COVID-19 Prediction Using Supervised Machine Learning

by

Irfan Ali

B.Sc., Sir Syed University of Engineering and Technology, 2015

A report submitted in partial fulfillment of the requirements for the degree of

Master of Engineering

in the Department of Electrical and Computer Engineering

©Irfan Ali, 2023

University of Victoria

All rights reserved. This report may not be reproduced in whole or in part, by photocopy or other means, without the permission of the author

COVID-19 Prediction Using Supervised Machine Learning

by

Irfan Ali

B.Sc., Sir Syed University of Engineering and Technology, 2015

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
Department of Electrical and Computer Engineering

Dr. Mihai Sima, Departmental Member
Department of Electrical and Computer Engineering

Abstract

Early diagnosis is important to stop the spread of illnesses that endanger human life. COVID-19 is a contagious disease that has mutated into multiple variants and created a global epidemic that requires immediate diagnosis. With the increase in COVID-19 cases, the amount of associated data grows every day, and data mining can be used to extract information from this data. In this project, a COVID-19 symptoms and presence dataset is used with several supervised machine learning algorithms to predict COVID-19 in the human body by examining the symptoms. The Bayes Net, Simple Logistic, Bagging, Support Vector Machine (SVM), and AdaBoost M1 classifiers are considered using the open-source Waikato Environment for Knowledge Analysis (WEKA) Machine Learning (ML) tool. Principal Component Analysis (PCA) is used to reduce the number of features in the dataset based on eigenvalues. Then the model is trained and tested using 5-fold cross-validation, 10-fold cross-validation, and 66/34 and 34/66 splits. The performance of the models is evaluated based on accuracy, precision, recall, F-measure, and execution time. The results obtained show that Bagging outperforms the other classifiers with an accuracy of 99.3% and an execution time of 0.10 s for a 66/34 split using 10 features.

Contents

List of Figures

List of Tables

Glossary

ANN ..... Artificial Neural Network
CLI ..... Command Line Interface
CNN ..... Convolutional Neural Network
DNN ..... Deep Neural Network
FP ..... False Positive
GPU..... General Processing Unit
KNN ..... K-Nearest Neighbor
MAE..... Mean Absolute Error
ML..... Machine Learning
PCA..... Principal Component Analysis
RMSD ..... Root Mean Square Difference
SMOTE ..... Synthetic Minority Oversampling Technique
SVM ..... Support Vector Machine
TP ..... True Positive
WEKA..... Waikato Environment for Knowledge Analysis

Acknowledgment

First, I very much thank God for his countless blessings. I also thank my parents for their continuous love, support, and encouragement.

I am immensely thankful to Dr. T. Aaron Gulliver for accepting me into his research group as a master's student. I would not have achieved academic success without his support and encouragement.

Dedication

To my parents and my brother Nasir Ali, their unconditional love and support throughout all stages of my life have known no bounds. Their encouragement is a great reason for my success. I dedicate this to all three of them.

Chapter 1 Introduction

COVID-19 was first identified at the end of December 2019 in Wuhan, Hubei province, China [1]. It was declared a global emergency and a threat to human health at the end of January 2020 by the World Health Organization (WHO), and at the start of March 2020 WHO declared it a global pandemic [2]. COVID-19 is an infectious disease that primarily affects the respiratory system [3]. From the start of 2020 through 2021, COVID-19 spread throughout the world at an incredible rate and caused extreme fear and panic around the globe. In 2022, people learned to coexist with COVID-19 in the same manner as with previous pandemics like influenza.

COVID-19 is a multisystemic disease, not merely a respiratory one. According to recent research, this virus can affect most body organs and triggers an extensive inflammatory response [3]. Approximately 10-15% of people affected by COVID-19 are expected to develop severe symptoms. There are also possible complications in the nervous system, heart, and lungs due to COVID-19. It can stay active in the human body for up to 10 days after a person begins to experience symptoms [4].

People who were diagnosed with COVID-19 reported symptoms like cough, fever, loss of smell, tiredness, sore throat, loss of taste, headache, diarrhea, and body aches ranging from mild to severe. People with extreme symptoms were told to seek medical attention immediately, but in the case of mild symptoms, the public was advised to stay home and follow all precautions. During the outbreak, people were asked to follow measures such as keeping a safe distance from each other, getting vaccinated as soon as possible, wearing masks, and washing their hands frequently [5]. In this project, supervised Machine Learning (ML) models are used to predict COVID-19. The presence of this virus in the human body is predicted using the symptoms in the COVID-19 dataset from the Kaggle website.

1.1 Motivation

As of May 16, 2022, there have been 521,366,398 cases of COVID-19 recorded worldwide with 6,288,682 deaths due to this disease. Overall, 491,948,944 patients have recovered from the disease [6]. Due to genetic changes, COVID-19 continuously evolves into new variants, and these have been reported in countries such as South Africa, India, UK, USA, and Brazil. This has resulted in COVID-19 becoming more severe and dangerous as the transmission rate and death rate both increased, and vaccines became less effective [7]. Although significant efforts have been made to stop COVID-19, large outbreaks have occurred which temporarily immobilized daily human life.

Due to the COVID-19 outbreak, hospital resources became depleted with insufficient healthcare staff, medical equipment, and testing kits [6]. As access to COVID-19 testing kits was limited, it was difficult to diagnose the disease in its early stages which hindered the healthcare system. Therefore, it would be helpful to have a machine-based predictive model for determining whether COVID-19 is present in the human body.

1.2 Related Work

In the past, several supervised ML algorithms have been developed to predict diseases. There have been attempts to predict the existence of COVID-19 in its early stages. A Convolutional Neural Network (CNN) was used to obtain an accuracy of 76% using a dataset from a hospital in São Paulo, Brazil [8]. Another attempt was made to create a model using preliminary blood test data which indicated that white blood cells, age, neutrophils, and lymphocytes were the primary causes of severity, and this information was used to increase prediction accuracy [9].

ML was employed in [10] to predict the mortality rate of COVID-19 patients using an XGBoost model. The numbers of deceased, recovered, asymptomatic, symptomatic, and life-threatening patients were predicted in [11] using a Stochastic Fractal Search. Polynomial Neural Networks (PNNs) were used in [12] to predict COVID-19 confirmed cases and fatalities in Asia. This project considers supervised ML models to predict COVID-19 in the human body by examining the symptoms. The Waikato Environment for Knowledge Analysis (WEKA) tool is employed for this purpose [13]. It provides a variety of applications for data classification, processing, regression, association, and clustering.

1.3 Report Outline

This report is structured as follows.

Chapter 1 gave a brief overview of COVID-19 including its symptoms and methods of prevention. It also provided global COVID-19 statistics. Related work in the field was discussed along with the motivation for this study.

Chapter 2 introduces the ML techniques used to build a COVID-19 prediction model. The WEKA tool is introduced. It is used for the prediction of COVID-19 in the human body by examining the symptoms.

Chapter 3 presents the COVID-19 prediction system and provides information about the testing environment. Then the classifiers used in this project, namely Bayes Net, Simple Logistic, AdaBoost M1, SVM, and Bagging, are introduced.

Chapter 4 presents the evaluation metrics and the performance results for the classifiers.

Chapter 5 concludes this report and summarizes the results. Some suggestions for future work are also given.

Chapter 2 Machine Learning

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables learning without explicit programming. ML origins are in mathematics, computer science, data mining, and statistics. A dataset is used for training, and then the ML algorithm is used to predict patterns in new data. The types of ML algorithms are described below.

2.1 Types of Machine Learning Algorithms

ML algorithms can be divided into three types.

- Supervised learning
- Unsupervised learning
- Reinforcement learning

2.1.1 Supervised Learning

Supervised learning is commonly used because it is the easiest ML technique. A supervised ML algorithm is trained on labeled datasets to produce the intended output [14]. After training, the labels of new data are predicted. Applications of supervised learning include the classification of spam and face recognition.

There are two types of supervised learning.

- Regression algorithms predict continuous values like price, age, and wages.
- Classification algorithms determine discrete values like numbers of children and can be binary or multi-class.

Bayes Net, Support Vector Machine (SVM), and Simple Logistic are examples of supervised learning algorithms.

2.1.2 Unsupervised Learning

Unsupervised learning is an ML technique used to determine hidden patterns from unlabeled data. As a result, the model is trained without labels so the results may be less precise than with supervised learning. Unsupervised learning is used for grouping and dimensionality reduction [15]. Hidden Markov Model (HMM) and K-means clustering are examples of unsupervised learning algorithms.

2.1.3 Reinforcement Learning

Reinforcement learning is a type of ML that learns by understanding and analyzing the surroundings [15]. It is analogous to how people think and absorb information from their environment. Some well-known reinforcement learning methods are Q-learning and temporal difference.

2.2 Introduction to WEKA

The Waikato Environment for Knowledge Analysis (WEKA) is an open software package that provides tools for data analysis, data processing, classification, regression, clustering, and data visualization. WEKA can easily be installed and has a user-friendly Graphical User Interface (GUI). It can convert datasets from one format to another. It also facilitates the installation of additional packages via the package manager GUI available under the tools bar of the WEKA home page using the Command Line Interface (CLI). The package manager GUI has a search bar to find packages that are either installed or available to install. The home page of WEKA is shown in Figure 2.1. The WEKA applications are described below.

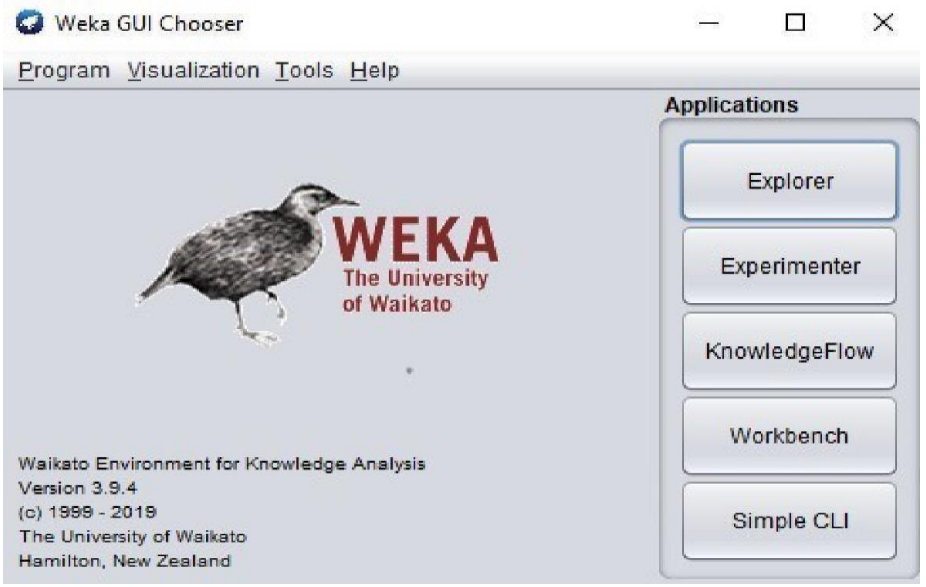


Figure 2-1 The WEKA home page.

2.2.1 Explorer

Explorer is used to load, classify, save, and configure datasets and perform ML analysis.

2.2.2 Experimenter

Experimenter allows users to setup, run, and then analyze the results of large-scale experiments.

2.2.3 Knowledge Flow

Knowledge flow is used to analyze and visualize the data.

2.2.4 Workbench

Workbench combines the WEKA GUI windows into a single convenient interface. It is helpful when multiple WEKA applications are being used.

2.2.5 Simple CLI

Simple CLI provides access to WEKA applications such as classifiers, filters, and clusters.

Chapter 3 COVID-19 Prediction System

Figure 3.1 gives the COVID-19 prediction system for training and testing the ML models. Each component is described in detail in the following sections.

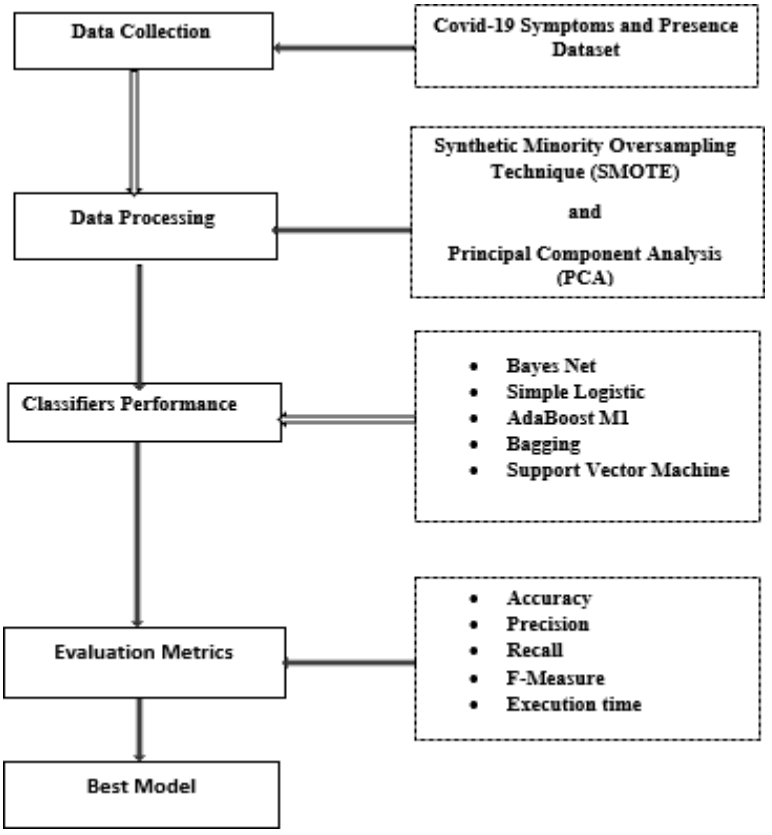


Figure 3-1 The COVID-19 prediction system using supervised machine learning algorithms.

3.1 Data Collection

The COVID-19 dataset was obtained from the Kaggle website which is publicly available [16]. The dataset consists of 21 features. They can be used as input to a ML model and the outputs fall into one of two classes, infected (yes) or not infected (no). The first 20 features give information related to having COVID-19 and the last feature indicates if the patient has COVID-19 or not. There are 5434 entries of which 4383 entries belong to the yes class and 1051 to the no class. This dataset was compiled by the World Health Organization (WHO) and All India Institute of Medical Sciences (AIIMS) from April 17, 2020, to August 20, 2020 [16].Table 3.1 gives the feature names and their descriptions.

Feature Name	**Description**
Breathing Problem	The person has shortness of breath.
Fever	The person has a temperature above normal.
Dry Cough	The person has continuous coughing without phlegm.
Sore Throat	The person has a sore throat.
Running Nose	The person has a runny nose.
Asthma	The person has asthma.
Chronic Lung Disease	The person has lung disease.
Headache	The person has a headache.
Heart Disease	The person suffers from or has a history of heart disease.
Diabetes	The person suffers from or has a history of diabetes.
Hypertension	The person has high blood pressure.
Fatigue	The person has tiredness.
Gastrointestinal	The person has gastrointestinal problems.
Abroad Travel	The person recently left the country.
Contact with COVID-19 Patient	The person had close contact with people infected with COVID-19.
Attended Large Gathering	The person or someone from their family recently attended a mass gathering.
Visited Public Exposed Places	The person has recently visited malls, temples, or other public places.
Family Working in Public Exposed Places	The person or someone in their family is working in a market, hospital, or another crowded place.
Wearing Mask	The person is wearing a face mask properly.
Sanitation from Market	The person bought sanitizing products but did not use them.
COVID-19	The presence of COVID-19.

Table 3-1 Feature names and their descriptions.

3.2 Data Processing

An integral part of developing an ML model is data processing. First, WEKA explorer is used to load the dataset. Then, the feature names, missing values, data types, and labels are visible. The COVID-19 symptoms and presence dataset have an imbalance between the yes (blue) class with 4383 values and the no (red) class with 1051 values. This imbalance is shown in the bar chart in Figure 3.2.

3.3 Synthetic Minority Oversampling Technique (SMOTE)

To overcome class imbalance, SMOTE is applied using WEKA. SMOTE uses oversampling to balance a dataset. It is used in ML algorithms to avoid overfitting issues. SMOTE employs the K-Nearest Neighbor (KNN) algorithm to balance the data. The synthetic data is formed by selecting values between random minority class data and randomly selected nearest neighbors [23]. This procedure is repeated until the data classes are near equal. After applying SMOTE, the number of minority class (red) values is 4204 as shown in Figure 3.3.

3.4 Principal Component Analysis (PCA)

PCA decreases the number of dimensions in the dataset while preserving information variation. This reduction is achieved by identifying the principal components along which the data variation is greatest [17]. The WEKA tool is first used in this project to normalize the dataset features. The relationship between the features is then identified using the correlation matrix. The eigenvectors and eigenvalues are obtained through eigen decomposition. The primary components are represented by the eigenvectors while the eigenvalues give the variances of the components. The largest eigenvectors in the dataset correspond to the principal components. The smaller components with small eigenvalues are removed. In this project, the initial 21 features are reduced to 15, 10, and 5 features. The first 20 features in Table 3.1 are in order from largest to smallest eigenvalues. The resulting datasets are then used to train and test the ML classifiers, namely Bayes Net, Simple logistic, AdaBoost M1, Bagging, and SVM. These classifiers are described in the next section.

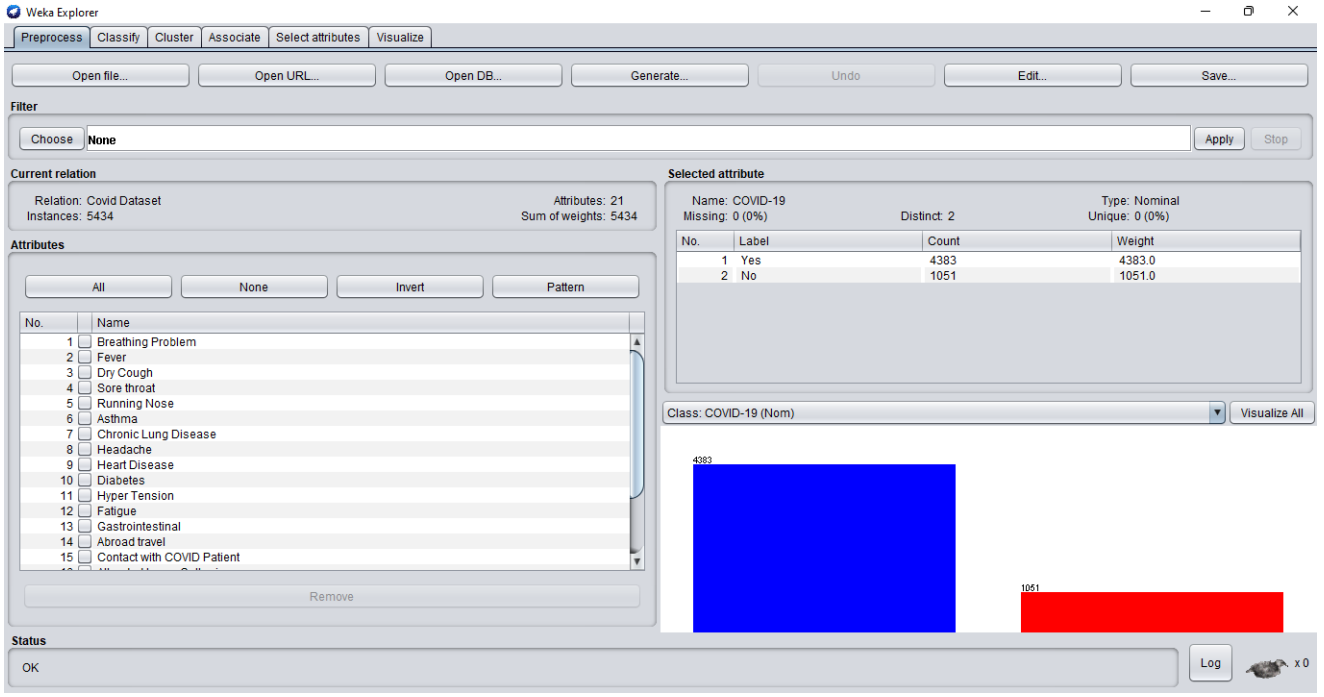


Figure 3-2 The WEKA explorer window.

.

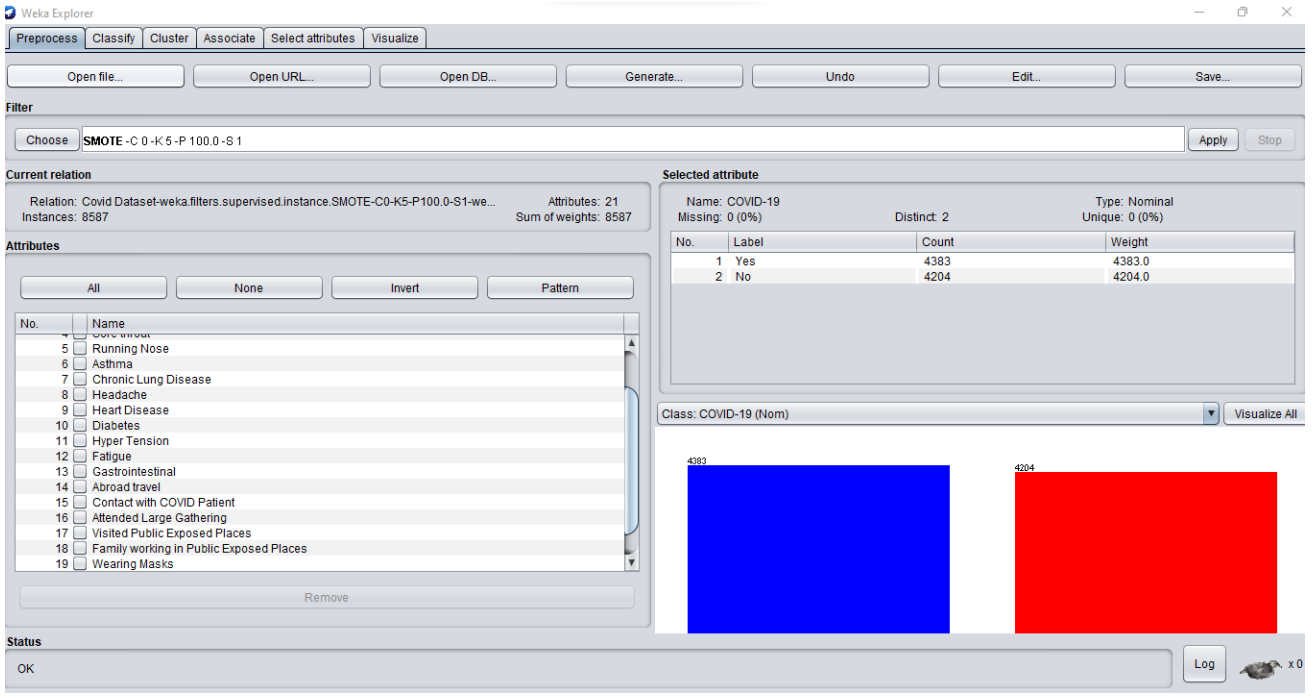


Figure 3-3 The balanced dataset after applying SMOTE.

3.5 Machine Learning Classifiers

The ML classifiers used in this project are given below.

3.5.1 Bayes Net

Bayes Net is a type of directed graph used for probabilistic models. The nodes of the network denote random variables and the links indicate how they are related [18]. It can represent knowledge about casual probability relationships between variables which helps in decision-making. Bayes Net has dependencies and conditional probabilities and corresponds to a directed acyclic graph with no self-connections or loops.

3.5.2 Simple Logistic

Simple Logistic is a statistical model that predicts the probability that an event will occur [19]. It works well when the data are simply related, but when the data is complex and non-linear it may work poorly. Simple Logistic is sensitive to unusually large or small values.

3.5.3 AdaBoost M1

AdaBoost M1 is an adaptive boosting algorithm used to improve classifier accuracy. It combines several classifiers to create a strong classifier with a high degree of accuracy. The idea underlying AdaBoost M1 is to train using the data and set the classifier weights to provide accurate predictions [20].

3.5.4 Bagging

Bagging is an ensemble ML technique that reduces overfitting and improves model performance. It is also called bootstrap aggregating. It is helpful with large datasets because it reduces the variance with the ML algorithms. Bagging is used with both classification and regression models, particularly decision tree algorithms [21].

3.5.5 Support Vector Machine (SVM)

SVM can be used to predict both continuous and discrete data. Initially, this algorithm was used for linear data classification. It was subsequently used to classify non-linear data as it provides higher accuracy with less computational effort than other methods [22]. SVM builds a hyperplane in multi-dimensional space between support vectors for classification.

Chapter 4 Performance Evaluation

Five supervised ML algorithms, namely Bayes Net, Simple Logistic, Bagging, AdaBoost M1, and SVM are evaluated in this chapter. The results were obtained using a personal computer with the hardware and software parameters listed in Table 4.1.

Item	**Parameter**
Manufacturer	HP
Model	HP 15- dy1xxx
Operating System	Windows 11 Professional 64-bit
Processor Type	Intel(R) Core (TM) i7-1065G7
Installed Memory (RAM)	16 GB
Processor Speed	2.60 GHz
Number of Cores	6
Number of Threads	12
Machine Learning Tool	WEKA version 3.9.5

Table 4-1 The hardware and software parameters.

4.1 Evaluation Metrics

The metrics used to evaluate the ML models are described below.

Precision is the ratio of true positive to the sum of true positive and false positive

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

where true positive (TP) is the number of correctly classified positive (yes) instances and false positive(FP) is the number of incorrectly classified positive instances.

Recall is the ratio of true positive to the sum of true positive and false negative

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

where false negative (FN) is the number of incorrectly classified negative (no) instances.

Accuracy is the ratio of the number of correctly predicted instances to the total number of predictions made by the model

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

where true negative (TN) is the number of correctly classified negative instances.

F-measure is the geometric mean of precision and recall

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

Execution Time is the time to train the algorithm in seconds (s).

In this chapter, accuracy, precision, recall, and F-measure are expressed as percentages.

4.2 Performance with 5-fold cross-validation using 21 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using all 21 features with 5-fold cross-validation. The results are given in Table 4.2. This shows that Bagging has the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, and 98.8, followed by SVM at 98.1, 98.2, 98.1, 98.1, Simple Logistic at 96.7, 96.2, 96.1, 96.1, Bayes Net at 94.2, 94.3, 92.4, 94.2, and AdaBoost M1 at 91.9, 92.1, 91.9, 91.9. Bayes Net had the lowest execution time at 0.02 s followed by AdaBoost M1 at 0.11 s, Bagging at 0.15 s, SVM at 0.40 s, and Simple Logistic at 0.62 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	94.2	94.3	92.4	94.2	0.02
Simple Logistic	96.7	96.2	96.1	96.1	0.62
AdaBoost M1	91.9	92.1	91.9	91.9	0.11
Bagging	98.7	98.8	98.8	98.8	0.15
SVM	98.1	98.2	98.1	98.1	0.40

Table 4-2 Performance with 5-fold cross-validation using 21 features.

4.3 Performance with 5-fold cross-validation using 15 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 15 features with 5-fold cross-validation. The results are given in Table 4.3. This shows that Bagging and SVM have the highest accuracy, precision, recall, and F-measure at 98.7, 98.7, 98.7, 98.7, and 98.7, 98.8, 98.8, and 98.8, followed by Bayes Net at 98.6, 98.6, 98.6, 98.6, and AdaBoost M1 at 94.2, 94.3, 94.2, 94.2, and Simple Logistic at 94.2, 94.3, 94.3, 94.3. Bagging had the lowest execution time at 0.01 s followed by Bayes Net at 0.06 s, AdaBoost M1 at 0.12 s, SVM at 0.22 s, and Simple Logistic at 0.43 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.6	98.6	98.6	98.6	0.06
Simple Logistic	94.2	94.3	94.3	94.3	0.43
AdaBoost M1	94.2	94.3	94.2	94.2	0.12
Bagging	98.7	98.7	98.7	98.7	0.01
SVM	98.7	98.8	98.8	98.8	0.22

Table 4-3 Performance with 5-fold cross-validation using 15 features.

4.4 Performance with 5-fold cross-validation using 10 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 10 features with 5-fold cross-validation. The results are given in Table 4.4. This shows that SVM has the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, 98.8, followed by Bayes Net at 98.6, 98.7, 98.7, 98.7, Bagging at 98.6, 98.7, 98.7, 98.7, Simple Logistic at 94.6, 94.6, 94.6, 94.6, and AdaBoost M1 at 94.1, 94.2, 94.1, 94.1. Simple Logistic had the lowest execution time at 0.06 s followed by AdaBoost M1 at 0.18 s, SVM at 0.25 s, Bayes Net at 0.37 s, and Bagging at 0.38 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.6	98.7	98.7	98.7	0.37
Simple Logistic	94.6	94.6	94.6	94.6	0.06
AdaBoost M1	94.1	94.2	94.1	94.1	0.18
Bagging	98.6	98.7	98.7	98.7	0.38
SVM	98.7	98.8	98.8	98.8	0.25

Table 4-4 Performance with 5-fold cross-validation using 10 features.

4.5 Performance with 5-fold cross-validation using 5 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 5 features with 5-fold cross-validation. The results are given in Table 4.5. This shows that Bayes Net has the highest accuracy, precision, recall, and F-measure at 98.7, 98.7, 98.6, 98.8, followed by Bagging at 98.6, 98.7, 98.7, 98.7, SVM at 98.5, 98.6, 98.6, 98.6, Simple Logistic at 94.6, 94.7, 94.5, 94.4, and AdaBoost M1 at 94.1, 94.2, 94.1, 94.1. Bayes Net had the lowest execution time at 0.01 s followed by AdaBoost M1 at 0.10 s, SVM at 0.17 s, Bagging at 0.26 s, and Simple Logistic at 0.32 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.7	98.7	98.6	98.8	0.01
Simple Logistic	94.6	94.7	94.5	94.4	0.32
AdaBoost M1	94.1	94.2	94.1	94.1	0.10
Bagging	98.6	98.7	98.7	98.7	0.26
SVM	98.5	98.6	98.6	98.6	0.17

Table 4-5 Performance with 5-fold cross-validation using 5 features.

4.6 Performance with 10-fold cross-validation using 21 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using all 21 features with 10-fold cross-validation. The results are given in Table 4.6. This shows that Bagging has the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, 98.8, followed by SVM at 98.1, 98.3, 98.2, 98.2, Simple Logistic at 95.9, 96.1, 96.2, 96.3, Bayes Net at 94.2, 94.2, 94.3, 94.4, and AdaBoost M1 at 92.1, 92.2, 92.1, 92.1, Bayes Net had the lowest execution at 0.01 s followed by AdaBoost M1 at 0.08 s, Bagging at 0.35 s, SVM at 0.47 s, and Simple Logistic at 0.53 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	94.2	94.2	94.3	94.4	0.01
Simple Logistic	95.9	96.1	96.2	96.3	0.53
AdaBoost M1	92.1	92.2	92.1	92.1	0.08
Bagging	98.7	98.8	98.8	98.8	0.35
SVM	98.1	98.3	98.2	98.2	0.47

Table 4-6 Performance with 10-fold cross-validation using 21 features.

4.7 Performance with 10-fold cross-validation using 15 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 15 features with 10-fold cross-validation. The results are given in Table 4.7. This shows that Bagging and SVM have the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, 98.9, and 98.7, 98.7, 98.8, 98.6, followed by Bayes Net at 98.6, 98.6, 98.7, 98.5, Simple Logistic at 94.5, 94.6, 94.5, 94.6, and AdaBoost M1 at 94.1, 94.2, 94.1, 94.1, AdaBoost M1 had the lowest execution at 0.01 s followed by Bayes Net at 0.05 s, SVM at 0.18 s, Bagging at 0.19 s, and Simple Logistic at 0.39 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.6	98.6	98.7	98.5	0.05
Simple Logistic	94.5	94.6	94.5	94.6	0.39
AdaBoost M1	94.1	94.2	94.1	94.1	0.01
Bagging	98.7	98.8	98.8	98.9	0.19
SVM	98.7	98.7	98.8	98.6	0.18

Table 4-7 Performance with 10-fold cross-validation using 15 features.

4.8 Performance with 10-fold cross-validation using 10 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 10 features with 10-fold cross-validation. The results are given in Table 4.8. This shows that Bagging has the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.9, 98.8, followed by SVM at 98.6, 98.8, 98.7, 98.8, Bayes Net at 98.5, 98.6, 98.7, 98.7, Simple Logistic at 94.6, 94.7, 94.7, 94.7, and AdaBoost M1 at 94.1, 94.2, 94.1, 94.1. Bayes Net had the lowest execution time at 0.04 s followed by AdaBoost M1 at 0.08 s, Bagging at 0.14 s, SVM at 0.24 s, and Simple Logistic at 0.39 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.5	98.6	98.7	98.7	0.04
Simple Logistic	94.6	94.7	94.7	94.7	0.39
AdaBoost M1	94.1	94.2	94.1	94.1	0.08
Bagging	98.7	98.8	98.9	98.8	0.14
SVM	98.6	98.8	98.7	98.8	0.24

Table 4-8 Performance with 10- fold cross-validation using 10 features.

4.9 Performance with 10-fold cross-validation using 5 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 5 features with 10-fold cross-validation. The results are given in Table 4.9. This shows that Bayes Net and Bagging have the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, 98.8, and 98.7, 98.7, 98.7, 98.7, followed by SVM at 98.6, 98.7, 98.7, 98.7, Simple Logistic at 94.6, 94.6, 94.5, 94.5, and AdaBoost M1 at 94.1, 94.2, 94.1, 94.1. Bayes Net had the lowest execution time at 0.02 s followed by AdaBoost M1 at 0.04 s, Bagging at 0.08 s, SVM at 0.17 s, and Simple Logistic at 0.36 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.7	98.8	98.8	98.8	0.02
Simple Logistic	94.6	94.6	94.5	94.5	0.36
AdaBoost M1	94.1	94.2	94.1	94.1	0.04
Bagging	98.7	98.7	98.7	98.7	0.08
SVM	98.6	98.7	98.7	98.7	0.17

Table 4-9 Performance with 10-fold cross-validation using 5 features.

4.10 Performance with a 66/34 split using 21 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using all 21 features with a 66/34 split. In this split, 64% of the data is used for training and 34% for testing. The results are given in Table 4.10. This shows that Bagging has the highest accuracy, precision, recall, and F- measure at 99.1, 99.1, 99.1, 99.1, followed by SVM at 98.2, 98.3, 98.3, 98.3, Simple Logistic at 96.1, 96.2, 96.2, 96.2, Bayes Net at 94.2, 94.3,94.3, 94.3, and AdaBoost M1 at 92.3, 92.4, 92.3, 92.3. Bayes Net had the lowest execution time at 0.01 s followed by AdaBoost M1 at 0.02 s, Bagging at 0.16 s, SVM at 0.23 s, and Simple Logistic at 0.56 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	94.2	94.3	94.3	94.3	0.01
Simple Logistic	96.1	96.2	96.2	96.2	0.56
AdaBoost M1	92.3	92.4	92.3	92.3	0.02
Bagging	99.1	99.1	99.1	99.1	0.16
SVM	98.2	98.3	98.3	98.3	0.23

Table 4-10 Performance with a 66/34 split using 21 features.

4.11 Performance with a 66/34 split using 15 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 15 features with a 66/34 split. The results are given in Table 4.11. This shows that SVM, Bayes Net, and Bagging have the highest accuracy, precision, recall, and F-measure at 99.2, 99.2, 99.2, 99.2, and 99.1, 99.1, 99.1, 99.1, and 99.1, 99.1, 99.1, 99.1, followed by Simple Logistic at 94.1, 94.0, 94.0, 94.0, and AdaBoost M1 at 93.6, 94.8, 93.7, 93.7. Bayes Net had the lowest execution time at 0.05 s, followed by AdaBoost M1 at 0.13 s, SVM at 0.17 s, Bagging at 0.19 s, and Simple Logistic at 0.44 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	99.1	99.1	99.1	99.1	0.05
Simple Logistic	94.1	94.0	94.0	94.0	0.44
AdaBoost M1	93.6	93.8	93.7	93.7	0.13
Bagging	99.1	99.1	99.1	99.1	0.19
SVM	99.2	99.2	99.2	99.2	0.17

Table 4-11 Performance with a 66/34 split using 15 features.

4.12 Performance with a 66/34 split using 10 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 10 features with a 66/34 split. The results are given in Table 4.12. This shows that Bagging has the highest accuracy, precision, recall, and F-measure at 99.3, 99.3, 99.2, 99.3, followed by SVM at 99.2, 99.1, 99.2, 99.2, Bayes Net at 99.1, 99.0, 99.1, 99.1, Simple Logistic at 93.9, 94.1, 94.0, 94.0, and AdaBoost M1 at 93.6, 93.8, 93.7, 93.7. AdaBoost M1 had the lowest at 0.11 s followed by Bayes Net at 0.04 s, Bagging at 0.10 s, SVM at 0.19 s, and Simple Logistic at 0.35 s.

.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	99.1	99.0	99.1	99.1	0.04
Simple Logistic	93.9	94.1	94.0	94.0	0.35
AdaBoost M1	93.6	93.8	93.7	93.7	0.11
Bagging	99.3	99.3	99.2	99.3	0.10
SVM	99.2	99.1	99.2	99.2	0.19

Table 4-12 Performance with a 66/34 split using 10 features.

4.13 Performance with a 66/34 split using 5 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 5 features with a 66/34 split. The results are given in Table 4.13. This shows that Bayes Net has the highest accuracy, precision, recall, and F-measure at 99.1, 99.2, 99.2, 99.2, followed by Bagging at 98.9, 99.1, 99.1, 99.1, SVM at 98.6, 98.7, 98.7, 98.7, Simple Logistic at 94.1, 94.2, 94.1, 94.1, and AdaBoost M1 at 93.6, 93.8, 93.7, 93.7. Bayes Net had the lowest execution time at 0.02 s followed by AdaBoost M1 at 0.06 s, Bagging at 0.07 s, SVM at 0.16 s, and Simple Logistic at 0.34 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	99.1	99.2	99.2	99.2	0.02
Simple Logistic	94.1	94.2	94.1	94.1	0.34
AdaBoost M1	93.6	93.8	93.7	93.7	0.06
Bagging	98.9	99.1	99.1	99.1	0.07
SVM	98.6	98.7	98.7	98.7	0.16

Table 4-13 Performance with a 66/34 split using 5 features.

4.14 Performance with a 34/66 split using 21 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using all 21 features with a 34/64 split. In this split, 34% of the data is used for training and 66% for testing. The results are given in Table 4.14. This shows that Bagging has the highest accuracy, precision, recall, and F-measure at 98.4, 98.5, 98.5,98.5, followed by SVM at 96.7, 96.8, 96.7, 96.7, Simple Logistic at 95.3, 95.3, 95.3, 95.3, Bayes Net at 93.9, 94.0, 94.0, 94.0, and AdaBoost M1 at 91.9, 92.0, 91.9, 91.9. Bayes Net had the lowest execution time at 0.01 s followed by AdaBoost M1 at 0.03 s, Bagging at 0.16 s, SVM at 0.41 s, and Simple Logistic at 0.59 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	93.9	94.0	94.0	94.0	0.01
Simple Logistic	95.3	95.3	95.3	95.3	0.59
AdaBoost M1	91.9	92.0	91.9	91.9	0.03
Bagging	98.4	98.5	98.5	98.5	0.16
SVM	96.7	96.8	96.7	96.7	0.41

Table 4-14 Performance with a 34/66 split using 21 features.

4.15 Performance with a 34/66 split using 15 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 15 features with a 34/66 split. The results are given in Table 4.15. This shows that SVM has the highest accuracy, precision, recall, and F-measure at 98.9, 99.0, 99.0, 99.0, followed by Bayes Net at 98.8, 98.9, 98.9, Bagging at 98.7, 98.8, 98.7, 98.7, Simple Logistic at 94.3, 95.3, 95.3, 95.3, and AdaBoost M1 at 94.1, 94.0, 94.0, 94.0. Bayes Net had the lowest execution time at 0.05 s followed by AdaBoost M1 at 0.11 s, Bagging at 0.10 s, Simple Logistic at 0.59 s, and SVM at 0.22 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.8	98.9	98.9	98.9	0.05
Simple Logistic	94.3	95.3	95.3	95.3	0.59
AdaBoost M1	94.1	94.0	94.0	94.0	0.11
Bagging	98.7	98.8	98.7	98.7	0.10
SVM	98.9	99.0	99.0	99.0	0.22

Table 4-15 Performance with a 34/66 split using 15 features.

4.16 Performance with a 34/66 split using 10 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 10 features with a 34/66 split. The results are given in Table 4.16. This shows that SVM and Bayes Net have the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, 98.8, and 98.7, 98.7, 98.7, 98.7, followed by Bagging at 98.5, 98.6, 98.5, 98.5, Simple Logistic at 94.1, 94.0, 94.0, 94.0, and AdaBoost M1 at 93.9, 94.0, 93.9, 93.9. Bayes Net had the lowest execution time at 0.04 s followed by AdaBoost M1at 0.08 s, Bagging at 0.12 s, SVM at 0.18 s and Simple Logistic at 0.39 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.7	98.7	98.7	98.7	0.04
Simple Logistic	94.1	94.0	94.0	94.0	0.39
AdaBoost M1	93.9	94.0	93.9	93.9	0.08
Bagging	98.5	98.6	98.5	98.5	0.12
SVM	98.7	98.8	98.8	98.8	0.18

Table 4-16 Performance with a 34/64 split using 10 features.

4.17 Performance with a 34/66 split using 5 features

To determine the performance of the models, the same dataset was used for every algorithm. Each model was trained and tested using 5 features with a 34/66 split. The results are given in Table 4.17. This shows that Bagging has the highest accuracy, precision, recall, and F-measure at 98.7, 98.8, 98.8, 98.8, followed by Bayes Net at 98.6, 98.7, 98.6, 98.6, SVM at 97.1, 97.1, 97.1, 97.1, Simple Logistic at 94.1, 94.1, 94.1, 94.1, and AdaBoost M1 at 93.9, 94.0, 93.9, 93.9. Bayes Net had the lowest execution time at 0.01 s followed by AdaBoost M1 at 0.06 s, Bagging at 0.07 s, SVM at 0.14 s, and Simple Logistic at 0.34 s.

Classifier	Accuracy	Precision	Recall	F-Measure	Execution Time (s)
Bayes Net	98.6	98.7	98.6	98.6	0.01
Simple Logistic	94.1	94.1	94.1	94.1	0.34
AdaBoost M1	93.9	94.0	93.9	93.9	0.06
Bagging	98.7	98.8	98.8	98.8	0.07
SVM	97.1	97.1	97.1	97.1	0.14

Table 4-17 Performance with a 34/66 split using 5 features.

4.18 Discussion

Bagging and SVM performed better than the other classifiers in terms of accuracy, precision, recall, and F-measure with 5-fold cross-validation, 10-fold cross-validation, and 66/34 and 34/66 splits with 15, 10, and 5 features. Bayes Net was the fastest classifier in terms of execution time, while the slowest classifiers were SVM and Simple Logistic.

In terms of 5-fold cross-validation using 21 features, Bagging has the highest accuracy at 98.7% with an execution time of 0.15 s. SVM has the second-highest accuracy at 98.1%, while the execution time of SVM was 0.40 s as shown in Table 4.2. The slowest classifier was Simple Logistic with an execution time of 0.62 s and an accuracy of 96.7%, as shown in Table 4.2. Bayes Net has better accuracy at 94.2% with an execution time of 0.02 s than AdaBoost M1 at 91.9% with an execution time of 0.11 s.

In terms of 10-fold cross-validation using 21 features, Bagging and SVM have the highest accuracy at 98.7% and 98.1% with execution times of 0.35 s and 0.47 s, respectively. Simple Logistic has an accuracy of 95.9%, while the execution time of Simple Logistic was 0.53 s which is the slowest as shown in Table 4.6. Bayes Net had better accuracy at 94.2% than AdaBoost M1 at 94.2% with an execution time of 0.01 s. The results are given in Figure 4.2 and show that 10-fold cross-validation gives better accuracy for Bayes Net and AdaBoost M1, while the accuracy for Bagging and SVM is similar for both 5-fold cross-validation and 10-fold cross-validation. Simple Logistic is the only classifier that has a higher accuracy for 5-fold cross-validation.

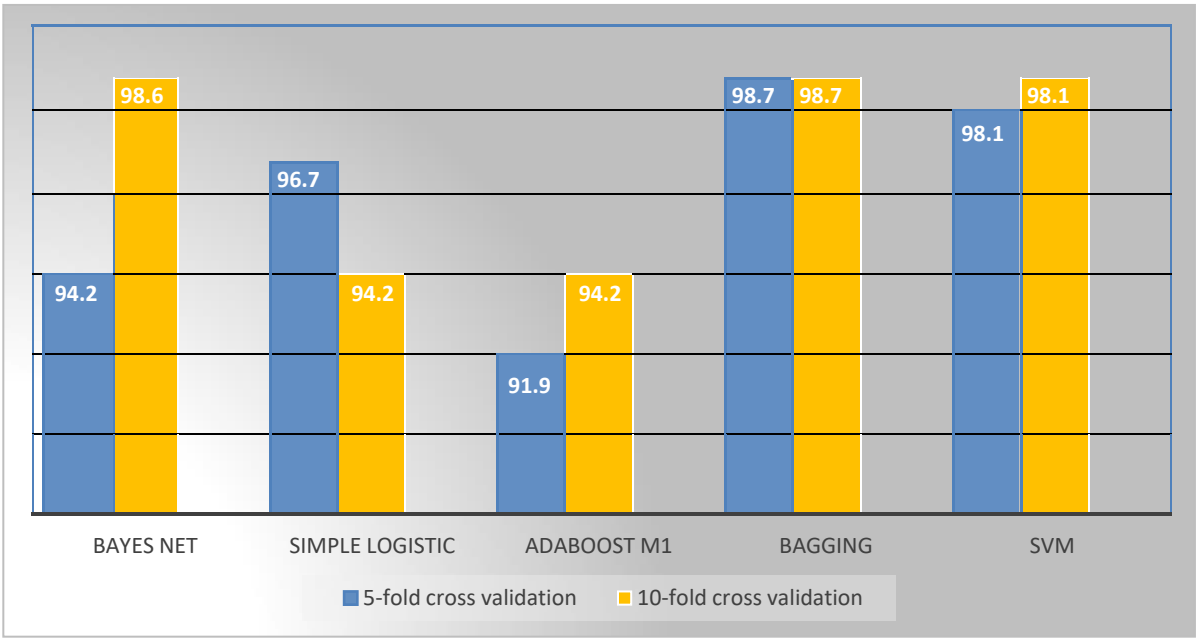


Figure 4-1 5-fold and 10-fold cross-validation accuracy with 21 features.

In terms of 5-fold cross-validation with 15 features, Bagging and SVM have the highest accuracy at 98.7% with execution times of 0.01 s and 0.22 s, respectively. Bayes Net has accuracy 98.6%, while the execution time is only 0.06 s as shown in Table 4.3. AdaBoost M1 and Simple Logistic have the same accuracy at 94.2% with execution times of 0.12 s and 0.43 s, respectively, as shown in Table 4.3.

In terms of 10-fold cross-validation with 15 features, SVM and Bagging have the highest accuracy at 98.7% and execution times 0.18 s and 0.19 s. Bayes Net has accuracy 98.6%, while the execution time of Bayes Net is 0.05 s as shown in Table 4.7. Simple Logistic was the slowest classifier with an execution time of 0.39 s and an accuracy of 94.5%. AdaBoost M1 has the lowest accuracy at 94.1% but also the lowest execution time at 0.01 s. The results are given in Figure 4.2 and show that both 5-fold and 10-fold cross-validation give similar accuracy for Bayes Net, Bagging and SVM, whereas the accuracy for Simple Logistic and AdaBoost M1 is similar for 5-fold cross validation but differs by 0.4% for 10-fold cross-validations.

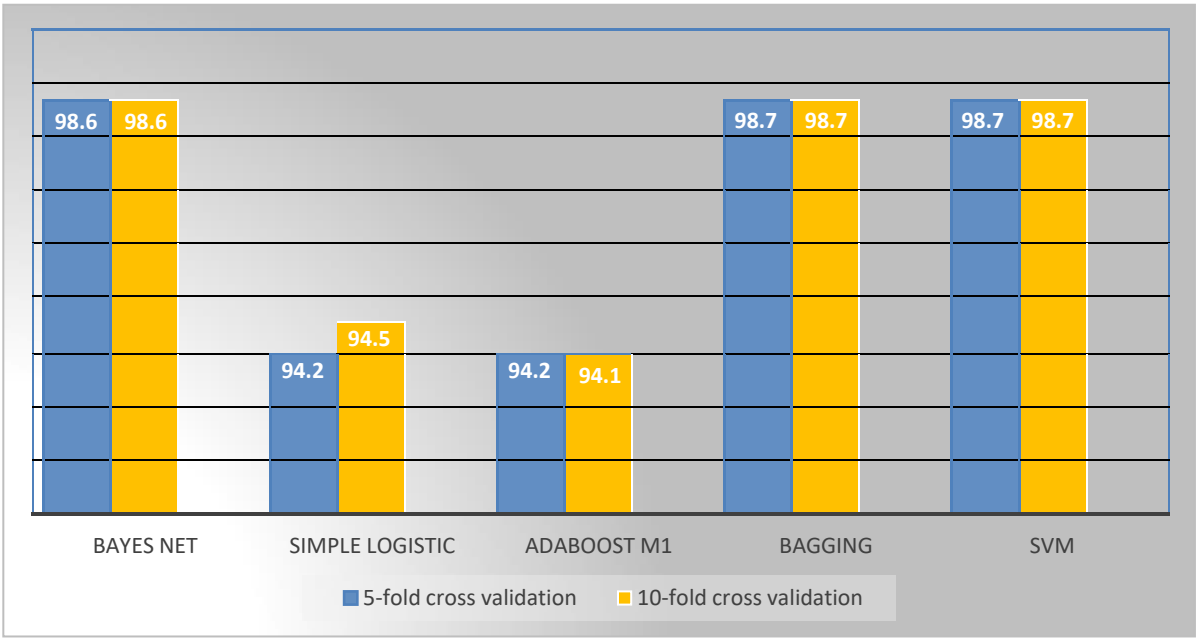


Figure 4-2 5-fold and 10-fold cross-validation accuracy with 15 features.

In terms of 5-fold cross-validation using 10 features, SVM has the highest accuracy at 98.7% with an execution time of 0.25 s. Bagging and Bayes Net have accuracy 98.6%, while the execution time of Bagging is 0.38 s which is higher than Bayes Net at 0.37 s, as shown in Table 4.4. Simple Logistic was the fastest classifier with an execution time of 0.06 s. Simple Logistic had higher accuracy at 94.6% than AdaBoost M1 at 94.1% while the execution time of AdaBoost M1 was only 0.10 s, as shown in Table 4.4.

In terms of 10-fold cross-validation using 10 features, Bagging has the highest accuracy at 98.7% with an execution time of 0.14 s. SVM has the second-highest accuracy at 98.6%, while the execution time of SVM is 0.24 s as shown in Table 4.8. Simple Logistic was the slowest classifier with an execution time of 0.39 s and an accuracy of 94.6%. Bayes Net had better accuracy at 98.5% than AdaBoost M1 at 94.1% with a higher execution time of 0.04 s as shown in Table 4.8. The results are given in Figure 4 and show that 5-fold and 10-fold cross-validation give similar accuracy for Simple Logistic, Bagging and AdaBoost M1, whereas Bayes Net and SVM have higher accuracy for 5-fold cross-validation.

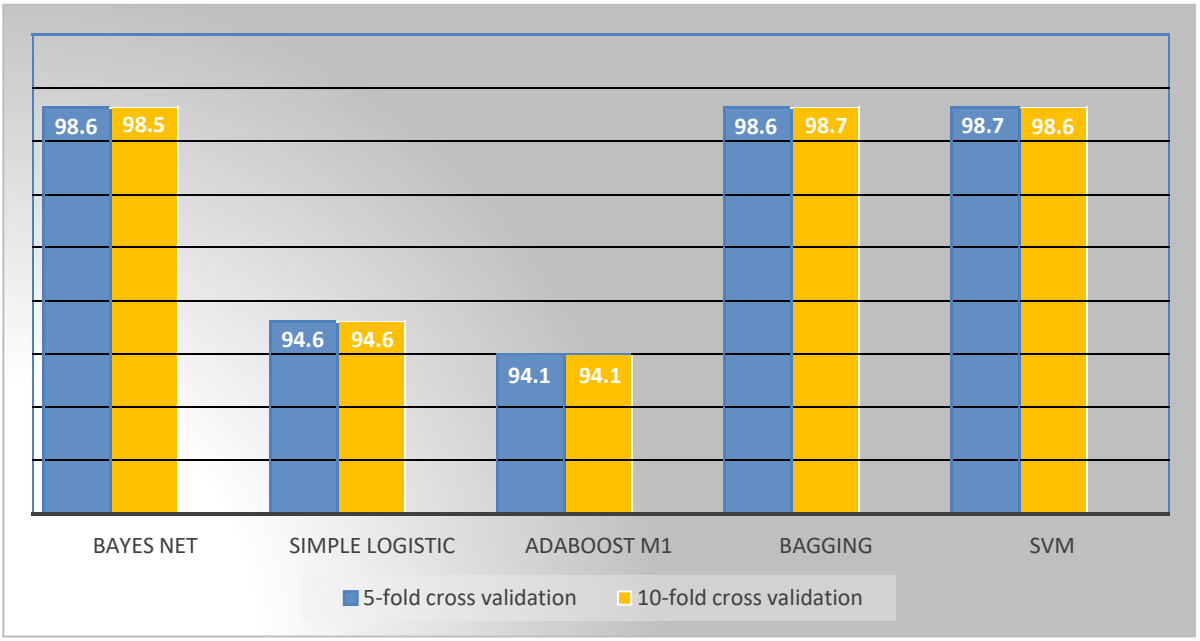


Figure 4-3 5-fold and 10-fold cross-validation accuracy with 10 features.

In terms of 5-fold cross-validation using 5 features, Bayes Net has the highest accuracy at 98.7% with an execution time of 0.01 s. Bagging has the second-highest accuracy at 98.6% while the execution time of Bagging is 0.26 s as shown in Table 4.5. Simple Logistic was the slowest classifier with an execution time of 0.32 s and an accuracy of 94.6%. SVM has higher accuracy at 98.5% than AdaBoost M1 at 94.1% with an execution time of 0.17 s as shown in Table 4.5.

In terms of 10-fold cross-validation using 5 features, Bayes Net and Bagging have the highest accuracy at 98.7% with execution times 0.02 s and 0.08 s, respectively. SVM has the second-highest accuracy of 98.6% with execution time of 0.17 s as shown in Table 4.9. Simple Logistic has better accuracy at 94.6% than AdaBoost M1 at 94.1% while Simple Logistic has an execution time of 0.36 s which is higher than AdaBoost M1 at 0.04 s. The results are given in Figure 4.4 and show that Bagging and SVM have higher accuracy for 10-fold cross-validation, whereas Bayes Net, Simple Logistic, and AdaBoost M1 have similar accuracy for 5-fold and 10-fold cross-validation.

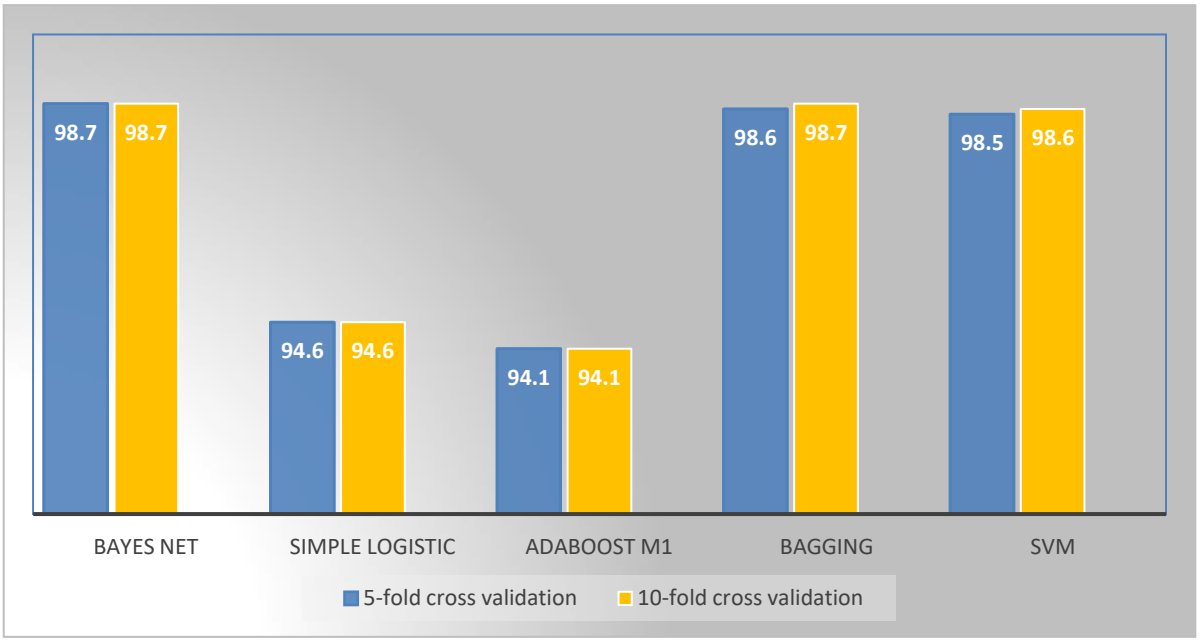
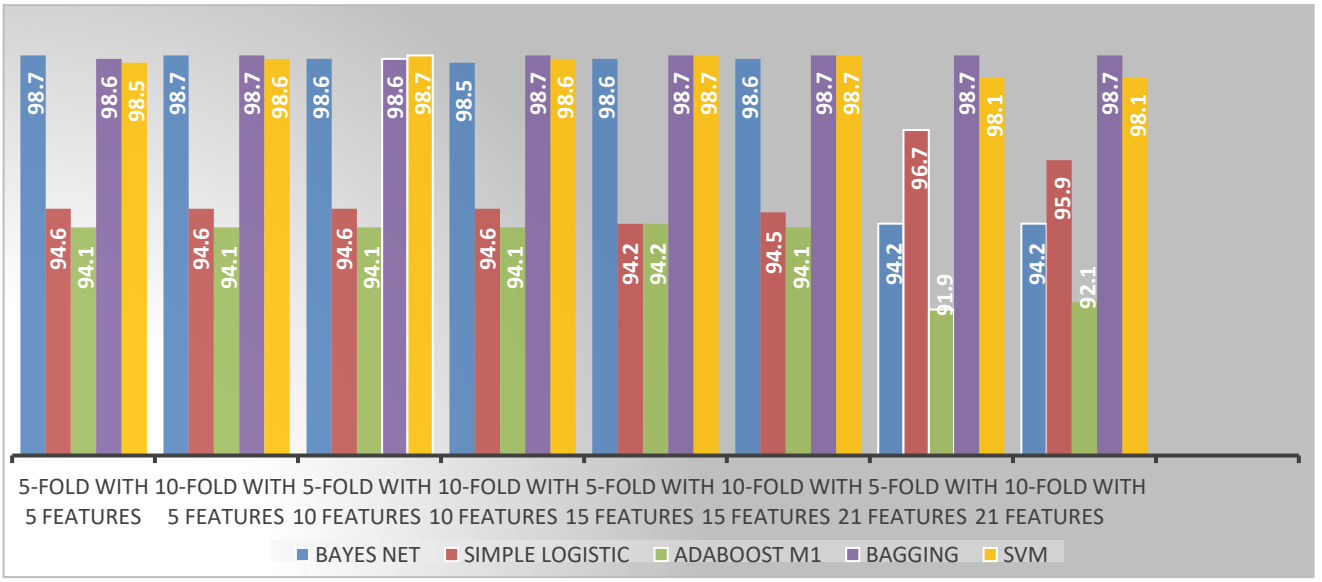


Figure 4-4 5-fold and 10-fold cross-validation accuracy with 5 features.

The results for both 5-fold and 10-fold cross-validation with 5, 10, 15, and 21 features are given in Figure 4.5. This shows that Bayes Net, Bagging, and SVM have the highest accuracy for both cross-validations, whereas Simple Logistic and AdaBoost M1 are the least accurate classifiers.



For a 66/34 split, 66% of the data is used for training and 34% for testing. For a 66/34 split with 21 features, Bagging has the highest accuracy at 99.1% with an execution time of 0.16 s. SVM has the second-highest accuracy at 98.2%, while the execution time of SVM is 0.23 s as shown in Table 4.10. The slowest classifier was Simple Logistic with an execution time of 0.56 s and an accuracy of 96.1%. Bayes Net had better accuracy at 94.2% and a higher execution time of 0.01 s than AdaBoost M1 with accuracy 91.9% and execution time 0.02 s as shown in Table 4.10.

For a 34/66 split, 34% of the data is used for training and 66% for testing. For a 34/64 split with 21 features, Bagging has the highest accuracy at 98.4% with an execution time of 0.16 s. SVM has the second-highest accuracy at 96.7% while the execution time is 0.41 s as shown in Table 4.14. Simple Logistic was the slowest classifier with an execution time of 0.59 s and an accuracy of 95.3%. Bayes Net has an accuracyof 93.9% which is better than AdaBoost M1 at 91.9% and an execution time of 0.01 s. The results are given in Figure 4.5 and show that a 66/34 split gives higher accuracy than a 34/64 split for all five classifiers.

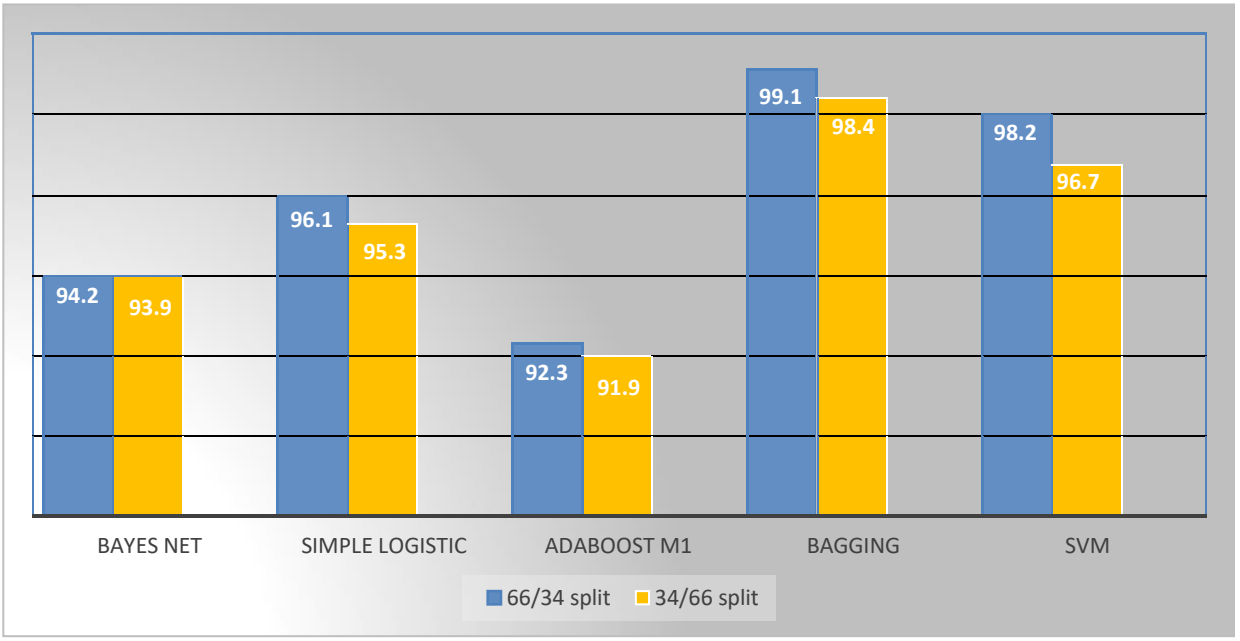


Figure 4-6 66/34 and 34/66 split accuracy with 21 features.

For a 66/34 split using 15 features, SVM has the highest accuracy at 99.2% with an execution time of 0.17 s. Bayes Net and Bagging have the same accuracy at 99.1% and execution times 0.05 s and 0.19 s, respectively. The slowest classifier was Simple Logistic with an execution time of 0.44 s and an accuracy of 94.1%, followed by AdaBoost M1 with accuracy 93.6% and execution time 0.13 s as shown in Table 4.11.

For a 34/66 split using 15 features, SVM has the highest accuracy at 98.9% with an execution time of 0.22 s. Bayes Net has the second-highest accuracy at 98.8% with an execution time of 0.05 s as shown in Table 4.15. Simple Logistic was the slowest classifier with an execution time of 0.59 s and an accuracy of 94.3%. Bagging with accuracy 98.7% and execution time 0.10 s is better than AdaBoost M1 at 94.1% with an execution time of 0.11 s. The results are given in Figure 4.6 and show that a 66/34 split gives higher accuracy for Bayes Net, Bagging, and SVM, whereas the accuracy for Simple Logistic and AdaBoost M1 is lower with a 66/34 split.

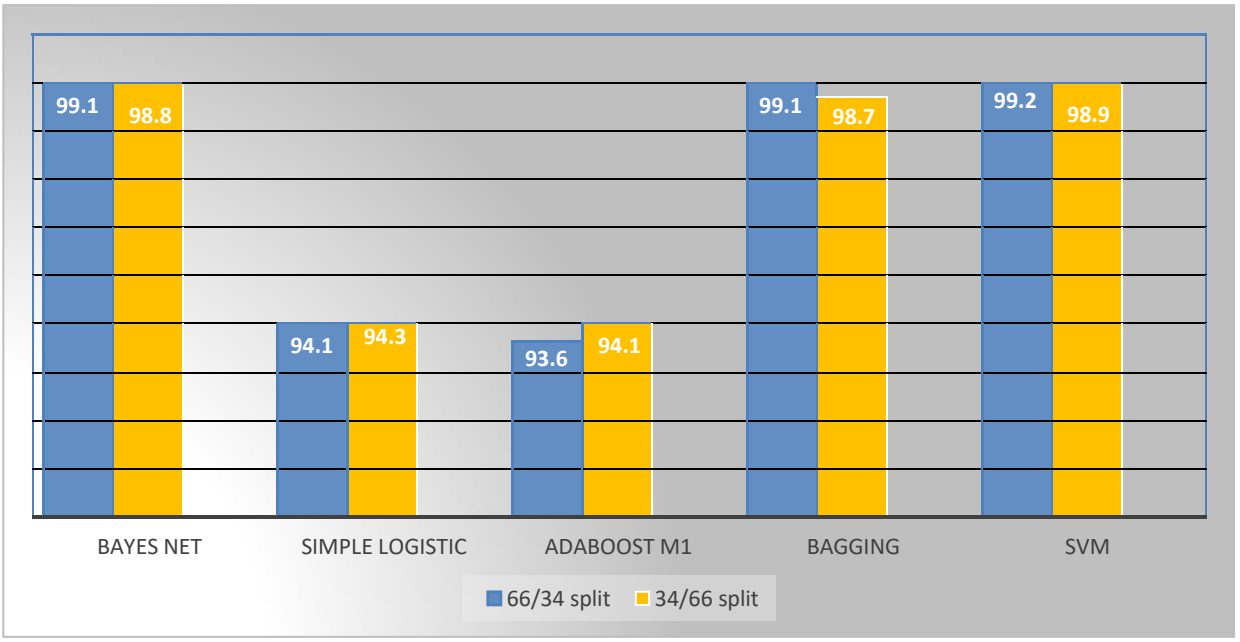


Figure 4-7 66/34 and 34/66 split accuracy with 15 features.

For a 66/34 split using 10 features, Bagging has the highest accuracy at 99.3% and an execution time of 0.10 s. SVM has the second-highest accuracy at 99.2% with an execution time of 0.19 s as shown in Table 4.12. The slowest classifier was Simple Logistic with an execution time of 0.35 s and an accuracy of 93.9%. Bayes Net accuracy of 99.1% and execution time 0.04 s is higher than AdaBoost M1 with an accuracy of 93.6% and execution time 0.11 s as shown in Table 4.12.

For a 34/66 split with 10 features, Bayes Net and SVM have the highest accuracy at 98.7% and execution times 0.04 s and 0.18 s, respectively. Bagging has the third-highest accuracy at 98.5% with an execution time of 0.12 s as shown in Table 4.16. Simple Logistic was the slowest classifier with an execution time of 0.39 s and an accuracy of 94.1% followed by AdaBoost M1 with an accuracy of 93.9% and an execution time of 0.08 s as shownin Table 4.16. The results are given in Figure 4.7 and show that Bayes Net, Bagging, and SVM have higher accuracy with a 66/34 split, whereas Simple Logistic and AdaBoost M1 accuracy is lower with a 66/34 split.

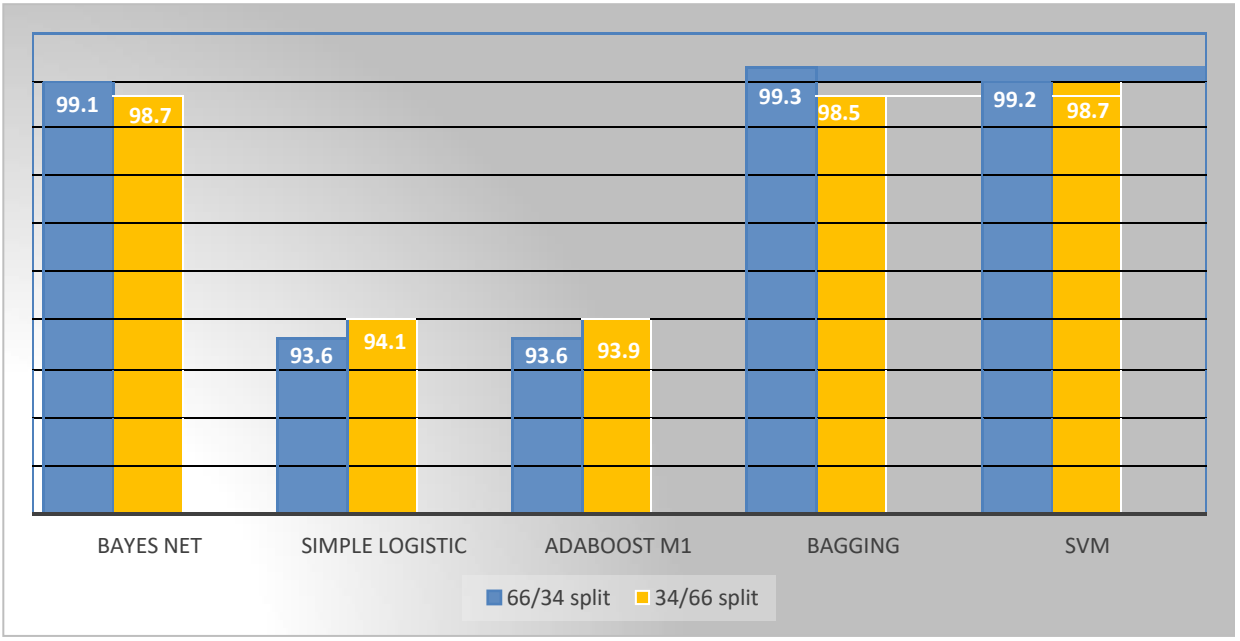


Figure 4-8 66/34 and 34/66 split accuracy with 10 features.

For a 66/34 split using 5 features, Bayes Net has the highest accuracy at 99.1% and an execution time of 0.02 s. Bagging has the second-highest accuracy at 98.9% with an execution time of 0.07 s as shown in Table 4.13. The slowest classifier was Simple Logistic with an execution time of 0.34 s and an accuracy of 94.1%. SVM has higher accuracy at 98.6% than AdaBoost M1 at 98.6%, with an execution time of 0.16 s, as shown in Table 4.13.

For a 34/66 split using 5 features, Bagging has the highest accuracy at 98.7% and an execution time at 0.07 s. Bayes Net has the second-highest accuracy at 98.6% with an execution time of 0.01 s as shown in Table 4.17. Simple Logistic was the slowest classifier with an execution time of 0.34 s and an accuracy of 94.1%. SVM has higher accuracy at 97.1% than AdaBoost M1 at 93.9%, with an execution time of 0.14 s. The results are given in Figure 4.8 and show that a 66/34 split gives higher accuracy for Bayes Net, Bagging, and SVM, whereas the accuracy for Simple Logistic and AdaBoost differs by 0.5% for a 34/66 split and 0.8% for a 66/34 split.

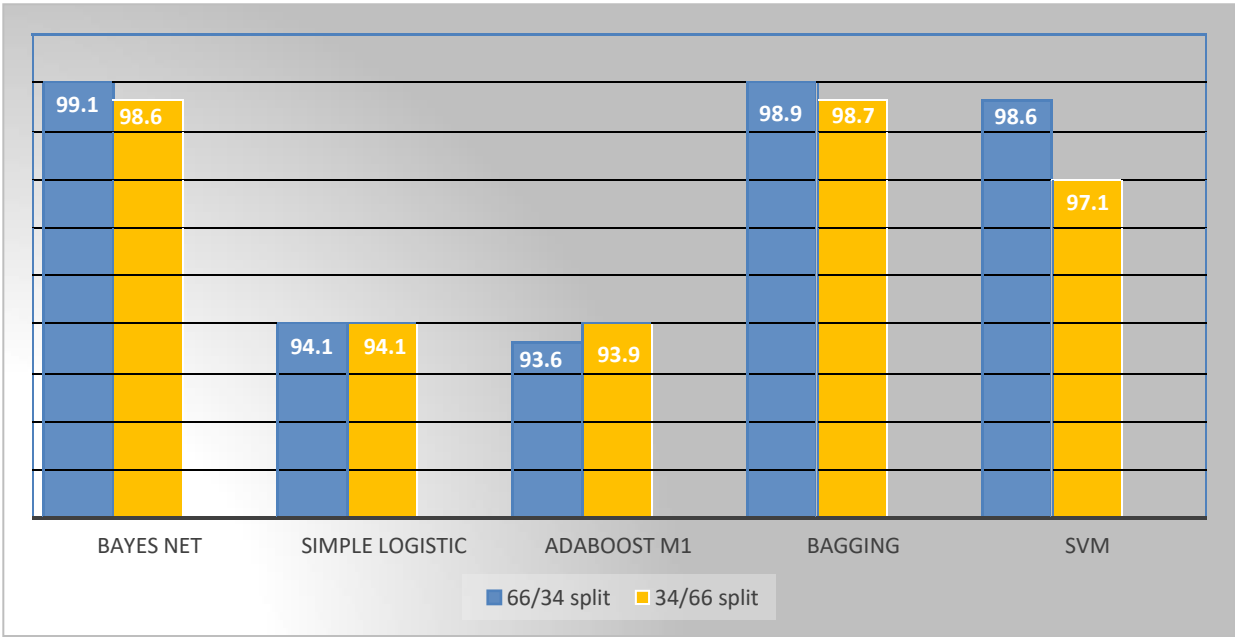


Figure 4-9 66/34 and 34/66 split accuracy with 5 features.

By comparing 66/34 and 34/64 split accuracy using 5, 10, 15, and 21 features, it is observed that a 66/34 split gives higher accuracy for all five classifiers. Simple Logistic and AdaBoost M1 are the least accurate classifiers for both splits.

Chapter 5 Conclusion and Future Work

This project considered COVID-19 prediction using five supervised machine learning algorithms, namely Bayes Net, AdaBoost M1, Bagging, SVM, and Simple Logistic. The performance of each model was evaluated based on accuracy, precision, recall, F-measure, and execution time. SMOTE was used to balance the dataset and PCA used for feature reduction based on the eigenvalues. This reduced the 21 features to 15, 10, and 5 features. WEKA was used to evaluate the performance of the models using 5-fold and 10-fold cross-validation, and 66/34 and 34/66 percentage splits. The results obtained show that Bagging outperforms the other classifiers with an accuracy of 99.3% using a 66/34 percentage split with 10 features. Further, Bayes Net performs better than the other classifiers in terms of execution time. SVM is the second-best algorithm as it has almost the same accuracy as Bagging and a lower execution time. Bayes Net is ranked third, Simple Logistic fourth, and AdaBoost M1 fifth. Overall, a 66/34 percentage split provides better performance than 5-fold and 10-fold cross-validation, and a 34/66 split.

For future work, this system can be employed with datasets for other diseases. It is also possible to develop a model that can determine the probable severity of COVID-19. This would provide further information regarding important steps to take and other therapies that may be employed. Further, unsupervised ML algorithms can be considered.

Bibliography

[1] T. Singhal, A Review of Coronavirus Disease COVID-19, The Indian Journal of Pediatrics, vol. 87, no. 4, pp. 281-286, 2020.

[2] C. Domenico and M. Vanelli, WHO Declares COVID-19 a Pandemic, vol. 91, no.1, pp. 157-160, 2020.

[3] M. N. Temgoua, F. T. Endomba, J. R. Nkeck, G. U. Kenfack, J. N. Techie, and M. Essouma, Coronavirus Disease 2019 (COVID-19) as a Multi-Systemic Disease and Its Impact in Low-and Middle-Income Countries (LMICs), SN Comprehensive Clinical Medicine, vol. 2, no. 9, pp. 1377-1387, 2020.

[4] H. Ames, How Long Does Coronavirus Last in The Body, Air, and in Food, October 2020, https://www.medicalnewstoday.com/articles/how-long-does-coronavirus- last/.

[5] I. Kapoor, H. Prabhakar, and C. Mahajan, Introduction: History of Coronavirus Disease Pandemic, Clinical Synopsis of COVID-19, pp. 1-4, India, 2020.

[6] M. A. Arshid, M. Mumtaz, and R. Nazir, Unforeseen Challenges to Global Health System, in Particular Context to COVID-19 Pandemic and Health Care Personnel, Arab Journal of Basic and Applied Sciences, vol. 28, no.1, pp. 145-153, 2021.

[7] D. Vasireddy, R, Vanaparthy, G. Mohan, Rachana, S. V. Malayala, and P. Atluri Review of COVID-19 Variants and COVID-19 Vaccine Efficacy, What the Clinician Should Know, Journal of Clinical Medicine Research, vol, 13, no. 6, pp. 317–325, 2021.

[8] H. Turabieh and W. B. A. Karaa, Predicting the Existence of COVID-19 Using Machine Learning Based on Laboratory Findings, International Conference of Women in Data Science, Saudi Arabia, 2021.

[9] J. Luo, L. Zhou, Y. Feng, B. Li, and S. Guo, The Selection of Indicators from Initial

Blood Routine Test Results to Improve the Accuracy of Early Prediction of COVID-19 Severity, PLoS ONE, vol. 16, no. 6, art. no. e0253329, 2021.

[10] L.Yan, H. Zhang, J. Goncalves, Y. Xiao, M. Wang, Y. Guo, C. Sun, X. Tang, L. Jing, M. Zhang, X. Huang, Y. Xiao, H. Cao, Y. Chen, T. Ren, F. Wang, Y. Xiao, S. Huang, X. Tan, N. Huang, B. Jiao, C. Cheng, Y. Zhang, A. Luo, L. Mombaerts, J. Jin, Z. Cao, S. Li, H. Xu and Y. Yuan, An Interpretable Mortality Prediction Model For COVID-19 Patients, Nature Machine Intelligence, vol. 2, no. 5, pp. 283-288, 2020.

[11] S. Khalilpourazari and H. H. Doulabi, Robust Modelling and Predication of the COVID-19 Pandemic in Canada, International Journal of Production Research, pp. 1-17, 2021.

[12] P. Majumder, Daily Confirmed Cases and Deaths Prediction of Novel Coronavirus in Asian Continent Polynomial Neural Network, Biomedical Engineering Tools for Management for Patients with COVID-19, pp. 163-172, 2021.

[13] G. Holmes, A. Donkin, and I. H. Witten, WEKA, A Machine Learning Workbench, Proceedings of Australian New Zealand Intelligent Information Systems Conference, pp. 357- 361, Australia, 1994.

[14] IBM Cloud Education, What is Supervised Learning, 2020. https:// www.ibm.com/cloud/learn/supervised-learning.

[15] B. Mahesh, Machine Learning Algorithms a Review, International Journal of Science and Research, vol. 9, pp. 381-386, 2019.

[16] H. Harikrishnan, Symptoms and COVID Presence, 2020. https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid- presence/.

[17] M. Ringner, What is Principal Component Analysis, Nature Biotechnology, vol. 26, no. 3, pp. 303-304, 2008.

[18] C. Sammut and G. I. Webb, Encyclopedia of Machine Learning and

Data Mining, Bayesian Network, pp. 106-107, 2017.

[19] R. N. Sucky, Complete Details of Simple Logistic Regression Model and Inference in R, 2021. https://medium.com/codex/complete-details-of-simple-logistic-regression-model-and-inference-in-r-eedb1c84b65f.

[20] Z. Zhang and X. Xie, Research on AdaBoost.M1 with Random Forest, 2nd International Conference on Computer Engineering and Technology, v.1, pp. 647-652, China, 2010.

[21] T. G. Dietterich, Ensemble Methods in Machine Learning, In Multiple Classifier Systems, Lecture Notes in Computer Science, vol 1857, pp. 1-15, 2000.

[22] R. Pupale, Support Vector Machines SVM An Overview, 2018, https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989/.

[23] K. Oksuz, B. C. Cam, S. Kalkan, and E. Akbas, A. Emre, Imbalance Problems in Object Detection: A Review, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3388-3415, 2021.