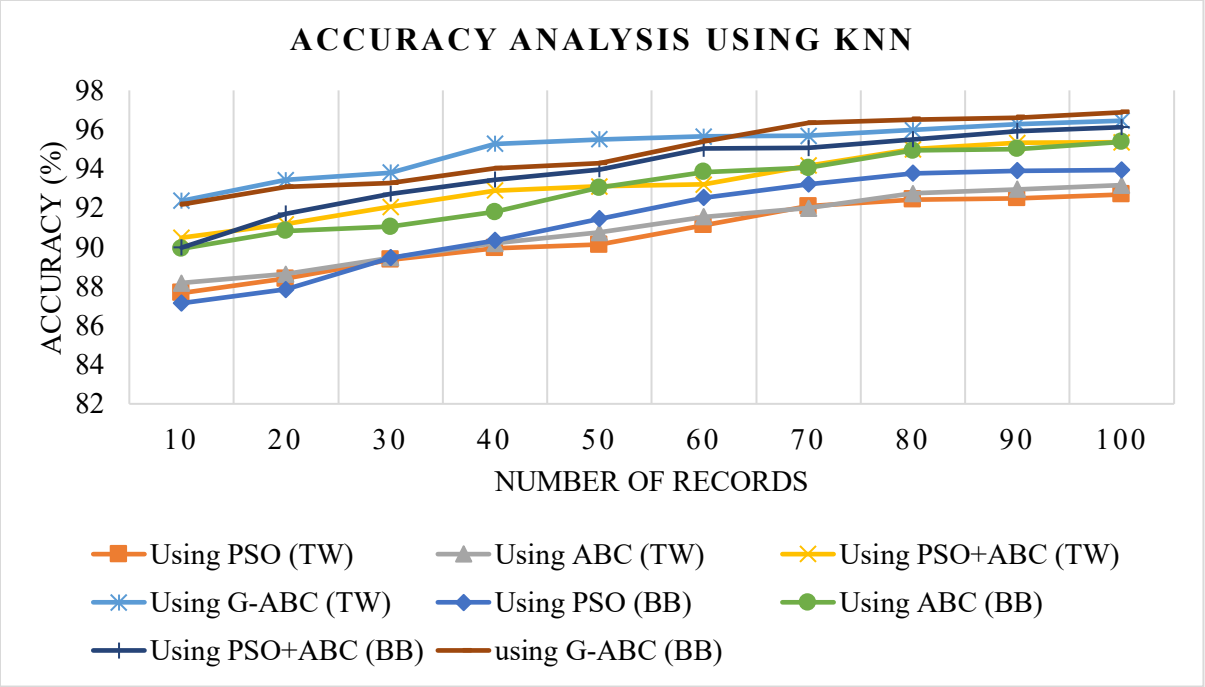

Figure 6.16 Accuracy Analysis using KNN

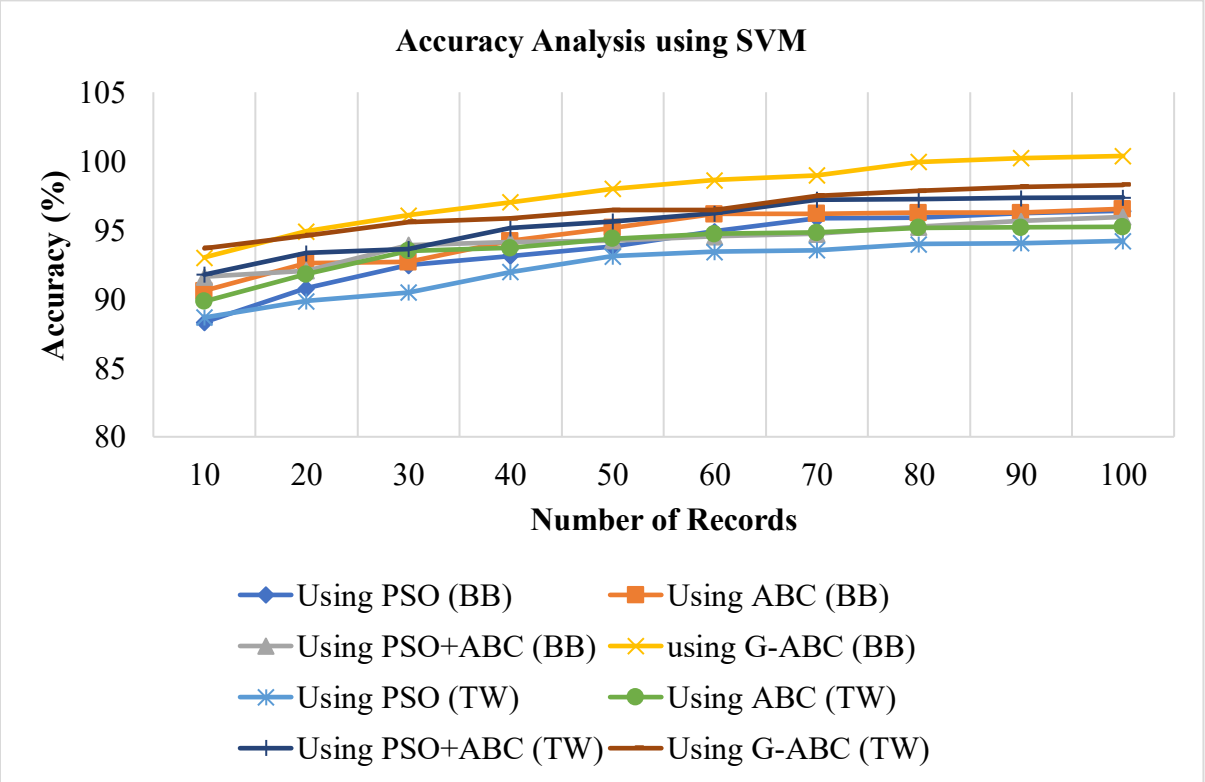

Figure 6.17 Accuracy Analysis using SVM

Figure 6.17 shows the accuracy analysis using the SVM technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for baseball dataset and least performance is shown using the PSO technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques.

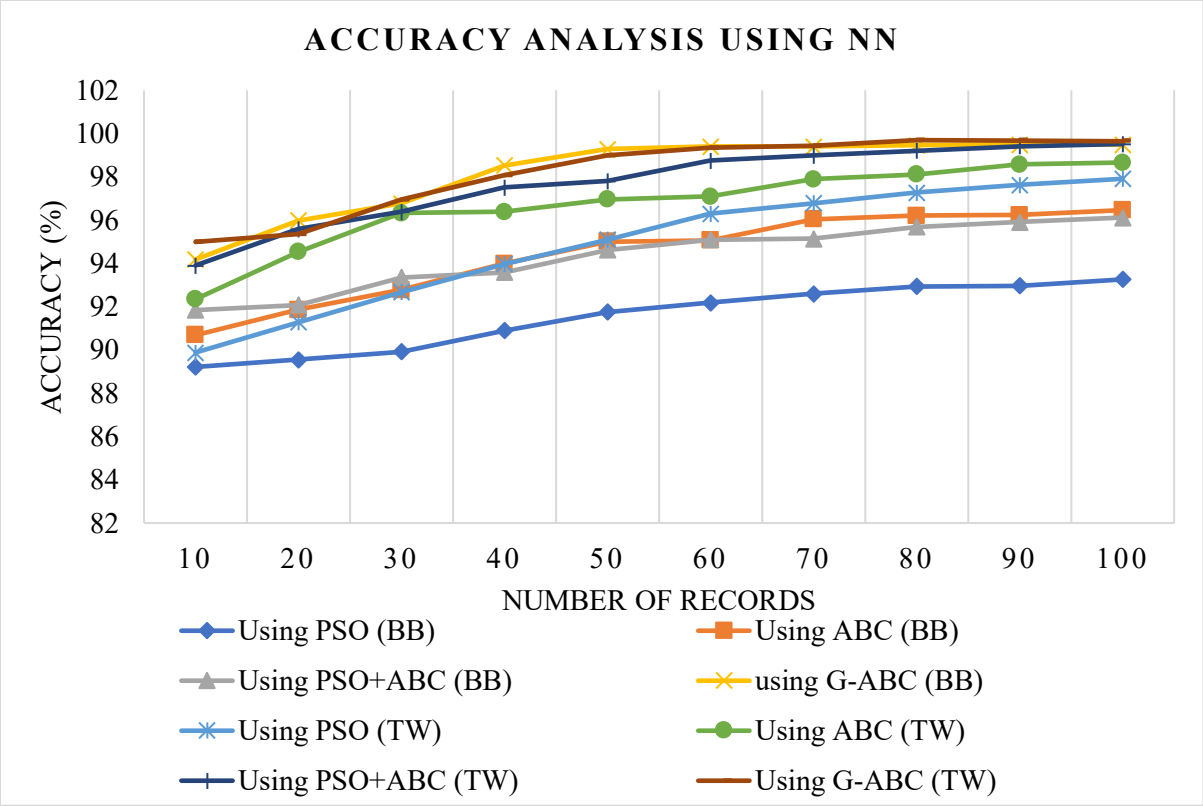

Figure 6.18 Accuracy Analysis using NN

Figure 6.18 shows the accuracy analysis using the NN technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for baseball dataset and least performance is shown using the ABC technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques.

- **Execution Time Analysis**

Execution time is defined as the time required completing the number of records in a given time using the baseball and twitter dataset. The analysis result are given using the different optimization techniques.

Table 6.11 Execution Time Analysis using Baseball Dataset

Using Naïve Bayes				
Number of Records	**Us ing PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	2.292358	2.246898	2.279432	2.21457
20	2.300933	2.25533	2.297398	2.220316
30	2.346113	2.295154	2.303323	2.238494
40	2.407314	2.314299	2.333839	2.291482
50	2.416418	2.336898	2.357115	2.333012
60	2.522625	2.400817	2.465025	2.347426
70	2.662125	2.534203	2.468958	2.363498
80	2.704805	2.648281	2.55014	2.380352
90	2.755864	2.709386	2.650774	2.384724
100	2.892204	2.858939	2.651019	2.534153
Using KNN				
10	2.395477	2.372946	2.373744	2.34185
20	2.410683	2.382473	2.390567	2.367364
30	2.426147	2.391967	2.408667	2.387851
40	2.490227	2.429193	2.481918	2.40712
50	2.579595	2.477738	2.54119	2.444474
60	2.628884	2.591696	2.675182	2.510387
70	2.674946	2.646444	2.797962	2.524027
80	2.749457	2.777772	2.890806	2.543901
90	2.877425	2.876535	3.035772	2.689994
100	3.050005	3.054986	3.046547	2.7253

Using SVM				
10	2.46989	2.430089	2.447437	2.41287
20	2.470268	2.457614	2.471227	2.424815
30	2.484969	2.510297	2.499457	2.451329
40	2.545563	2.574783	2.546921	2.515295
50	2.604938	2.587246	2.561239	2.553063
60	2.634102	2.651733	2.61532	2.575145
70	2.678892	2.722783	2.707608	2.632219
80	2.792686	2.812573	2.725683	2.782704
90	3.008334	2.82304	2.753357	2.862206
100	3.029831	2.919239	2.922248	3.081483
Using NN				
10	2.732805	2.663629	2.694157	2.647458
20	2.760385	2.663944	2.717758	2.66732
30	2.790805	2.721657	2.746976	2.700914
40	2.875261	2.793591	2.777416	2.719035
50	2.925319	2.818424	2.843451	2.748748
60	3.044931	2.917346	2.856155	2.837018
70	3.142673	3.028845	2.911184	2.983382
80	3.311608	3.210732	2.940032	3.063632
90	3.44838	3.370517	2.972843	3.124292
100	3.4599	3.504811	3.083374	3.204676

Table 6.11 shows that the execution time computed using the NB, KNN, SVM, and NN classifier for baseball dataset. The results showed that average value using the NB classifier, 2.33s for G-ABC, 2.4s for ABC and PSO+ABC respectively. Furthermore, execution time using the G-ABC

technique for NN and SVM classifier is 2.8s and 2.6s respectively comparatively less than other techniques. The performance of the G-ABC technique is superior in comparison to other optimization techniques.

Table 6.12 Execution Time Analysis using Twitter Dataset

Using Naïve Bayes				
Number of Records	**Using PSO**	**Using ABC**	**Using PSO+ABC**	**Using G-ABC**
10	2.28737	2.251295	2.256893	2.21457
20	2.307754	2.263198	2.280646	2.229202
30	2.325366	2.290894	2.324279	2.270479
40	2.384818	2.36616	2.382061	2.330796
50	2.416631	2.382073	2.472127	2.409447
60	2.526032	2.462218	2.47902	2.420502
70	2.639074	2.53205	2.518797	2.531748
80	2.736885	2.541215	2.629557	2.56811
90	2.848233	2.665368	2.675701	2.575121
100	2.925599	2.830034	2.731973	2.671082
Using KNN				
10	2.424445	2.350887	2.398132	2.33241
20	2.440015	2.364919	2.414471	2.361923
30	2.479812	2.368864	2.445515	2.414407
40	2.564486	2.370627	2.52669	2.45524
50	2.591052	2.454576	2.530344	2.514502
60	2.616172	2.541276	2.54093	2.582963
70	2.667796	2.626612	2.653786	2.59852

80	2.710885	2.649121	2.695688	2.691695
90	2.84755	2.687968	2.753179	2.843817
100	2.951421	2.884363	2.773363	2.959328
Using SVM				
10	2.479106	2.438906	2.476207	2.41347
20	2.494944	2.45357	2.487478	2.424846
30	2.543404	2.482742	2.507213	2.445142
40	2.626874	2.55053	2.545028	2.498703
50	2.720898	2.645316	2.634254	2.527234
60	2.838444	2.67526	2.745666	2.661643
70	2.983362	2.797693	2.752208	2.76841
80	3.158942	2.912228	2.776764	2.829276
90	3.37396	2.949632	2.930565	2.896688
100	3.398968	2.998519	3.135796	3.019422
Using NN				
10	2.739749	2.670072	2.707701	2.63123
20	2.769312	2.672741	2.709284	2.67709
30	2.81731	2.680801	2.758437	2.690325
40	2.826424	2.76211	2.78895	2.77718
50	2.900623	2.853954	2.811035	2.81411
60	2.998325	2.962791	2.927425	2.883825
70	3.059274	3.053529	2.928769	2.955529
80	3.238971	3.11506	3.010654	3.007262
90	3.358891	3.176902	3.147441	3.104758
100	3.588911	3.393799	3.364217	3.109473

Table 6.12 shows that the execution time computed using the NB, KNN, SVM, and NN classifier for twitter dataset. The results showed that average value using the NB classifier, about 2.5s for PSO, ABC and PSO+ABC respectively. Furthermore, execution time using the G-ABC technique for NN and SVM classifier is 2.86s and 2.64s respectively which is comparatively less than other techniques. The performance of the G-ABC technique is superior in comparison to other optimization techniques.

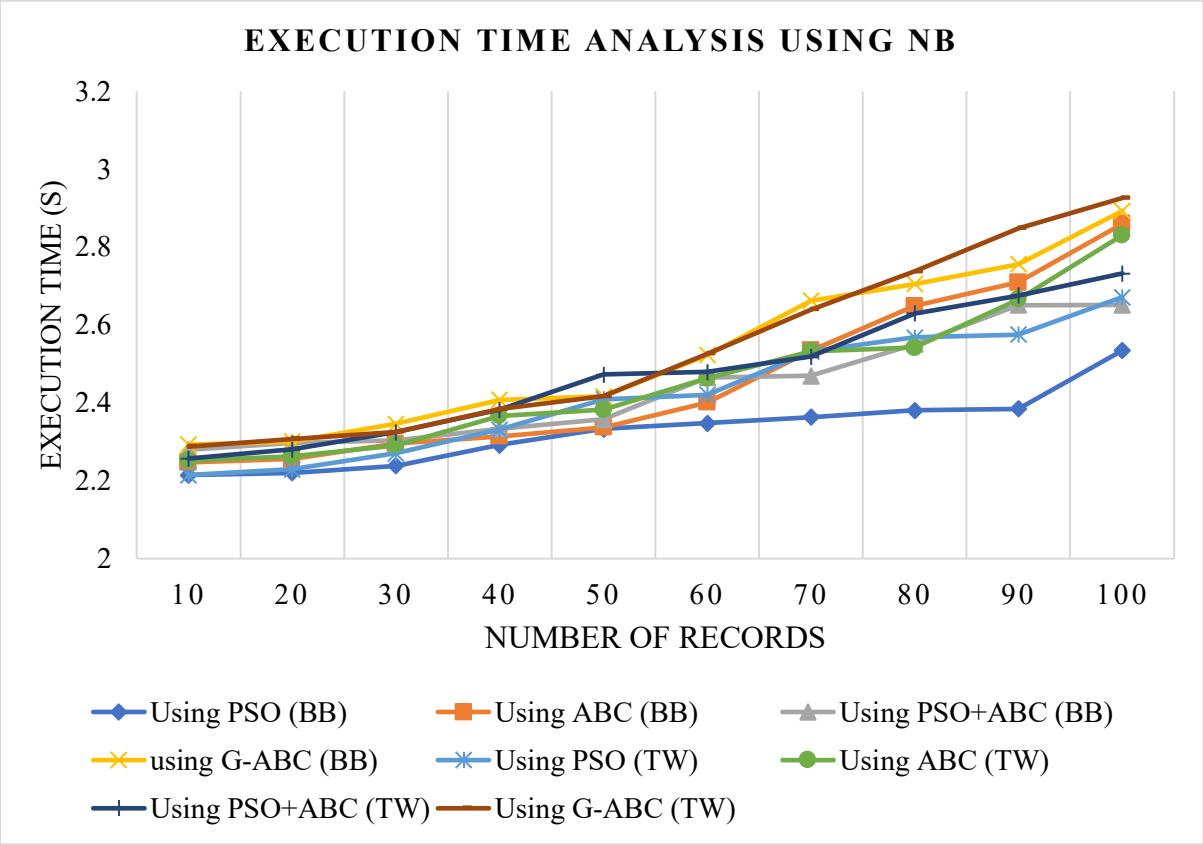

Figure 6.19 Execution time Analysis using NB

Figure 6.19 shows the execution time analysis using the NB technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for baseball and twitter dataset and least performance is shown using the PSO technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques.

Figure 6.20 shows the execution time analysis using the KNN technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for baseball and twitter dataset and least performance is shown using the PSO technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques.

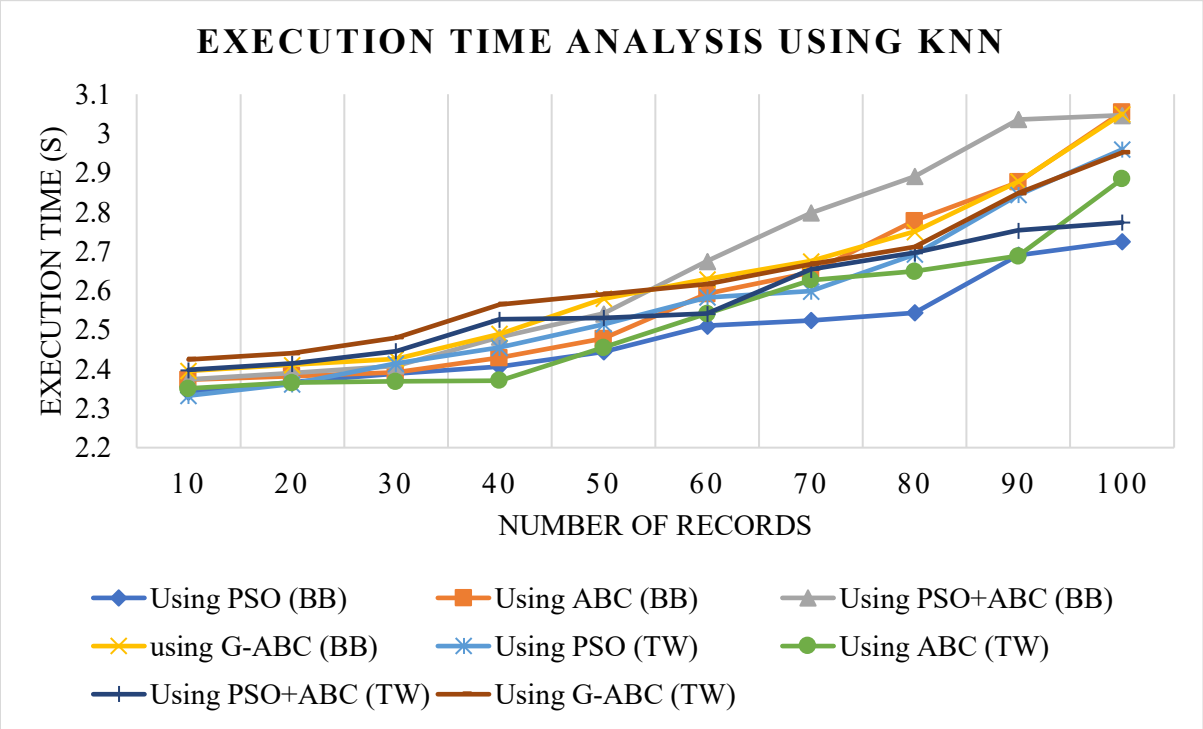

Figure 6.20 Execution time Analysis using KNN

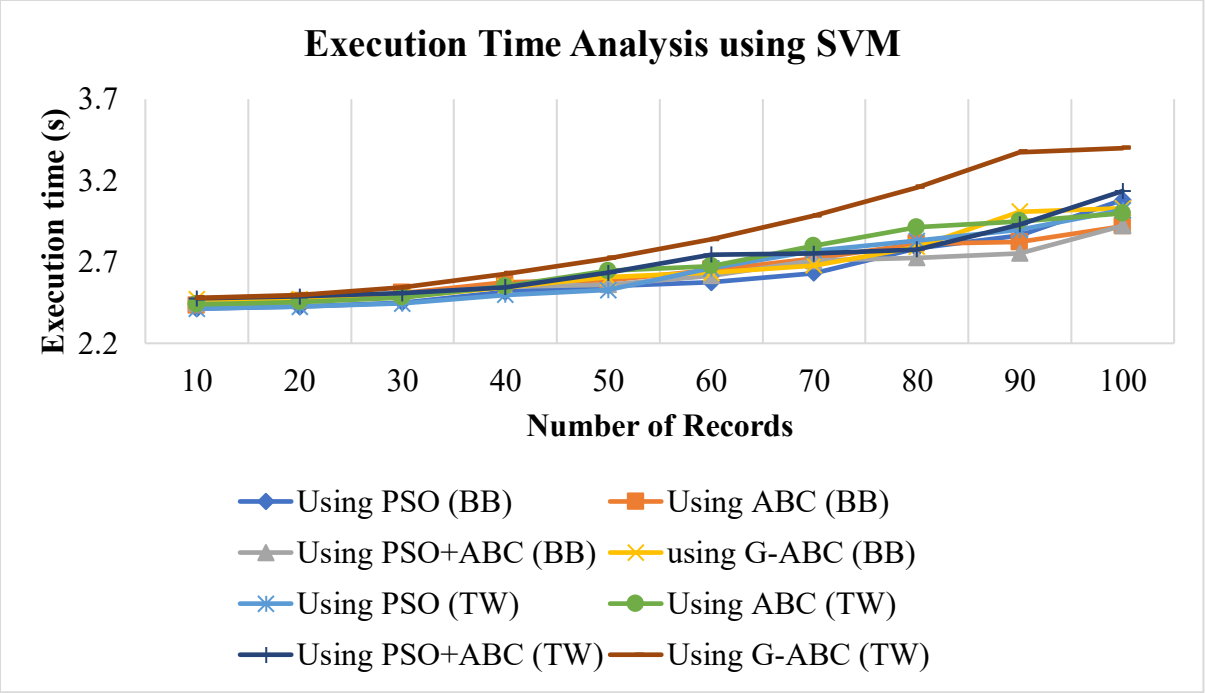

Figure 6.21 Execution time Analysis using SVM

Figure 6.21 shows the execution time analysis using the SVM technique for twitter and baseball dataset. The better performance is shown using the G-ABC technique for baseball and twitter dataset and least performance is shown using the PSO technique for twitter dataset. Thus, G-ABC technique performs better in comparison to other techniques.

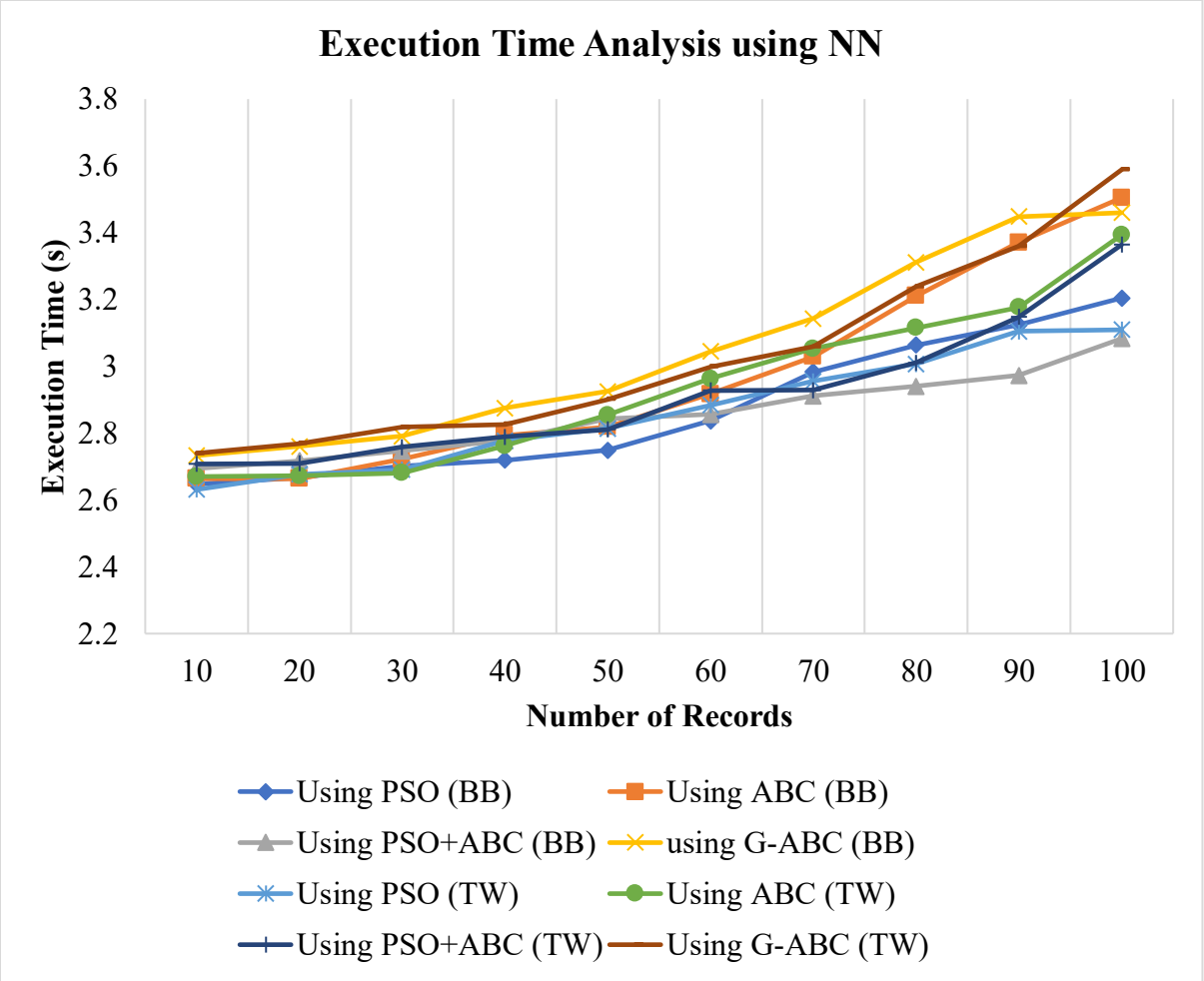

Figure 6.22 Execution time Analysis using NN

Figure 6.22 shows the execution time analysis using the NN classifier for different optimization techniques using the baseball and twitter dataset. The execution time analysis result shown that least performance is shown by using the ABC technique for baseball dataset and better performance is shown using the G-ABC technique for both datasets.

- **Overall Evaluation**

At the classification level, the proposed work involved four classifiers namely, NB, KNN, SVM and NN. The comparative analysis done for to justify the best performance of G-ABC with associated rule mining, mean variance optimization and NN based classification architecture is summarized below.

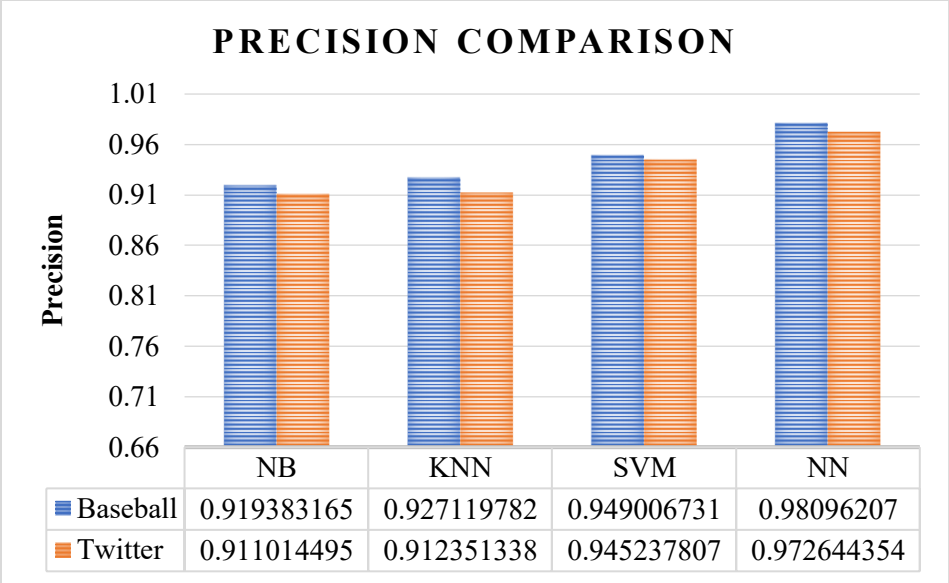

Figure 6.23 Precision Comparative Analysis for G-ABC and NN

Figure 6.23 shows the comparison of precision for G-ABC analysis using the NB, KNN, SVM, and NN for baseball and twitter dataset. The analysis result shown that least performance is shown by KNN for twitter dataset followed by the NB classifier for baseball dataset and better performance is shown by NN for both datasets.

Figure 6.24 shows the comparison of sensitivity for G-ABC analysis using the NB, KNN, SVM, and NN for baseball and twitter dataset. The analysis result shown that least sensitivity is shown by KNN for both datasets followed by the NB classifier and SVM classifier for both dataset. The better performance is shown by NN for both datasets. Thus, NN classifier provides better results with G-ABC for rule mining.

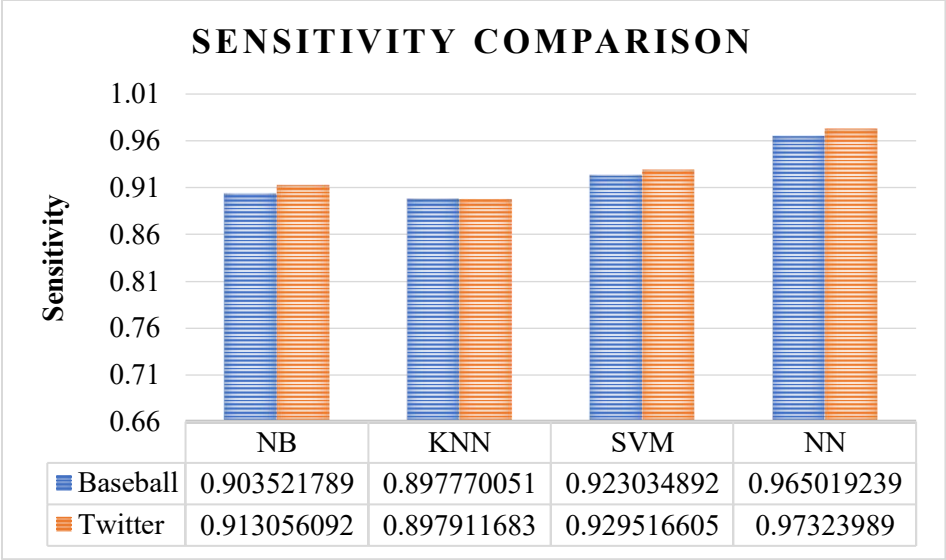

Figure 6.24 Sensitivity Comparative Analysis for G-ABC and NN

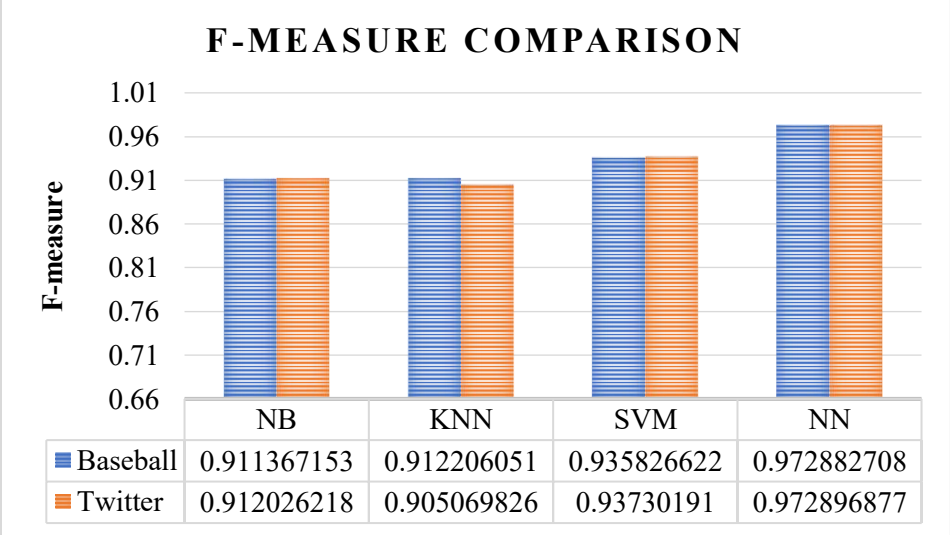

Figure 6.25 F-measure Comparative Analysis for G-ABC and NN

Figure 6.25 shows the F-measure comparison for G-ABC analysis using the NB, KNN, SVM, and NN for baseball and twitter dataset. The analysis result shown that least F-measure is shown by KNN followed by the NB classifier and SVM classifier for both dataset. The better performance is shown by NN for both datasets. Thus, NN classifier provides better results with G-ABC for rule mining.

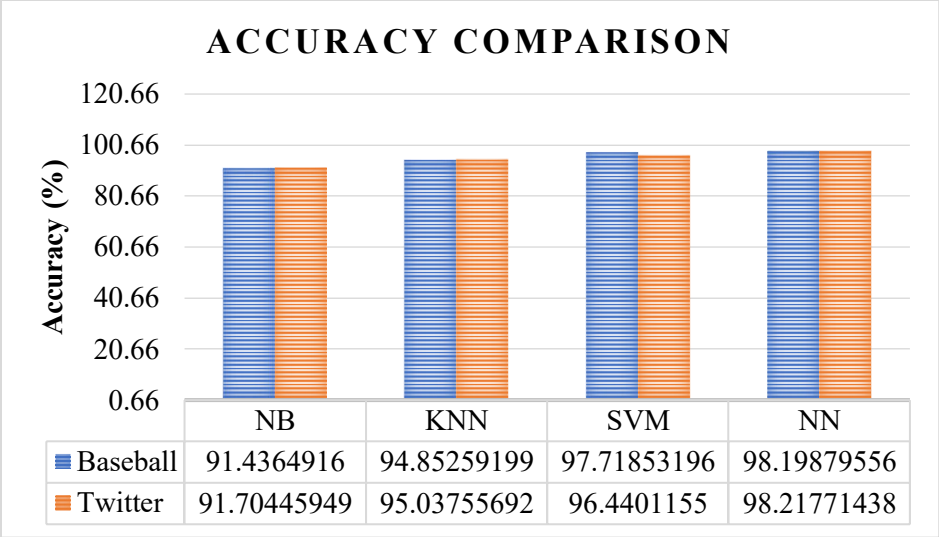

Figure 6.26 Accuracy Comparative Analysis for G-ABC and NN

Figure 6.26 shows the accuracy comparison for G-ABC analysis using the NB, KNN, SVM, and NN for baseball and twitter dataset. The analysis result shown that least accuracy is shown by NB classifier followed by the KNN classifier and SVM classifier for both dataset. The better performance is shown by NN for both datasets. Thus, NN classifier provides better results with G-ABC for rule mining.

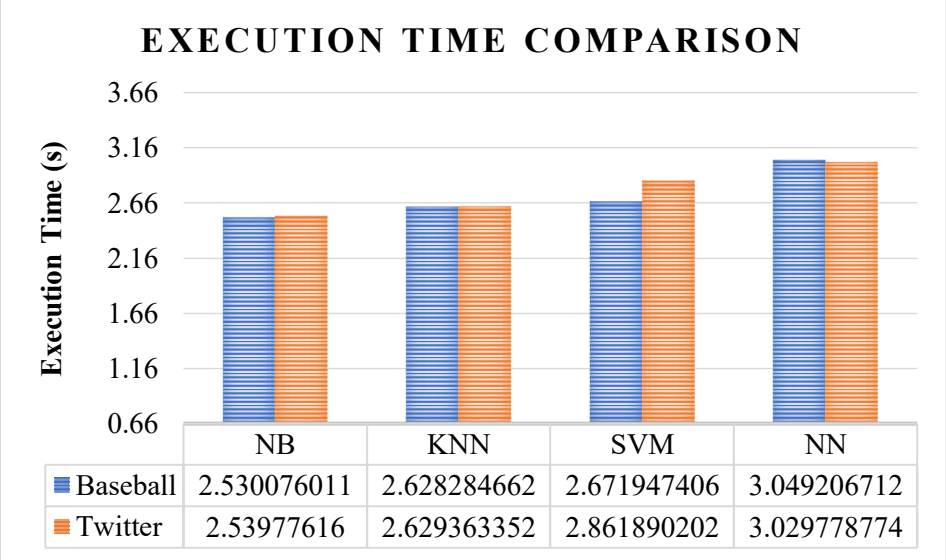

Figure 6.27 Execution Time Comparative Analysis for G-ABC and NN

Figure 6.27 shows the execution time comparison for G-ABC analysis using the NB, KNN, SVM, and NN for baseball and twitter dataset. The analysis result shown that least performance is shown by NB followed by the KNN classifier and SVM classifier for both dataset. The better performance

is shown by NN for both datasets. Thus, NN classifier provides better results with G-ABC for rule mining.

6.4 Evaluation using Multiple Simulations

The proposed work has also been evaluated for various number of simulations. To do so, the other ordinal variables that includes total number of data samples have been kept constant. The evaluation of the results has been made on all evaluated parameters that has been used for the comparison that has been made earlier.

Table 6.13 illustrates the results for increasing number of simulations. The total simulation count is 1000 in case of proposed work scenario. The proposed simulation scenario has been kept in such a manner that the assembly of algorithms has been executed for both the datasets namely the baseball and the twitter set. The minimum number of simulations for this analysis has been set to be 200 while maximum simulation performed in each case are 1000. The performance is analysed for precision, recall, f-measure, accuracy and the execution time reflected for variation in the simulations.

Table 6.13 Precision Analysis for Multiple Simulations

Total number	Baseball G-ABC+NN 5-10 L	Twitter G-ABC+N	Twitter NN only	Baseball NN only	Twitter G-ABC+N	Baseball G-ABC+N
200	0.9661162	0.95789474	0.95918367	0.95918367	0.95789474	0.95652174
300	0.99836285	0.98876404	0.98837209	0.98913043	0.98863636	0.98837209
400	0.95917806	0.96555556	0.95348837	0.95555556	0.95789474	0.95698925
500	0.98130169	0.97894737	0.97826087	0.97701149	0.97849462	0.97938144
600	0.97825673	0.96590909	0.96907216	0.96907216	0.96842105	0.96774194
700	0.97753105	0.97741935	0.96590909	0.96511628	0.96551724	0.96774194
800	0.96397202	0.95789474	0.95652174	0.95604396	0.95789474	0.95876289
900	0.97375363	0.96470588	0.96808511	0.96590909	0.96808511	0.9673913

1000	0.97323559	0.97629213	0.96629213	0.96703297	0.96511628	0.96629213

The performance observed concerning the number of simulations are found to be random for all the cases. For instance, for 500 simulations, the precision values observed using Baseball G-ABC+NN 5-10 L is 0.981301691, Twitter G-ABC+NN 5-10 L is 0.978947368, Twitter NN only is 0.97826087, Baseball NN only is 0.977011494, Twitter G-ABC+NN 15-30 L is 0.978494624, and Baseball G-ABC+NN 15-30 L is 0.979381443. The average precision analysis depicted in figure 6.28 shows that overall, an average precision observed using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.9746, 0.9704, 0.9672, 0.9671, 0.9676, and 0.9677, respectively. Thus, multiple simulation analysis performed for precision analysis shows that both for Twitter and Baseball datasets, the proposed G-ABC+NN utilizing 5 to 10 neural layers outperformed the other scenarios.

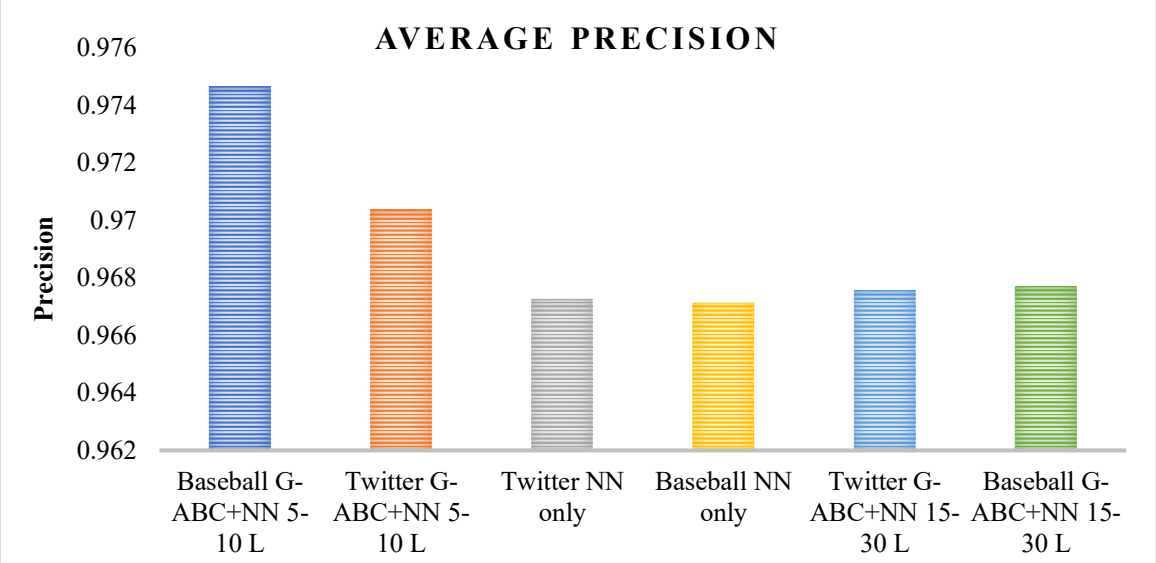

Figure 6.28 Precision Analysis for Multiple Simulations

The observations for the recall values observed for the analysis for multiple simulations and different scenarios is tabulated in Table 6.14. It is generalized that increasing the number of simulations from 200 to 1000 does not definitely increase the values of the performance parameter. However, the recall values observed against 500 simulations using Baseball G-ABC+NN 5-10 L is 0.989690722, Twitter G-ABC+NN 5-10 L is 0.989361702, Twitter NN only is 0.989010989, Baseball NN only is 0.977011494, Twitter G-ABC+NN 15-30 L is 0.947916667, and Baseball G-ABC+NN 15-30 L is 0.979381443.

Table 6.14 Recall Analysis for Multiple Simulations

Total number	Baseball G-ABC+N	Twitter G-ABC+N	Twitter NN only	Baseball NN only	Twitter G-ABC+N	Baseball G-ABC+N
200	0.96039604	0.957894737	0.959183673	0.979166667	0.989130435	0.977777778
300	0.958333333	0.956521739	0.95505618	0.919191919	0.887755102	0.913978495
400	1	1	1	0.924731183	0.892156863	0.927083333
500	0.989690722	0.989361702	0.989010989	0.977011494	0.947916667	0.979381443
600	0.989583333	0.988372093	0.989473684	0.912621359	1	0.909090909
700	0.989361702	0.989010989	0.988372093	0.954022989	0.903225806	0.957446809
800	0.96039604	0.957894737	0.956521739	0.97752809	0.947916667	0.978947368
900	0.989361702	0.987951807	0.989130435	0.87628866	0.968085106	0.881188119
1000	0.989361702	0.988505747	0.988505747	0.916666667	0.912087912	0.914893617

With further increase in the number of simulations variable performance values are obtained. Overall, average recall for 1000 simulations shown in figure 6.29 using Baseball G-ABC+NN 5-10 L**,** Twitter G-ABC+NN 5-10 L**,** Twitter NN only**,** Baseball NN only**,** Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.9807, 0.9795, 0.9795, 0.9375, 0.9387, and 0.9378, respectively. Here, higher recall is observed for the proposed G-ABC+NN based 5 to 10 neural layers for both the datasets under study. Thus, utilizing 5 to 10 neural layers performed better than even when increased number of neural layers are included in the simulation analysis.

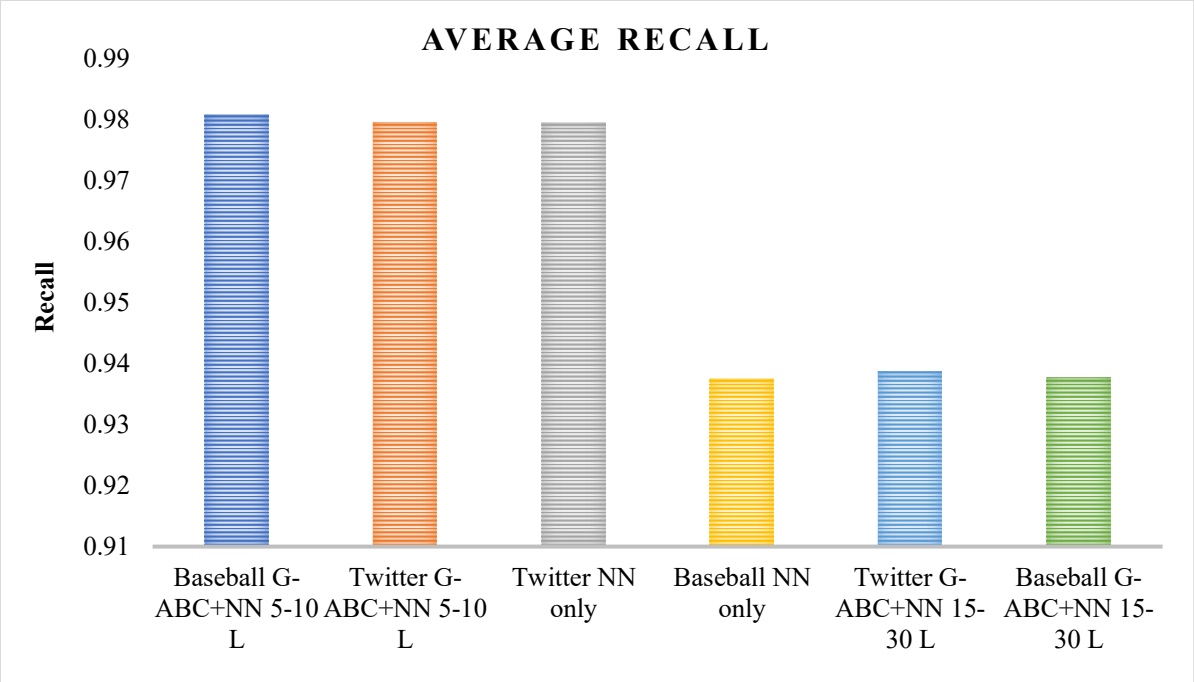

Figure 6.29 Recall Analysis for Multiple Simulations

F-measure analysis performed on various scenarios and multiple simulations ranging from 200 to 1000 is given in Table 6.15. The parametric values for the proposed G-ABC based rule mining architecture is analysed for variation in the number of neural layers used in the prediction analysis. Moving from 200 simulations to 1000 simulations, variable f-measure has been observed for each scenarios and the dataset under study. It is observed that f-measure values for G-ABC+NN utilizing 5 to 10 neural layers exhibited higher values for both datasets in comparison to other scenarios used in the study. For instance, for 500 simulations, f-measure using Baseball G-ABC+NN 5-10 L is 0.985478353, Twitter G-ABC+NN 5-10 L is 0.984126984, Twitter NN only is 0.983606557, Baseball NN only is 0.977011494, Twitter G-ABC+NN 15-30 L is 0.962962963, and Baseball G-ABC+NN 15-30 L is 0.979381443.

Table 6.15 F-measure Analysis for Multiple Simulations

Total number	Baseball G-ABC+NN 5-10 L	Twitter G-ABC+NN	Twitter NN only	Baseball NN only	Twitter G-ABC+NN	Baseball G-ABC+NN
200	0.963247626	0.957894737	0.959183673	0.969072165	0.973262032	0.967032967

300	0.977938634	0.972375691	0.971428571	0.952879581	0.935483871	0.94972067
400	0.979163743	0.977272727	0.976190476	0.93989071	0.923857868	0.941798942
500	0.985478353	0.984126984	0.983606557	0.977011494	0.962962963	0.979381443
600	0.983887435	0.977011494	0.979166667	0.94	0.983957219	0.9375
700	0.983410797	0.97826087	0.977011494	0.959537572	0.933333333	0.962566845
800	0.962180706	0.957894737	0.956521739	0.966666667	0.952879581	0.96875
900	0.981495617	0.976190476	0.978494624	0.918918919	0.968085106	0.922279793
1000	0.981232393	0.977272727	0.977272727	0.941176471	0.937853107	0.93989071

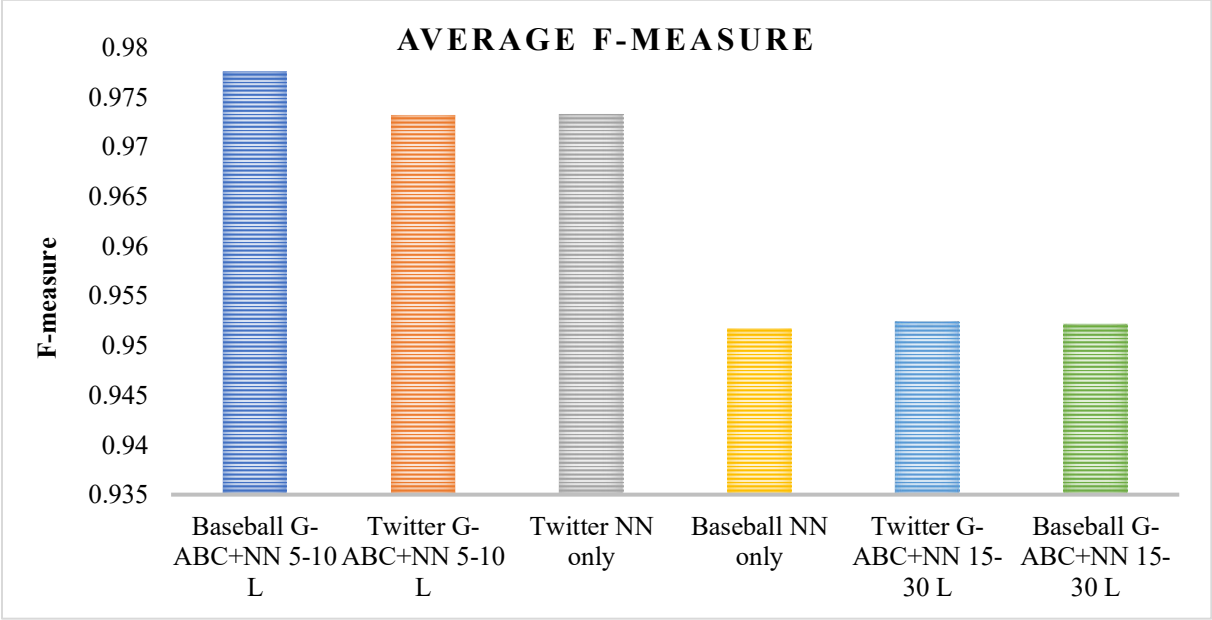

Figure 6.30 F-measure Analysis for Multiple Simulations

The graphical analysis of the average values of f-measure analysis is shown in figure 6.30. The graph shows that the average f-measure observed using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.9776, 0.9731, 0.9732, 0.9517, 0.9524, and 0.9521, respectively. This shows that the proposed prediction analysis performed using 5 to 10 neural layers outperformed the only NN and NN with 15 to 30 neural layers while utilizing rule mining architecture.

The observed accuracy values for the simulation rounds varied from 200 to 1000 is tabulated in Table 6.16 for different scenarios used for the multiple simulation analysis. In addition to only NN two scenarios for the proposed work by varying the number of neural layers are also presented namely, using 5 to 10 neural layers and 15 to 30 neural layers. The variation observed in the accuracy values varies with number of simulations performed. For instance, for 500 simulations accuracy observed using Baseball G-ABC+NN 5-10 L is 94.117%, Twitter G-ABC+NN 5-10 L is 93.939%, Twitter NN only is 93.75%, Baseball NN only is 92.391%, Twitter G-ABC+NN 15-30 L is 90.099%, and Baseball G-ABC+NN 15-30 L is 93.137%. Thus, for 500 simulations G-ABC+NN having 5 to 10 neural layers outperformed the other scenarios for both datasets. This shows that the proposed rule mining mechanism have significantly improved the overall performance when 5 to 10 neural layers are involved in the analysis.

Table 6.16 Accuracy Analysis for Multiple Simulations

Total number	Baseball G-ABC+NN 5-10 L	Twitter G-ABC+N	Twitter NN only	Baseball NN only	Twitter G-ABC+N	Baseball G-ABC+N
200	89.8148148	89.2156863	89.5238095	91.2621359	91.9191919	90.7216495
300	92	91.6666667	91.3978495	88.3495146	85.2941176	87.628866
400	93	92.4731183	92.1348315	86	83.4862385	86.407767
500	94.1176471	93.9393939	93.75	92.3913043	90.0990099	93.1372549
600	93.1372549	92.3913043	93.0693069	86.2385321	93.877551	85.7142857
700	93	92.7835052	92.3913043	89.2473118	84.8484848	90
800	89.8148148	89.2156863	88.8888889	90.625	88.3495146	91.1764706
900	93	92.1348315	92.8571429	82.5242718	91	83.1775701
1000	93	92.4731183	92.4731183	86.2745098	85.5670103	86

The graphical illustration of the average value of the accuracy obtained from the table is given in figure 6.31. The graph shows that the proposed G-ABC+NN outperform the only NN. Further, it

is observed that G-ABC+NN using 5 to 10 neural layers outperformed the F-ABC+NN using 15 to 30 neural layers. This, observation remain the same when different datasets namely, Twitter or Baseball are included in the simulation study. Overall, an average accuracy using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 92.321%, 91.810%, 91.832%, 88.101%, 88.271%, and 88.218%, respectively.

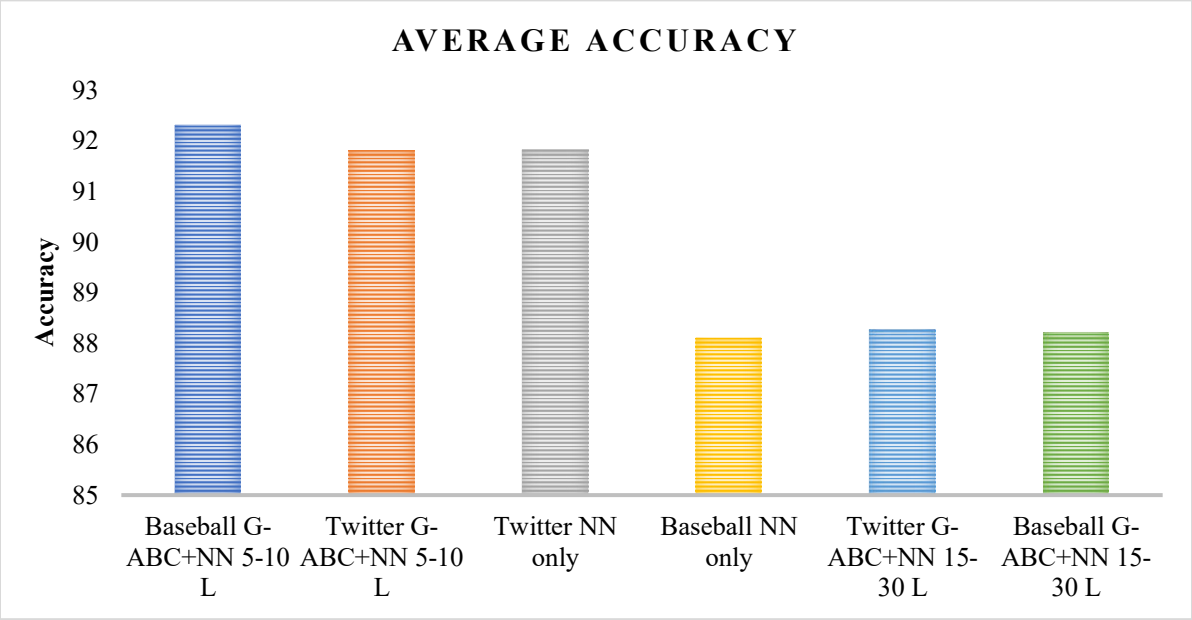

Figure 6.31 Accuracy Analysis for Multiple Simulations

The execution time analysis for multiple simulations performed using both the datasets for different scenarios are summarized in Table 6.17. The table depicts the execution time with respect to change in the number of simulations that are varied from 200 to 1000. The table generalizes that with increase in the number of simulations more time is required to execute the work. However, against 500 simulations accuracy using Baseball G-ABC+NN 5-10 L is 0.265615461s, Twitter G-ABC+NN 5-10 L is 0.266974219s, Twitter NN only is 0.269741194s, Baseball NN only is 0.27779926s, Twitter G-ABC+NN 15-30 L is 0.287035971s, and Baseball G-ABC+NN 15-30 L is 0.294714825s. Thus, it is observed that when more neural layers are involved in the simulation analysis, the execution time also increases. Therefore, more execution time is required for simulation of 15 to 30 neural layered architecture.

Table 6.17 Execution Time Analysis for Multiple Simulations

Total number	Baseball G-ABC+NN 5-10 L	Twitter G-ABC+NN 5-10 L	Twitter NN only	Baseball NN only	Twitter G-ABC+NN 15-30 L	Baseball G-ABC+NN 15-30 L
200	0.25166846	0.25465502	0.25914103	0.26909468	0.25748965	0.26089434
300	0.26343533	0.27928526	0.27153885	0.26557552	0.26387801	0.27880309
400	0.28208645	0.31361309	0.28674148	0.29394332	0.30751303	0.30927762
500	0.26561546	0.26697422	0.26974119	0.27779926	0.28703597	0.29471483
600	0.30303975	0.33071854	0.32593393	0.33067934	0.33543912	0.32984335
700	0.25242489	0.25899495	0.2617714	0.26163001	0.26946471	0.25479206
800	0.27606631	0.27992782	0.29103642	0.28652881	0.29978858	0.30090451
900	0.29375484	0.31539777	0.31091253	0.31451973	0.30153375	0.31330152
1000	0.28717846	0.32131588	0.31360088	0.30186829	0.30558813	0.31381439

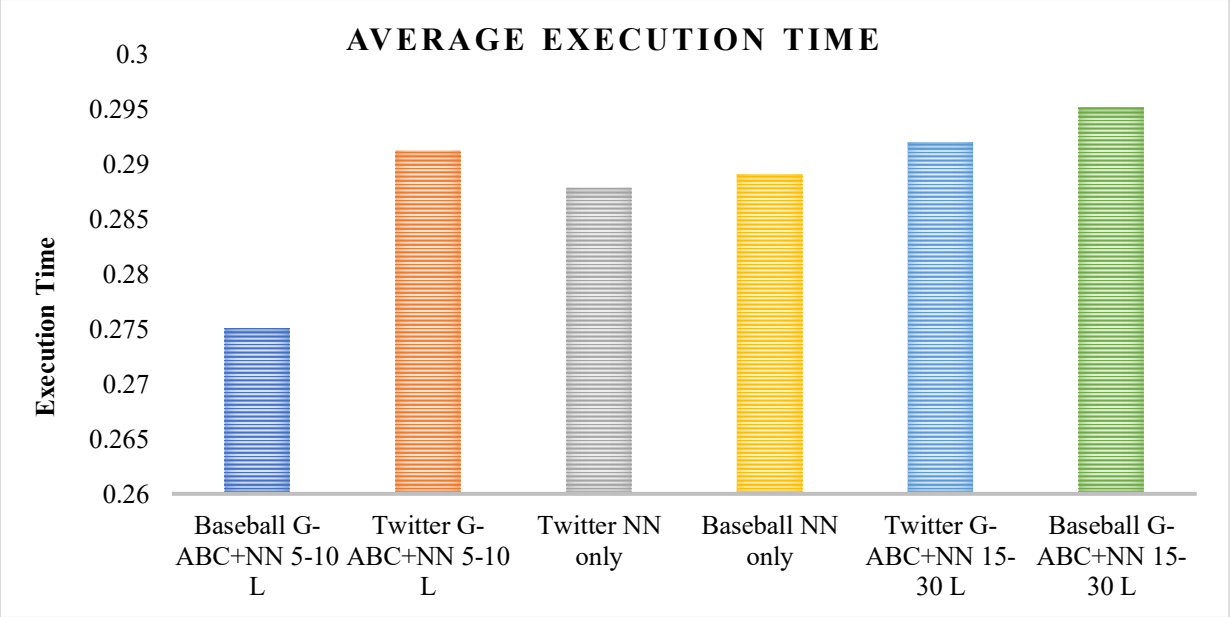

Figure 6.33 Execution Time Analysis for Multiple Simulations

The average execution time using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.2750s, 0.2912s, 0.2878s, 0.2891s, 0.2920s, and 0.2951s, respectively. The graphical illustration of the observed average execution time for each study is shown in figure 6.32. The graph shows that an increase in the number of neural layers in the proposed G-ABC+NN considerably increases the overall execution time. Therefore, the number of neural layers is restricted to 5 to 10 so as to lower the overall execution time without compromising the performance of the system.

6.5 Evaluation with more data sample

The results present performance metrics for several machine learning models across varying dataset sizes. We observe that the "Proposed G-ABC + Neural" model consistently achieves high precision, with values ranging from approximately 0.947 to 0.981 as the dataset size increases from 1000 to 10000 samples.

In comparison, the "PSO + Neural" model achieves lower precision scores, ranging from about 0.944 to 0.968, while the "ABC + Neural" model also exhibits lower precision, varying from approximately 0.946 to 0.969. However, when we consider the model that combines PSO, ABC, and Neural components, its precision ranges from roughly 0.938 to 0.955, which is lower than that of the "Proposed G-ABC + Neural" model.

Moving on to recall, the "Proposed G-ABC + Neural" model maintains consistently high values, ranging from approximately 0.990 to 0.986 across different dataset sizes. In contrast, the "PSO + Neural" model achieves recall scores ranging from about 0.988 to 0.986, while the "ABC + Neural" model exhibits recall values varying from approximately 0.985 to 0.980. The model combining PSO, ABC, and Neural components achieves recall scores ranging from roughly 0.978 to 0.967, which are lower than those of the "Proposed G-ABC + Neural" model.

Focusing on the F-measure, which balances precision and recall, the "Proposed G-ABC + Neural" model consistently achieves high scores, ranging from approximately 0.985 to 0.978 as the dataset size increases. The "PSO + Neural" model attains lower F-measure values, ranging from about 0.967 to 0.979, and the "ABC + Neural" model exhibits F-measure scores varying from approximately 0.970 to 0.978. The model combining PSO, ABC, and Neural components achieves

F-measure scores ranging from roughly 0.966 to 0.978, which are consistently lower than those of the "Proposed G-ABC + Neural" model.

Lastly, considering accuracy, the "Proposed G-ABC + Neural" model consistently maintains accuracy scores above 93%, ranging from approximately 93.7% to 93.033% across different dataset sizes. This indicates the model's ability to make correct predictions. In contrast, other models, such as "PSO + Neural," "ABC + Neural," and the combined model, achieve lower accuracy scores, further emphasizing the superior performance of the "Proposed G-ABC + Neural" model in accurately classifying data points.

- **Precision**

Table 6.18 Precision Analysis using Twitter Dataset

Number of Records	Using PSO	Using ABC	Using PSO+ABC	Using G-ABC
1000	0.988217968	0.990990991	0.990291262	0.992876191
2000	0.96799117	0.96810934	0.96061885	0.98063925
3000	0.96319499	0.95169531	0.96462899	0.97062982
4000	0.94747407	0.95284502	0.95481094	0.97206317
5000	0.94996228	0.95721217	0.95473351	0.97158873
6000	0.95342364	0.94887064	0.95328095	0.9669437
7000	0.95277373	0.95014451	0.9442694	0.96449105
8000	0.9475976	0.95158938	0.94925417	0.96633025
9000	0.94534535	0.9461865	0.93847823	0.9650119
10000	0.94892096	0.94917212	0.95504628	0.96944565
Average	**0.95649**	**0.95668**	**0.95654**	**0.972**

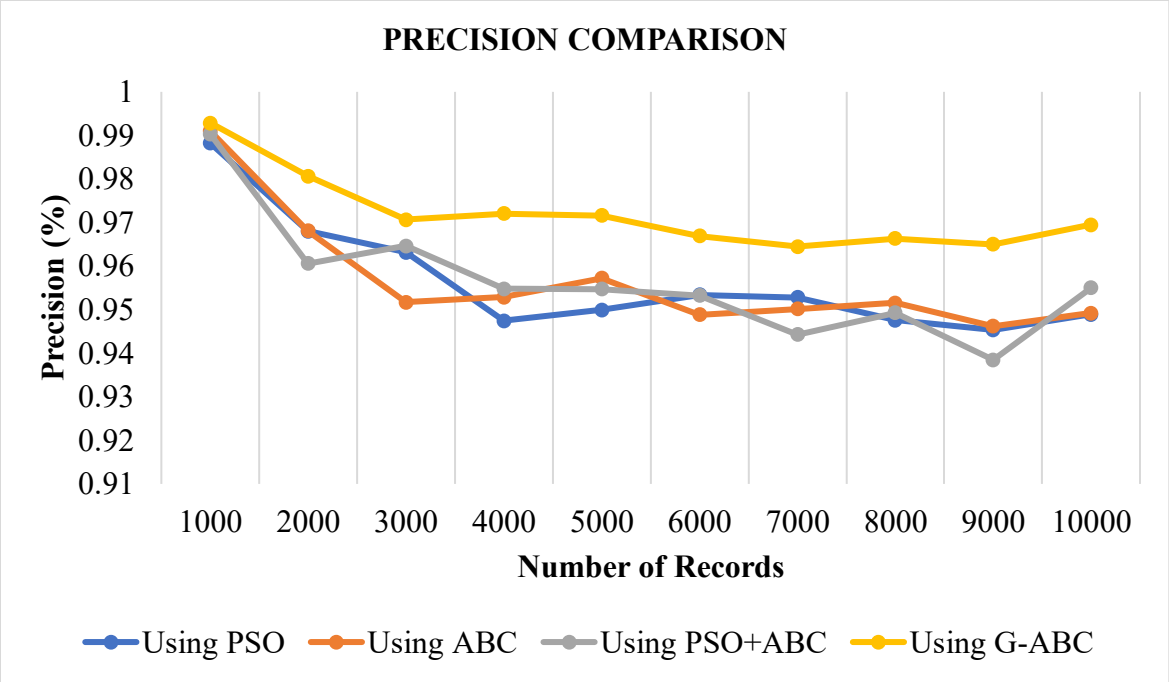

Figure 6.33 Precision Analysis using Twitter Dataset

The "Precision Proposed G-ABC +Neural" model has an average precision of approximately 0.970, which means it achieves a high level of accuracy in positive predictions on average across all dataset sizes. Comparatively, the "Precision PSO +Neural" model and the "Precision ABC + Neural" model have lower average precisions of around 0.953, indicating that, on average, they make somewhat less accurate positive predictions. The model combining PSO, ABC, and Neural components, represented by "Precision PSO + ABC +Neural," also has an average precision of approximately 0.953, similar to the "PSO +Neural" and "ABC + Neural" models.

- **Recall**

Table 6.19 Recall Analysis using Twitter Dataset

Number of Records	Using PSO	Using ABC	Using PSO + ABC	Using G-ABC
1000	0.8472222	0.8700565	0.711864	0.886812
2000	0.988726	0.9889471	0.91555	0.9904863
3000	0.9867629	0.9846228	0.87666	0.9886645

4000	0.9867596	0.9866537	0.778056	0.9898694
5000	0.9866806	0.9883853	0.81934	0.9899616
6000	0.9886473	0.9861289	0.914257	0.9894887
7000	0.9869325	0.9855724	0.888951	0.9889959
8000	0.9873279	0.9884883	0.855542	0.990095
9000	0.9863701	0.9871429	0.748209	0.9900024
10000	0.9861977	0.9866578	0.82325	0.989925
Average	**0.97316**	**0.97527**	**0.83317**	**0.97943**

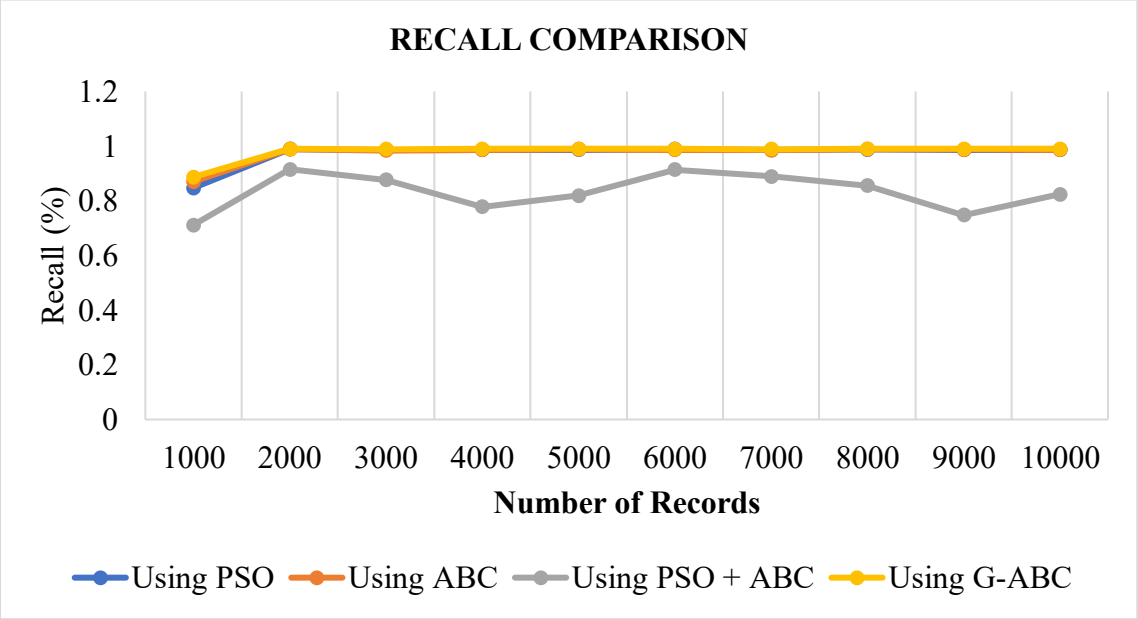

Figure 6.34 Recall Analysis using Twitter Dataset

The "Recall Proposed G-ABC +Neural" model has an average recall of approximately 0.990, indicating its strong ability to correctly identify positive cases on average across all dataset sizes. In contrast, the "Recall PSO +Neural," "Recall ABC + Neural," and "Recall PSO + ABC +Neural" models have lower average recalls of around 0.987, suggesting that, on average, they capture a slightly lower proportion of positive cases.

- **F-Measure**

Table 6.20 F-measure Analysis using Twitter Dataset

Number of Records	Using PSO	Using ABC	Using PSO + ABC	Using G-ABC
1000	0.9123046	0.9265945	0.828306	0.9368517
2000	0.9782488	0.9784173	0.937543	0.9855382
3000	0.9748365	0.9678791	0.918543	0.9795642
4000	0.9667179	0.9694547	0.857419	0.9808855
5000	0.9679734	0.972549	0.88187	0.9806891
6000	0.970716	0.9671411	0.933361	0.9780863
7000	0.9695523	0.9675343	0.915775	0.9765898
8000	0.9670549	0.969688	0.899965	0.9780683
9000	0.9654221	0.9662309	0.832612	0.9773474
10000	0.9672003	0.967552	0.884264	0.9795783
Average	**0.964**	**0.9653**	**0.88897**	**0.97532**

The "F-measure Proposed G-ABC +Neural" model exhibits an average F-measure of about 0.979, which indicates a balanced performance in terms of precision and recall on average across all dataset sizes. In comparison, the "F-measure PSO +Neural," "F-measure ABC + Neural," and "F-measure PSO + ABC +Neural" models have slightly lower average F-measures of around 0.970, implying a somewhat less balanced performance on average.

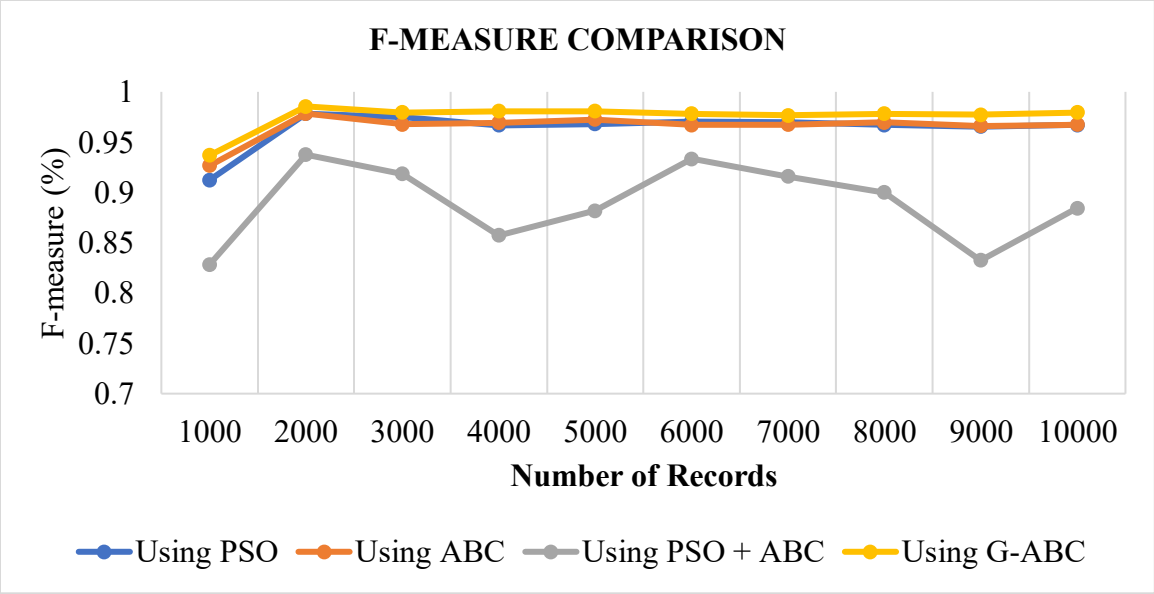

Figure 6.35 F-measure Analysis using Twitter Dataset

- **Accuracy**

Table 6.21 Accuracy Analysis using Twitter Dataset

Number of Records	Using PSO	Using ABC	Using PSO + ABC	Using G-ABC
1000	0.806490385	0.835140998	0.685220729	0.854
2000	0.926082365	0.922909881	0.849502488	0.937
3000	0.917910448	0.90026362	0.821814609	0.930333333
4000	0.898477157	0.905046282	0.729107981	0.92825
5000	0.90253225	0.915998312	0.764121543	0.927
6000	0.91131607	0.902186646	0.842213884	0.925666667
7000	0.909245254	0.901611244	0.811886217	0.924428571
8000	0.903507516	0.90942029	0.793036042	0.924625
9000	0.893176337	0.898686435	0.688886361	0.924222222

10000	0.901492175	0.900401802	0.77028348	0.9236
Average	**0.897023**	**0.8991666**	**0.7756073**	**0.9199126**

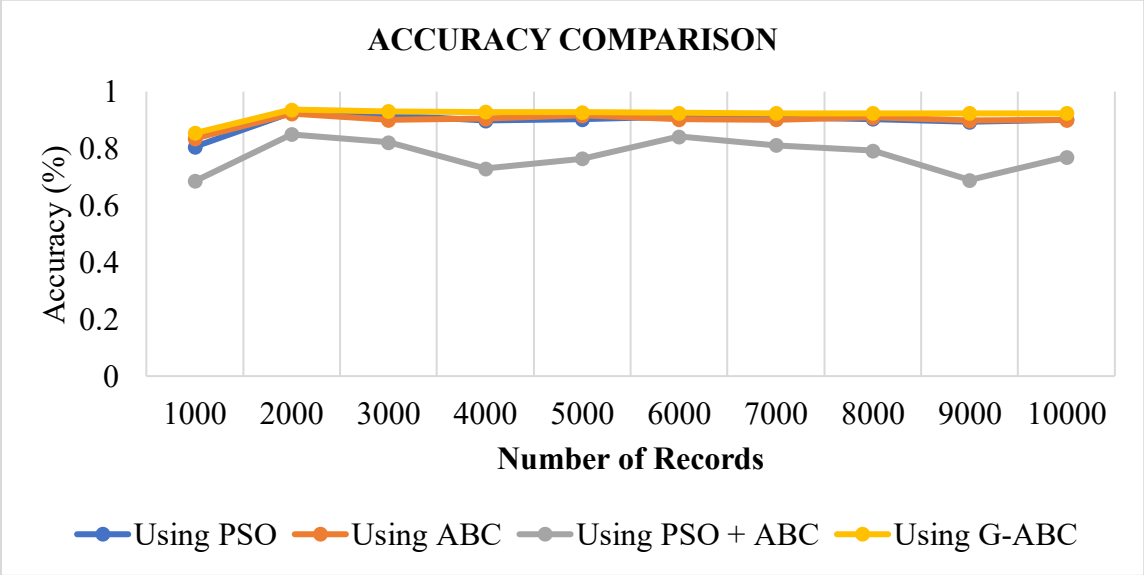

Figure 6.36 Accuracy Analysis using Twitter Dataset

The "Accuracy Proposed G-ABC + Neural" model boasts an average accuracy of approximately 92.72%, indicating its high average correctness in predictions across all dataset sizes. In contrast, the "Accuracy PSO +Neural," "Accuracy ABC + Neural," and "Accuracy PSO + ABC +Neural" models exhibit lower average accuracies, with values around 90.71%, 90.63%, and 78.57%, respectively. This highlights the superior average accuracy of the "Proposed G-ABC + Neural" model.

CHAPTER 7:
CONCLUSION AND FUTURE SCOPE

7.1 Conclusion

7.2 Future Scope

The research study has presented an advanced data mining framework based on G-ABC with associated rule mining and mean-variance optimization followed by neural network architecture. The objective of the study was to explore how Artificial Intelligence can be applied to rule mining while involving nature-inspired optimization approaches and rule-based data mining. The research comprises of twin evaluation that is performed at the feature extraction level as well as the classification level.

7.1 Conclusion

➢ Conclusion at Feature Selection

The different optimization techniques such as PSO, ABC, PSO+ABC, and G-ABC have been compared using the Twitter and baseball datasets. The performance metrics such as Precision, Sensitivity, F-measure, and Execution Time have been computed.

- The precision using the Twitter dataset is 0.97 while using the baseball dataset is 0.98 for the NB classifier.
- The average sensitivity value using the NN classifier using the proposed technique is 0.96, 0.95 for ABC+PSO, 0.92 for ABC, and 0.91 for PSO.
- The F-measure analysis shows that 0.91 is obtained using the PSO, 0.94 and 0.96 are obtained using the ABC and PSO+ABC respectively while results using the G-ABC technique are 0.97. Thus, the G-ABC technique provides better results.
- The accuracy analysis using the NN technique shows that 91.5% accuracy is obtained using the PSO, 94% accuracy is obtained using the ABC and PSO+ABC respectively while the

results using the G-ABC technique is 98%. Thus, the G-ABC technique performs better in comparison to other techniques.

- The execution time using the G-ABC technique for the NN classifier is 2.8s and 2.6s respectively comparatively less than other techniques. The performance of the G-ABC technique is superior in comparison to other optimization techniques.

➢ Conclusion at Classification Level

The different classifiers such as NB, KNN, SVM, and NN have been compared for different performance metrics. The analysis results are given as follows:-

- For precision analysis, the least performance is shown by KNN for the Twitter dataset followed by the NB classifier for the baseball dataset, and better performance is shown by NN for both datasets.
- For sensitivity analysis, the least value is shown by KNN followed by the NB classifier and SVM classifier for both datasets. The better performance is shown by NN for both datasets.
- For F-measure analysis, the least value is shown by KNN followed by the NB classifier and SVM classifier for both datasets. The results using the NN classifier for both datasets are better. Thus, the NN classifier provides better results with G-ABC for rule mining.
- For Accuracy analysis, the least accuracy is shown by the NB classifier followed by the KNN classifier and SVM classifier for both dataset. The NN technique performs better for both datasets.
- For execution time analysis, the least performance is shown by NB followed by the KNN classifier for both datasets. The better performance is shown by NN for both datasets.

Overall, the evaluation shows that G-ABC is found to be more suitable at the pre-processing stage for the feature selection. The complete evaluation showed that the neural network proved to be the best classifier in the present framework.

➢ Conclusion at Multiple Simulation Analysis

The simulation study is conducted using many simulations varying from 200 to 1000 for both datasets used in the evaluation. The major idea here is to evaluate the effect of an increase in the

simulation rounds and the neural layers on the performance of the proposed G-ABC+NN. The average performance analysis observed for 1,000 simulation rounds is summarised as follows.

- Overall, an average precision observed using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.9746, 0.9704, 0.9672, 0.9671, 0.9676, and 0.9677, respectively.
- The average recall using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.9807, 0.9795, 0.9795, 0.9375, 0.9387, and 0.9378, respectively.
- The average f-measure observed using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.9776, 0.9731, 0.9732, 0.9517, 0.9524, and 0.9521, respectively.
- An average accuracy using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 92.321%, 91.810%, 91.832%, 88.101%, 88.271%, and 88.218%, respectively.
- The average execution time using Baseball G-ABC+NN 5-10 L, Twitter G-ABC+NN 5-10 L, Twitter NN only, Baseball NN only, Twitter G-ABC+NN 15-30 L, and Baseball G-ABC+NN 15-30 L is 0.2750s, 0.2912s, 0.2878s, 0.2891s, 0.2920s, and 0.2951s, respectively.

The analysis using multiple simulations shows that an increase in the number of neural layers in the proposed G-ABC+NN considerably increases the overall execution time. In addition to this, it has been observed that G-ABC+NN with 5 to 10 neural layers performed better than using 15 to 30 neural layers. Therefore, the number of neural layers is restricted to 5 to 10 to lower the overall execution time without compromising the performance of the proposed framework.

7.2 Future Scope

In the present work, G-ABC with NN has been proposed for advanced data mining based on association rule mining and mean-variance optimization. In the future, more datasets will be involved to justify the effectiveness of the designed framework over a large number of datasets.

Further, more metaheuristic algorithms can be evaluated to resolve the challenges associated with prediction accuracy and selection problems. In addition to this, deep learning will be involved in the presented research work to further improve the feature extraction accuracy to reduce the overall execution time of the process.

REFERENCES

Agarwal, A., & Nanavati, N. (2016). Association rule mining using hybrid GA-PSO for multi-objective optimisation. 2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–7.

Agrawal, J., Agrawal, S., Singhai, A., & Sharma, S. (2015). SET-PSO-based approach for mining positive and negative association rules. Knowledge and information systems, 45, pp. 453-471.

Agrawal, P., Abutarboush, H. F., Ganesh, T., & Mohamed, A. W. (2021). Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019). IEEE Access, 9,pp. 26766–26791.

Agrawal, R., Imieliński, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216.

Agrawal, R., Srikant, R., & others. (1994). Fast algorithms for mining association rules. Proc. 20th Int. Conf. Very Large Data Bases, VLDB, 1215, pp. 487–499.

Al Amrani, Y., Lazaar, M., & El Kadirp, K. E. (2018). Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis. Procedia Computer Science, 127, 511–520. https://doi.org/10.1016/J.PROCS.2018.01.150

Alataş, B., & Akin, E. (2006). An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. Soft Computing, 10, pp. 230-237.

Al-Maolegi, M., & Arkok, B. (2014). An improved Apriori algorithm for association rules. ArXiv Preprint ArXiv:1403.3948.

Al-Maqaleh, B. M. (2013). Discovering interesting association rules: a multi-objective genetic algorithm approach. International Journal of Applied Information Systems, 5(3), pp. 47-52.

Al-Sarem, M.; Saeed, F.; Alsaeedi, A.; Boulila, W.; Al-Hadhrami, T. Ensemble Methods for Instance-Based Arabic Language Authorship Attribution. IEEE Access 2020, 8, pp. 17331–17345.

Alshari, E. M., Azman, A., Doraisamy, S., Mustapha, N., & Alkeshr, M. (2018). Effective method for sentiment lexical dictionary enrichment based on Word2Vec for sentiment analysis. 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP), pp. 1–5.

AlZu'bi, S., Hawashin, B., EIBes, M., & Al-Ayyoub, M. (2018). A novel recommender system based on apriori algorithm for requirements engineering. 2018 Fifth International Conference on Social Networks Analysis, Management and Security (Snams), pp. 323–327.

Amarnath, B., Balamurugan, S., & Alias, A. (2016). Review on feature selection techniques and its impact for effective data classification using UCI machine learning repository dataset. Journal of Engineering Science and Technology, 11(11), pp. 1639–1646.

Anandan, R. (2022). Adaptive Heuristic - Genetic Algorithms. Mathematics in Computational Science and Engineering, pp. 329–342. https://doi.org/10.1002/9781119777557.CH15

Atitallah, S. B., Driss, M., & Almomani, I. (2022). A novel detection and multi-classification approach for IoT-malware using random forest voting of fine-tuning convolutional neural networks. Sensors, 22(11), pp. 4302.

Ayubi, S., Muyeba, M. K., Baraani, A., & Keane, J. (2009). An algorithm to mine general association rules from tabular data. Information Sciences, 179(20), pp. 3520-3539.

Badal, V.D., Kundrotas, P.J. and Vakser, I.A., 2018. Natural language processing in text mining for structural modeling of protein complexes. BMC bioinformatics, 19, pp. 1-10.

Bandana, R. (2018). Sentiment analysis of movie reviews using heterogeneous features. 2018 2nd International Conference on Electronics, Materials Engineering \& Nano-Technology (IEMENTech), pp. 1–4.

Bao, F., Mao, L., Zhu, Y., Xiao, C., & Xu, C. (2021). An improved evaluation methodology for mining association rules. Axioms, 11(1), pp. 17.

Batool, S., Rashid, J., Nisar, M.W., Kim, J., Kwon, H.Y. and Hussain, A., (2023). Educational data mining to predict students' academic performance: A survey study. Education and Information Technologies, 28(1), pp. 905-971.

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2010). Cython: The best of both worlds. Computing in Science \& Engineering, 13(2), pp. 31–39.

Beiranvand, V., Mobasher-Kashani, M., & Bakar, A. A. (2014). Multi-objective PSO algorithm for mining numerical association rules without a priori discretization. Expert systems with applications, 41(9), pp. 4259-4273.

Bhadoriya, V. S., & Dutta, U. (2015). Improved association rules optimization using modified ABC algorithm. International Journal of Computer Applications, 122(13).

Bhatnagar, V., Ahuja, S., & Kaur, S. (2015). Discriminant analysis-based cluster ensemble. International Journal of Data Mining, Modelling and Management, 7(2), pp. 83–107.

Boulila, W., Farah, I. R., Ettabaa, K. S., Solaiman, B., & Ghézala, H. B. (2010, March). Spatio-Temporal Modeling for Knowledge Discovery in Satellite Image Databases. In CORIA, pp. 35-49.

Browne, S., Dongarra, J., Grosse, E., & Rowan, T. (1995). The Netlib mathematical software repository. D-Lib Magazine, 1(9).

Creighton, C., & Hanash, S. (2003). Mining gene expression databases for association rules. Bioinformatics, 19(1), pp. 79-86.

Chang, L., Fu, C., Zhu, W., & Liu, W. (2021). Belief rule mining using the evidential reasoning rule for medical diagnosis. International Journal of Approximate Reasoning, 130, pp. 273–291.

Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. Knowledge-Based Systems, 192, pp. 105361. https://doi.org/10.1016/J.KNOSYS.2019.105361

Chen, Y., Li, F., & Fan, J. (2015). Mining association rules in big data with NGEP. Cluster Computing, 18, pp. 577-585.

Chiclana, F., Kumar, R., Mittal, M., Khari, M., Chatterjee, J. M., Baik, S. W., & others. (2018). ARM--AMO: An efficient association rule mining algorithm based on animal migration optimization. Knowledge-Based Systems, 154, pp. 68–80.

Cioffi, R., Travaglioni, M., Piscitelli, G., Petrillo, A., & De Felice, F. (2020). Artificial intelligence and machine learning applications in smart production: Progress, trends, and directions. Sustainability, 12(2), pp. 492. https://doi.org/10.3390/SU12020492

Cios, K. J., Pedrycz, W., Swiniarski, R. W., Cios, K. J., Pedrycz, W., & Swiniarski, R. W. (1998). Data mining and knowledge discovery pp. 1-26, Springer US. https://doi.org/10.1007/978-1-4615-5589-6_1.

Dey, S., Wasif, S., Tonmoy, D. S., Sultana, S., Sarkar, J., & Dey, M. (2020). A Comparative Study of Support Vector Machine and Naive Bayes Classifier for Sentiment Analysis on Amazon Product Reviews. 2020 International Conference on Contemporary Computing and Applications, IC3A 2020, pp. 217–220. https://doi.org/10.1109/IC3A48958.2020.233300

Diwaker, C., Tomar, P., Poonia, R. C., & Singh, V. (2018). Prediction of software reliability using bio inspired soft computing techniques. Journal of Medical Systems, 42(5), pp. 1–16.

Djenouri, Y., & Comuzzi, M. (2017). Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem. Information Sciences, 420, pp. 1-15.

Djenouri, Y., Belhadi, A., Fournier-Viger, P., & Fujita, H. (2018). Mining diversified association rules in big datasets: A cluster/GPU/genetic approach. Information Sciences, 459, pp. 117-134.

Djenouri, Y., Belhadi, A., Fournier-Viger, P., & Fujita, H. (2018). Data Mining Algorithms and Techniques in Mental Health: A Systematic Review. In Journal of Medical Systems, 42, pp. 61, Springer.

Djenouri, Y., Djenouri, D., Belhadi, A., Fournier-Viger, P., & Lin, J. C.-W. (2018). A new framework for metaheuristic-based frequent itemset mining. Applied Intelligence, 48(12), pp. 4775–4791.

Dong, C., Xiong, Z., Liu, X., Ye, Y., Yang, Y., & Guo, W. (2019). Dual-Search Artificial Bee Colony Algorithm for Engineering Optimization. IEEE Access, 7, pp. 24571–24584. https://doi.org/10.1109/ACCESS.2019.2899743

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical Computer Science, 344(2–3), pp. 243–278.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE Computational Intelligence Magazine, 1(4), pp. 28–39. https://doi.org/10.1109/MCI.2006.329691

Driss, M., Almomani, I., e Huma, Z., & Ahmad, J. (2022). A federated learning framework for cyberattack detection in vehicular sensor networks. Complex & Intelligent Systems, 8(5), pp. 4221-4235.

Duneja, E., & Sachan, A. K. (2012). A survey on frequent itemset mining with association rules. International Journal of Computer Applications, 46(23), pp. 18–24.

Esling, P., & Agon, C. (2012). Time-series data mining. ACM Computing Surveys (CSUR), 45(1), pp. 1–34.

Favaretto, M., De Clercq, E., & Elger, B. S. (2019). Big Data and discrimination: perils, promises and solutions. A systematic review. Journal of Big Data, 6(1), pp. 1–27. https://doi.org/10.1186/S40537-019-0177-4/TABLES/7

Feng Wen, Guo Zhang, Lingfeng Sun, Xingqiao Wang, Xiaowei Xu, “A Hybrid Temporal Association Rules Mining method for Traffic Congestion Prediction”, Computers & Industrial Engineering, Elsevier, Vol 130. pp 779-787, 2019.

Fister, I., Yang, X. S., Fister, D., & Fister, I. (2014). Firefly algorithm: A brief review of the expanding literature. Studies in Computational Intelligence, 516, 347–360. https://doi.org/10.1007/978-3-319-02141-6_17/COVER

Freitas, D., Lopes, L. G., & Morgado-Dias, F. (2020). Particle Swarm Optimisation: A Historical Review Up to the Current Developments. Entropy 2020, 22(3), pp. 362. https://doi.org/10.3390/E22030362

Fung, K. Y., Kwong, C. K., Siu, K. W. M., & Yu, K. M. (2012). A multi-objective genetic algorithm approach to rule mining for affective product design. Expert Systems with Applications, 39(8), pp. 7411-7419.

Gaikwad, M. R., Umbarkar, A. J., & Bamane, S. S. (2020). Large-scale data clustering using improved artificial bee colony algorithm. In ICT Systems and Sustainability, pp. 467–475, Springer.

Ghaleb, F.A.; Maarof, M.A.; Zainal, A.; Al-Rimy, B.; Alsaeedi, A.; Boulila, W. Alrimy Ensemble-Based Hybrid Context-Aware Misbehavior Detection Model for Vehicular Ad Hoc Network. Remote Sens. 2019, 11, pp. 2852.

Gheware, S. D., Kejkar, A. S., & Tondare, S. M. (2014). Data mining: Task, tools, techniques and applications. International Journal of Advanced Research in Computer and Communication Engineering, 3(10).

Ghosh, A., & Nath, B. (2004). Multi-objective rule mining using genetic algorithms. Information Sciences, 163(1-3), pp. 123-133.

Ghosh, M., & Sanyal, G. (2018). An ensemble approach to stabilize the features for multi-domain sentiment analysis using supervised machine learning. Journal of Big Data, 5(1), pp. 1–25.

Ghosh, S., Biswas, S., Sarkar, D., & Sarkar, P. P. (2010). Mining frequent itemsets using genetic algorithm. arXiv preprint arXiv:1011.0328.

Gupta, A., Jain, S. and Tiwari, A., 2019. Optimization and Improvement of association rule mining using genetic algorithm and fuzzy logic. Available at SSRN 3358761.

Gupta, A., Shah, P., & Mehta, D. (2019). Hybrid System to find Association using Genetic Algorithm and Fuzzy Logic. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 7(3), pp. 17–23.

Gupta, B., Rawat, A., Jain, A., Arora, A., & Dhami, N. (2017). Analysis of various decision tree algorithms for classification in data mining. International Journal of Computer Applications, 163(8), pp. 15–19.

Gupta, M. K., & Chandra, P. (2020). A comprehensive survey of data mining. International Journal of Information Technology, 12(4), pp. 1243–1257.

Hammad, M., & Al-Awadi, M. (2016). Sentiment analysis for Arabic reviews in social networks using machine learning. Advances in Intelligent Systems and Computing, 448, pp. 131–139. https://doi.org/10.1007/978-3-319-32467-8_13/COVER

Han, J., Kamber, M., & Mining, D. (2006). Concepts and techniques. Morgan Kaufmann, 340, pp. 93205–94104.

Han, J., Kamber, M., Berzal, F., & Marín, N. (2002). Data mining. ACM SIGMOD Record, 31(2), pp. 66–68. https://doi.org/10.1145/565117.565130

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C. (2000). FreeSpan: frequent pattern-projected sequential pattern mining. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 355–359.

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery, 8(1), pp. 53–87. https://doi.org/10.1023/B:DAMI.0000005258.31418.83/METRICS

Heraguemi, K. E., Kamel, N., & Drias, H. (2016). Multi-swarm bat algorithm for association rule mining using multiple cooperative strategies. Applied Intelligence, 45(4), pp. 1021–1033.

Holzinger, A., Kieseberg, P., Weippl, E., & Tjoa, A. M. (2018). Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11015 LNCS, pp. 1–8. https://doi.org/10.1007/978-3-319-99740-7_1/FIGURES/2

Hong, T. P., Lee, Y. C., & Wu, M. T. (2014). An effective parallel approach for genetic-fuzzy data mining. Expert Systems with Applications, 41(2), pp. 655-662.

Hung, L. N., Thu, T. N. T., & Nguyen, G. C. (2015). An efficient algorithm in mining frequent itemsets with weights over data stream using tree data structure. International Journal of Intelligent Systems and Applications, 7(12), pp. 20.

Huo, W., Feng, X., & Zhang, Z. (2016). An efficient approach for incremental mining fuzzy frequent itemsets with FP-tree. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 24(03), pp. 367–386.

Ilango, S. S., Vimal, S., Kaliappan, M., & Subbulakshmi, P. (2019). Optimization using artificial bee colony based clustering approach for big data. Cluster Computing, 22(5), pp. 12169–12177.

Indira, K., & Kanmani, S. (2015). Mining association rules using hybrid genetic algorithm and particle swarm optimisation algorithm. International Journal of Data Analysis Techniques and Strategies, 7(1), pp. 59–76.

Ishibuchi, H. and Yamamoto, T., (2004). Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy sets and systems, 141(1), pp. 59-88.

Islam, M. S., Saha, S., Rahman, S., & Kashem Mia, M. A. (2018). Pattern identification on protein sequences of neurodegenerative diseases using association rule mining. Proceedings of the Seventh International Conference on Advances in Computing, Electronics and Communication (ACEC 2018), Kuala Lumpur, Malaysia, 10, pp.971–978.

Ismailov, A., Jalil, M. M. A., Abdullah, Z., & Abd Rahim, N. H. (2016). A comparative study of stemming algorithms for use with the Uzbek language. 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), pp. 7–12.

Jain, N., & Srivastava, V. (2013). Data mining techniques: a survey paper. IJRET: International Journal of Research in Engineering and Technology, 2(11), pp. 1163–2319.

Jamal, N., Xianqiao, C., & Aldabbas, H. (2019). Deep learning-based sentimental analysis for large-scale imbalanced twitter data. Future Internet, 11(9), pp. 190.

Jang, W.-I., Nasridinov, A., & Park, Y.-H. (2014). Analyzing and predicting patterns in baseball data using machine learning techniques. Advanced Science and Technology Letters, 62, pp. 37–40.

Jemmali, M., Denden, M., Boulila, W., Srivastava, G., Jhaveri, R. H., & Gadekallu, T. R. (2022). A Novel Model Based on Window-Pass Preferences for Data Emergency Aware Scheduling in Computer Networks. IEEE Transactions on Industrial Informatics, 18(11), pp. 7880-7888.

Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. IEEE Access, 5, pp. 2870–2879.

Jianqiang, Z., Xiaolin, G., & Xuejun, Z. (2018). Deep convolution neural networks for twitter sentiment analysis. IEEE Access, 6, pp. 23253–23260.

Kale, S., & Padmadas, V. (2017). Sentiment analysis of tweets using semantic analysis. 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1–3.

Karaa, W.B.A. and Gribâa, N., (2013). Information retrieval with porter stemmer: a new version for English. In Advances in Computational Science, Engineering and Information Technology: Proceedings of the Third International Conference on Computational Science, Engineering and Information Technology (CCSEIT-2013), KTO Karatay University, June 7-9, 2013, Konya, Turkey-Volume 1, pp. 243-254, Springer International Publishing.

Karunyalakshmi, M., & Tajunisha, N. (2017). Classification of cancer datasets using artificial bee colony and deep feed forward neural networks. International Journal of Advanced Research in Computer and Communication Engineering, 6(2), pp. 33-41.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80(5), pp. 8091–8126.

Ketkar, N. S., Holder, L. B., & Cook, D. J. (2005). Subdue: Compression-based frequent pattern discovery in graph data. Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations, pp. 71–76.

Khalili-Damghani, K., Abdi, F., & Abolmakarem, S. (2018). Hybrid soft computing approach based on clustering, rule mining, and decision tree analysis for customer segmentation problem: Real case of customer-centric industries. Applied Soft Computing, 73, pp. 816–828.

Khan, A., & Ansari, Z. (2015). Soft computing based medical image mining: a survey. International Journal of Computer Trends and Technology (IJCTT), 27(2), pp. 76–79.

Khan, K., & Ramsahai, E. (2020). Categorizing 2019-n-cov twitter hashtag data by clustering. Available at SSRN 3680616, 11(4), pp. 41–52.

Khodakarami, F., & Chan, Y. E. (2014). Exploring the role of customer relationship management (CRM) systems in customer knowledge creation. Information \& Management, 51(1), pp. 27–42.

Kim, K., Aminanto, M. E., & Tanuwidjaja, H. C. (2019). Sentiment Analysis for Movies Reviews Dataset Using Deep Learning Models. International Journal of Data Mining & Knowledge

Management Process (IJDKP), 9(2/3), pp. 27–34. https://doi.org/10.1007/978-981-13-1444-5_4

Kiranmai, B., & Damodaram, A. (2014). A review on evaluation measures for data mining tasks. International Journal Of Engineering And Computer Science, 3(7), pp. 7217–7220.

Krig, S., & Krig, S. (2014). Ground truth data, content, metrics, and analysis. Computer Vision Metrics: Survey, Taxonomy, and Analysis, pp. 283-311.

Kumar, P., Gospodaric, D., & Bauer, P. (2007). Improved genetic algorithm inspired by biological evolution. Soft Computing, 11(10), pp. 923–941.

Kumari, U. (2017). Soft computing applications: A perspective view. 2017 2nd International Conference on Communication and Electronics Systems (ICCES), pp. 787–789.

Kumbhare, T. A., & Chobe, S. V. (2014). An overview of association rule mining algorithms. International Journal of Computer Science and Information Technologies, 5(1), pp. 927–930.

Kuo, R.J., Gosumolo, M. and Zulvia, F.E., (2019). Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining. Neural Computing and Applications, 31(8), pp. 3559-3572.

Langdon, W.B. and Barrett, S.J., (2005). Genetic programming in data mining for drug discovery. Evolutionary computation in data mining, pp. 211-235.

Lee, D. G., Ryu, K. S., Bashir, M., Bae, J. W., & Ryu, K. H. (2013). Discovering medical knowledge using association rule mining in young adults with acute myocardial infarction. Journal of medical systems, 37, pp. 1-10.

Liao, S.-H., Chu, P.-H., & Hsiao, P.-Y. (2012). Data mining techniques and applications--A decade review from 2000 to 2011. Expert Systems with Applications, 39(12), pp. 11303–11311.

Liao, T. W., & Triantaphyllou, E. (2008). Recent advances in data mining of enterprise data: algorithms and applications. Google Books.

Lim, A. H., Lee, C. S., & Raman, M. (2012). Hybrid genetic algorithm and association rules for mining workflow best practices. Expert Systems with Applications, 39(12), pp. 10544-10551.

Liu, L., Wen, J., Zheng, Z., & Su, H. (2021). An improved approach for mining association rules in parallel using Spark Streaming. International Journal of Circuit Theory and Applications, 49(4), pp. 1028–1039.

Liu, Y. (2010). Study on application of apriori algorithm in data mining. 2010 Second International Conference on Computer Modeling and Simulation, 3, pp. 111–114.

Lovins, J. B. (1968). Development of a stemming algorithm. Mech. Transl. Comput. Linguistics, 11(1–2), pp. 22–31.

Luna, J. M., Romero, J. R., Romero, C., & Ventura, S. (2014). Reducing gaps in quantitative association rules: A genetic programming free-parameter algorithm. Integrated Computer-Aided Engineering, 21(4), pp. 321-337.

Lungeanu, D., Zaharie, D., & Zamfirache, F. (2008). Influence of Missing Values Handling on Classification Rules Evolved from Medical Data. In ICDM (Posters and Workshops) pp. 86-95.

Madasu, A., & Elango, S. (2019). Efficient feature selection techniques for sentiment analysis. Multimedia Tools and Applications 2019, 79(9), pp. 6313–6335. https://doi.org/10.1007/S11042-019-08409-Z

Makridakis, S. (2017). The forthcoming Artificial Intelligence (AI) revolution: Its impact on society and firms. Futures, 90, pp. 46–60. https://doi.org/10.1016/J.FUTURES.2017.03.006

Mampaey, M., & Vreeken, J. (2013). Summarizing categorical data by clustering attributes. Data Mining and Knowledge Discovery, 26(1), pp. 130–173.

Martínez-Ballesteros, M., Martínez-Álvarez, F., Troncoso, A., & Riquelme, J. C. (2011). An evolutionary algorithm to discover quantitative association rules in multidimensional time series. Soft Computing, 15, pp. 2065-2084.

Mata, J., Alvarez, J. L., & Riquelme, J. C. (2002). Discovering numeric association rules via evolutionary algorithm. In Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings 6, pp. 40-51, Springer Berlin Heidelberg.

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. Pattern recognition, 33(9), pp. 1455-1465.

Menaga, D. and Saravanan, S., 2021. GA-PPARM: constraint-based objective function and genetic algorithm for privacy preserved association rule mining. Evolutionary Intelligence, pp.1-12.

Minaei-Bidgoli, B., Barmaki, R., & Nasiri, M. (2013). Mining numerical association rules via multi-objective genetic algorithms. Information Sciences, 233, pp. 15-24.

Mohana, R. S., Kalaiselvi, S., Kousalya, K., Lohappriya, D., & others. (2021). Twitter based sentiment analysis to predict public emotions using machine learning algorithms. 2021 Third

International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1759–1763.

Mohanapriya, M., & Lekha, J. (2018). Comparative study between decision tree and knn of data mining classification technique. Journal of Physics: Conference Series, 1142(1), pp. 12011.

Mohapatra, D., Tripathy, J., Mohanty, K. K., & Nayak, D. S. K. (2021). Interpretation of Optimized Hyper Parameters in Associative Rule Learning using Eclat and Apriori. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), pp. 879–882.

Morik, K., Bhaduri, K., & Kargupta, H. (2012). Introduction to data mining for sustainability. Data Mining and Knowledge Discovery, 24(2), pp. 311.

Moslehi, F. and Haeri, A., (2020). A novel hybrid wrapper–filter approach based on genetic algorithm, particle swarm optimization for feature subset selection. Journal of Ambient Intelligence and Humanized Computing, 11(3), pp. 1105-1127.

Moslehi, F., Haeri, A. and Martínez-Álvarez, F., (2019). A novel hybrid GA–PSO framework for mining quantitative association rules. Soft Computing, pp. 1-22.

Nagarajan, S.M., Muthukumaran, V., Murugesan, R., Joseph, R.B. and Munirathanam, M., (2021). Feature selection model for healthcare analysis and classification using classifier ensemble technique. International Journal of System Assurance Engineering and Management, pp. 1-12.

Nahar, J., Imam, T., Tickle, K.S. and Chen, Y.P.P., (2013). Association rule mining to detect factors which contribute to heart disease in males and females. Expert Systems with Applications, 40(4), pp. 1086-1093.

Neysiani, B. S., Soltani, N., Mofidi, R., & Nadimi-Shahraki, M. H. (2019). Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm. International Journal of Information Technology and Computer Science, 11(2), pp. 48–55.

Nguyen, B. H., Xue, B., & Zhang, M. (2020). A survey on swarm intelligence approaches to feature selection in data mining. Swarm and Evolutionary Computation, 54, pp. 100663. https://doi.org/10.1016/J.SWEVO.2020.100663

Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., ... & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. Artificial Intelligence Review, 52, pp. 77-124.

Nti, I. K., Quarcoo, J. A., Aning, J., & Fosu, G. K. (2022). A mini-review of machine learning in big data analytics: Applications, challenges, and prospects. Big Data Mining and Analytics, 5(2), pp. 81–97. https://doi.org/10.26599/BDMA.2021.9020028

Othman, P. S., Ihsan, R. R., & Abdulhakeem, R. M. (2022). The Genetic Algorithm (GA) in Relation to Natural Evolution. Academic Journal of Nawroz University, 11(3), pp. 243–250. https://doi.org/10.25007/AJNU.V11N3A1414

Paice, C. D. (1990). Another stemmer. ACM Sigir Forum, 24(3), 56–61.

Pal, A., & Kumar, M. (2020). Distributed synthesized association mining for big transactional data. Indian Academy of Sciences, 45(1), 1–13.

Pei, J. (2001). Mining sequential patterns efficiently by prefix-projected pattern growth. International Conference of Data Engineering (ICDE2001), April.

Perera, I., & Caldera, H. A. (2017). Aspect based opinion mining on restaurant reviews. 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), pp. 542–546.

Wakabi-Waiswa, P. P., & Baryamureeba, V. (2008). K. arukeshi,". Generalized Association Rule Mining Using Genetic Algorithms..

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14, pp. 130–137.

Porter, M. F. (2001). Snowball: A language for stemming algorithms.

Pu, Q., Gan, J., Qiu, L., Duan, J. and Wang, H., (2021). An efficient hybrid approach based on PSO, ABC and k-means for cluster analysis. Multimedia Tools and Applications, pp. 1-19.

Qodmanan, H.R., Nasiri, M. and Minaei-Bidgoli, B., (2011). Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. Expert Systems with applications, 38(1), pp. 288-298.

Rajab, K. D. (2019). New associative classification method based on rule pruning for classification of datasets. IEEE Access, 7, pp. 157783–157795.

Rana, M., & Singla, J. (2022, April). A Framework for Selecting Features using Various Soft Computing Algorithms. In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1743-1748, IEEE.

Rehman, M. U., Shafique, A., Khalid, S., Driss, M., & Rubaiee, S. (2021). Future forecasting of COVID-19: a supervised learning approach. Sensors, 21(10), pp. 3322.

Sahota, S., & Verma, P. (2016). Improved association rule mining based on ABC. Int. J. Comput. Appl, 135(10), pp. 6–10.

Salamon, S. J., Hansen, H. J., & Abbott, D. (2019). How real are observed trends in small correlated datasets? Royal Society Open Science, 6(3), pp. 181089.

Sarkar, A. (2012). Application of fuzzy logic in transport planning. International Journal on Soft Computing (IJSC), 3(2), pp. 1–21.

Sarkar, S., Lohani, A., & Maiti, J. (2017). Genetic algorithm-based association rule mining approach towards rule generation of occupational accidents. In Computational Intelligence, Communications, and Business Analytics: First International Conference, CICBA 2017, Kolkata, India, March 24–25, 2017, Revised Selected Papers, Part II, pp. 517-530, Springer Singapore.

Sarker, I. H., & Kayes, A. S. M. (2020). ABC-RuleMiner: User behavioral rule-based machine learning method for context-aware intelligent services. Journal of Network and Computer Applications, 168, 102762.

Serrano-Guerrero, J., Romero, F. P., & Olivas, J. A. (2021). Fuzzy logic applied to opinion mining: a review. Knowledge-Based Systems, 222, pp. 107018.

Shahin, M., Inoubli, W., Shah, S. A., Yahia, S. Ben, & Draheim, D. (2021). Distributed scalable association rule mining over covid-19 data. International Conference on Future Data and Security Engineering, pp. 39–52.

Shanmuganathan, S. (2016). Artificial neural network modelling: An introduction. Studies in Computational Intelligence, 628, pp. 1–14. https://doi.org/10.1007/978-3-319-28495-8_1/COVER

Sharma, H., & Kumar, S. (2016). A survey on decision tree algorithms of classification in data mining. International Journal of Science and Research (IJSR), 5(4), pp. 2094–2097.

Sharma, M. (2014). Data mining: A literature survey. International Journal of Emerging Research in Management & Technology, 3(2).

Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. International Journal of Engineering Applied Sciences and Technology, 4(12), 310–316. http://www.ijeast.com

Sharmila, S. and Vijayarani, S., 2021. Association rule mining using fuzzy logic and whale optimization algorithm. Soft Computing, 25, pp.1431-1446.

Sharmila, S., & Vijayarani, S. (2021). Association rule mining using fuzzy logic and whale optimization algorithm. Soft Computing, 25(2), pp. 1431–1446.

Shawkat, M., Badawi, M., El-ghamrawy, S., Arnous, R., & El-desoky, A. (2022). An optimized FP-growth algorithm for discovery of association rules. The Journal of Supercomputing, 78(4), pp. 5479–5506.

Sherdiwala, K. B., & Khanna, S. O. (2018). Impact of Association Rule Mining in Stock Market. Print) International Research Journal of Management Science & Technology, 9(9), pp. 2348–9367. http://www.irjmst.com

Shively, T. S., Ansley, C. F., & Kohn, R. (1990). Fast evaluation of the distribution of the Durbin-Watson and other invariant test statistics in time series regression. Journal of the American Statistical Association, 85(411), pp. 676–685.

Sinaei, S., & Fatemi, O. (2018). Run-time mapping algorithm for dynamic workloads using association rule mining. Journal of Systems Architecture, 91, pp. 1–10.

Singh Raghuwanshi, A., & Kumar Pawar Asst prof, S. (2017). Polarity Classification of Twitter Data using Sentiment Analysis . International Journal on Recent and Innovation Trends in Computing and Communication, 5(6), pp. 434–439. http://www.ijritcc.org

Singh, P. (2016). Applications of Soft Computing in Time Series Forecasting. 330. https://doi.org/10.1007/978-3-319-26293-2

Singh, S., Garg, R., & Mishra, P. K. (2018). Performance optimization of MapReduce-based Apriori algorithm on Hadoop cluster. Computers \& Electrical Engineering, 67, pp. 348–364.

Smith-Miles, K., Wreford, B., Lopes, L., & Insani, N. (2013). Predicting metaheuristic performance on graph coloring problems using data mining. In Hybrid metaheuristics, pp. 417–432, Springer.

Soria, D., Garibaldi, J. M., Ambrogi, F., Biganzoli, E. M., & Ellis, I. O. (2011). A 'non-parametric' version of the naive Bayes classifier. Knowledge-Based Systems, 24(6), pp. 775–784. https://doi.org/10.1016/J.KNOSYS.2011.02.014

Sornalakshmi, M., Balamurali, S., Venkatesulu, M., Navaneetha Krishnan, M., Ramasamy, L. K., Kadry, S., Manogaran, G., Hsu, C.-H., & Muthu, B. A. (2020). Hybrid method for mining rules based on enhanced Apriori algorithm with sequential minimal optimization in healthcare industry. Neural Computing and Applications, pp. 1–14.

Sornalakshmi, M., Balamurali, S., Venkatesulu, M., Navaneetha Krishnan, M., Ramasamy, L.K., Kadry, S., Manogaran, G., Hsu, C.H. and Muthu, B.A., (2020). Hybrid method for mining rules based on enhanced Apriori algorithm with sequential minimal optimization in healthcare industry. Neural Computing and Applications, pp. 1-14.

Sriram, M., Bhanja, C., Naik, B., Behera, H. S., & Nayak, J. (2015). A Comprehensive Survey on Support Vector Machine in Data Mining Tasks: Applications & Challenges. International Journal of Database Theory and Application, 8(1), pp. 169–186. https://doi.org/10.14257/ijdta.2015.8.1.18

Stančin, I., & Jović, A. (2019). An overview and comparison of free Python libraries for data mining and big data analysis. 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 977–982.

Subbulakshmi, C. V, & Deepa, S. N. (2015). Medical dataset classification: a machine learning paradigm integrating particle swarm optimization with extreme learning machine classifier. The Scientific World Journal, 2015.

Sumathi, S., & Sivanandam, S. N. (2006). Introduction to data mining and its applications (Vol. 29). Springer.

Sumit, S. H., Hossan, M. Z., Al Muntasir, T., & Sourov, T. (2018). Exploring word embedding for bangla sentiment analysis. 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), pp. 1–5.

Telikani, A., Gandomi, A. H., & Shahbahrami, A. (2020). A survey of evolutionary computation for association rule mining. Information Sciences, 524, pp. 318–352.

Thurachon, W., & Kreesuradej, W. (2021). Incremental association rule mining with a fast incremental updating frequent pattern growth algorithm. IEEE Access, 9, pp. 55726–55741.

Tsai, C. W., Lai, C. F., Chao, H. C., & Vasilakos, A. V. (2015). Big data analytics: a survey. Journal of Big Data, 2(1), pp. 1–32. https://doi.org/10.1186/S40537-015-0030-3/TABLES/3

Tsang, C.H., Kwong, S. and Wang, H., (2007). Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. Pattern Recognition, 40(9), pp. 2373-2391.

Turban, E. (2011). Decision support and business intelligence systems. Pearson Education India.

Vanipriya, C. H., & Thammi Reddy, K. (2014). Indian Stock Market Predictor System. ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol II, pp. 17–26.

Venkatadri, M., & Reddy, D. L. C. (2011). A Review on Data mining from Past to the Future. International Journal of Computer Applications, 975(2011), pp. 8887.

Vijiyarani, S., & Sudha, S. (2013). Disease prediction in data mining technique--a survey. International Journal of Computer Applications \& Information Technology, 2(1), pp. 17–21.

Wang, B., Rahal, I., & Dong, A. (2011). Parallel hierarchical clustering using weighted confidence affinity. International Journal of Data Mining, Modelling and Management, 3(2), pp. 110–129.

Wang, H.-B., & Gao, Y.-J. (2021). Research on parallelization of Apriori algorithm in association rule mining. Procedia Computer Science, 183, pp. 641–647.

Ward, L., Dunn, A., Faghaninia, A., Zimmermann, N. E. R., Bajaj, S., Wang, Q., Montoya, J., Chen, J., Bystrom, K., Dylla, M., Chard, K., Asta, M., Persson, K. A., Snyder, G. J., Foster, I., & Jain, A. (2018). Matminer: An open source toolkit for materials data mining. Computational Materials Science, 152, pp. 60–69. https://doi.org/10.1016/J.COMMATSCI.2018.05.018

Washio, T., & Motoda, H. (2003). State of the art of graph-based data mining. Acm Sigkdd Explorations Newsletter, 5(1), pp. 59–68.

Yan, X., & Han, J. (2002). gspan: Graph-based substructure pattern mining. 2002 IEEE International Conference on Data Mining, 2002. Proceedings., pp. 721–724.

Yang, X.-S., & Deb, S. (2009). Cuckoo search via Lévy flights. 2009 World Congress on Nature \& Biologically Inspired Computing (NaBIC), pp. 210–214.

Yazgana, P., & Kusakci, A. O. (2016). A literature survey on association rule mining algorithms. Southeast Europe Journal of Soft Computing, 5(1).

Zhang, A., Shi, W., & Webb, G. I. (2016). Mining significant association rules from uncertain data. Data Mining and Knowledge Discovery, 30(4), pp. 928–963.

Zhang, F., Liu, M., Gui, F., Shen, W., Shami, A., & Ma, Y. (2015). A distributed frequent itemset mining algorithm using Spark for Big Data analytics. Cluster Computing, 18(4), pp. 1493–1501.

Zhang, M., Fan, J., Sharma, A. and Kukkar, A., (2022). Data mining applications in university information management system development. Journal of Intelligent Systems, 31(1), pp. 207-220.

Zheng, H., He, J., Liu, Q., Li, J., Huang, G., & Li, P. (2022). Multi-objective optimisation based fuzzy association rule mining method. World Wide Web, pp. 1–18.

Zimbra, D., Ghiassi, M., & Lee, S. (2016). Brand-related Twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. 2016 49th Hawaii International Conference on System Sciences (HICSS), pp. 1930–1938.

LIST OF PUBLICATIONS

Research Papers Published in Journals:

1. Mrinalini Rana, Omdev Dahiya, "Grouped ABC For Feature Selection And Mean-Variance Optimization For Rule Mining : A Hybrid Framework", IEEE Access, pp. 85747-85759, 2023. **(*SCIE*, Impact factor 3.47).**
2. Mrinalini Rana, Omdev Dahiya, "A Hybrid Grouped-Artificial Bee Colony Optimization (G-ABC) Technique for Feature Selection and Mean-Variance Optimization for Rule Mining," International Journal of Engineering Trends and Technology, vol. 71, no. 4, pp. 12-20, 2023. Crossref, https://doi.org/10.14445/22315381/IJETT-V71I4P202 (*Scopus indexed*).
3. Mrinalini Rana, Jimmy Singla, "A Comprehensive Review of Data Mining Rules Generation Algorithms.", Turkish Online Journal of Qualitative Inquiry, pp. 12(6), 2021.

Research Papers Published/Presented in Conference:

1. Mrinalini Rana, Jimmy Singla, "A Framework for Selecting Features using Various Soft Computing Algorithms.", In 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 1743-1748. IEEE, April 2022. (*Scopus indexed*)
2. Mrinalini Rana, Jimmy Singla, "A Pre-processing Model for Feature Extraction Based on K-mean, PSO and ABC", In 2021 International Conference on Computing Sciences (ICCS) (pp. 118-122). IEEE, December 2021. (*Scopus indexed*)
3. Mrinalini Rana, Jimmy Singla, "A systematic review on data mining rules generation optimizing via genetic algorithm", In Proceedings of the International Conference on Innovative Computing & Communications (*ICICC*), March 2020.
4. Mrinalini Rana, Omdev Dahiya," Rule mining using various soft computing algorithms: A Review" (*Communicated*)
5. Mrinalini Rana, Omdev Dahiya, "A hybrid framework using soft computing algorithm for rule mining " (*Communicated*)

Graphical Abstract submitted as Copyright :

1. Mrinalini Rana, Omdev Dahiya, “A hybrid approach using soft computing algorithm for feature selection and rule mining” (***Registered****, Registration Number: L-127326/2023*)