**Comparison of Machine Learning Methods for Breast Cancer Diagnosis**

**1. INTRODUCTION**

Cancer is the second reason of human death all over the world and accounts for roughly 9.6 million deaths in 2018. Globally, for 1 human death in 6 can be said that is caused by cancer. Almost 70 percent of the deaths from cancer disease happen in countries that have low and middle income. The most common cancer type among women are breast, lung and colorectal, which totally symbolize half of the all cancer cases. Also, breast cancer is responsible for the thirty percent of all new cancer diagnoses in women. Machine learning (ML) methods ensure analyzing the data and extracting key characteristics of relationships and information from dataset. Also, it creates a computational model for best description of the data. Especially, according to in researches about cancer disease, it can be said that ML techniques can be handled on early detection and prognosis of cancer. Asri et al. have compared some machine learning algorithms for the risk prediction and diagnosis of breast cancer. Support Vector Machine (SVM), k-Nearest Neighbors (kNN), Naive Bayes (NB) and Decision Tree (C4.5) have been applied Wisconsin Breast Cancer (Original) dataset. SVM classification method has been given the highest accuracy value (97.13 %) with least error rate when the experimental results were compared. Bazazeh and Shubair have investigated the comparative study of machine learning techniques as Support Vector Machine (SVM), Random Forest (RF) and Bayesian Network (BN) for detection and diagnosis of breast cancer. The Original Wisconsin Breast Cancer was used as a dataset and Weka software was used as a Machine Learning tool. The key performance parameters of machine learning classifiers have been compared according to accuracy, recall, precision and ROC area. They have suggested that BN has the best performance according to recall and precision values and RF technique has optimum performance in term of ROC area. Ahmad et al. have exercised machine learning algorithms for predicting the rate of two years recurrence of breast cancer disease. The dataset has been obtained from Iranian Center of Breast Cancer (ICBC) program, collected the time period of 1997-2008 years. The dataset is consisted of population characteristics and 22 input variables also the cases have been collected from 1189 women of diagnosed breast cancer. Artificial Neural Network (ANN), Support Vector Machine (SVM) and Decision Tree (DT) have been applied and SVM has been showed the best performance with highest accuracy and least error rate. Bektas and Babur have studied on diagnosis of breast cancer using machine learning techniques. Kent Ridge Microarray has been used 2 datasets and support vector machine, k-star, random forest algorithm and voted perceptron have been applied. Random forest algorithm has been showed more performance than applied feature selection method. Chen et al. have applied Support Vector Machine classification algorithm on Wisconsin Diagnostic Breast Cancer dataset. In the study, the training and testing sets have been split as 50-50%, 70-30% and 80-20%. According to different training/testing percent, accuracy values have been calculated.

**1.1 Objective of the Project**

Cancer is the second reason of human death all over the world and accounts for roughly 9.6 million deaths in 2018. The most common cancer type among women are breast, lung and colorectal, which totally symbolize half of the all cancer cases. Also, breast cancer is responsible for the thirty percent of all new cancer diagnoses in women. Machine learning (ML) methods ensure analyzing the data and extracting key characteristics of relationships and information from dataset. In this paper, two of the most popular machine learning techniques have been used for classification of Wisconsin Breast Cancer (Original) dataset and the classification performance of these techniques have been compared with each other using the values of accuracy, precision, recall and ROC Area.

**2. LITERATURE SURVEY**

**Machine learning for improved diagnosis and prognosis in healthcare**

Machine learning has gained tremendous interest in the last decade fueled by cheaper computing power and inexpensive memory – making it efficient to store, process and analyze growing volumes of data. Enhanced algorithms are being designed and applied on large datasets to help discover hidden insights and correlations amongst data elements not obvious to human. These insights help businesses take better decisions and optimize key indicators of interest. The growing popularity of machine learning also stems from the fact that learning algorithms are agnostic to the domain of application. Classification algorithms, for example, that could be applied to categorize faults in windmill blades can also be used for categorizing TV viewers in a survey. The actual value of machine learning however depends on the ability to adapt and apply these algorithms to solve specific real world problems. In this paper we discuss two such applications for interpreting medical data for automated analysis. Our first case study demonstrates the use of Bayesian Inference, a paradigm of machine learning, for diagnosing Alzheimer’s disease based on cognitive test results and demographic data. In the second case study we focus on automated classification of cell images to determine the advancement and severity of breast cancer using artificial neural networks. Although these research are still preliminary, they demonstrate the value of machine learning techniques in providing quick, efficient and automated data analysis. Machine learning offers hope with early diagnosis of diseases, help patients in making informed decisions on treatment options and can help in improving overall quality of their lives.

**Cancer statistics**

Each year, the American Cancer Society estimates the numbers of new cancer cases and deaths that will occur in the United States and compiles the most recent data on cancer incidence, mortality, and survival. Incidence data, available through 2014, were collected by the Surveillance, Epidemiology, and End Results Program; the National Program of Cancer Registries; and the North American Association of Central Cancer Registries. Mortality data, available through 2015, were collected by the National Center for Health Statistics. In 2018, 1,735,350 new cancer cases and 609,640 cancer deaths are projected to occur in the United States. Over the past decade of data, the cancer incidence rate (2005-2014) was stable in women and declined by approximately 2% annually in men, while the cancer death rate (2006-2015) declined by about 1.5% annually in both men and women. The combined cancer death rate dropped continuously from 1991 to 2015 by a total of 26%, translating to approximately 2,378,600 fewer cancer deaths than would have been expected if death rates had remained at their peak. Of the 10 leading causes of death, only cancer declined from 2014 to 2015. In 2015, the cancer death rate was 14% higher in non-Hispanic blacks (NHBs) than non-Hispanic whites (NHWs) overall (death rate ratio [DRR], 1.14; 95% confidence interval [95% CI], 1.13-1.15), but the racial disparity was much larger for individuals aged <65 years (DRR, 1.31; 95% CI, 1.29-1.32) compared with those aged ≥65 years (DRR, 1.07; 95% CI, 1.06-1.09) and varied substantially by state. For example, the cancer death rate was lower in NHBs than NHWs in Massachusetts for all ages and in New York for individuals aged ≥65 years, whereas for those aged <65 years, it was 3 times higher in NHBs in the District of Columbia (DRR, 2.89; 95% CI, 2.16-3.91) and about 50% higher in Wisconsin (DRR, 1.78; 95% CI, 1.56-2.02), Kansas (DRR, 1.51; 95% CI, 1.25-1.81), Louisiana (DRR, 1.49; 95% CI, 1.38-1.60), Illinois (DRR, 1.48; 95% CI, 1.39-1.57), and California (DRR, 1.45; 95% CI, 1.38-1.54). Larger racial inequalities in young and middle-aged adults probably partly reflect less access to high-quality health care. CA Cancer J Clin 2018;68:7-30.

**Using machine learning algorithms for breast cancer risk prediction and diagnosis.**

Breast cancer represents one of the diseases that make a high number of deaths every year. It is the most common type of all cancers and the main cause of women's deaths worldwide. Classification and data mining methods are an effective way to classify data. Especially in medical field, where those methods are widely used in diagnosis and analysis to make decisions. In this paper, a performance comparison between different machine learning algorithms: Support Vector Machine (SVM), Decision Tree (C4.5), Naive Bayes (NB) and k Nearest Neighbors (k-NN) on the Wisconsin Breast Cancer (original) datasets is conducted. The main objective is to assess the correctness in classifying data with respect to efficiency and effectiveness of each algorithm in terms of accuracy, precision, sensitivity and specificity. Experimental results show that SVM gives the highest accuracy (97.13%) with lowest error rate. All experiments are executed within a simulation environment and conducted in WEKA data mining tool.

**Comparative study of machine learning algorithms for breast cancer detection and diagnosis**

Breast cancer is one of the most widespread diseases among women in the UAE and worldwide. Correct and early diagnosis is an extremely important step in rehabilitation and treatment. However, it is not an easy one due to several uncertainties in detection using mammograms. Machine Learning (ML) techniques can be used to develop tools for physicians that can be used as an effective mechanism for early detection and diagnosis of breast cancer which will greatly enhance the survival rate of patients. This paper compares three of the most popular ML techniques commonly used for breast cancer detection and diagnosis, namely Support Vector Machine (SVM), Random Forest (RF) and Bayesian Networks (BN). The Wisconsin original breast cancer data set was used as a training set to evaluate and compare the performance of the three ML classifiers in terms of key parameters such as accuracy, recall, precision and area of ROC. The results obtained in this paper provide an overview of the state of art ML techniques for breast cancer detection.

**Using three machine learning techniques for predicting breast cancer recurrence.**

Abstract Objective: The number and size of medical databases are increasing rapidly but most of these data are not analyzed for finding the valuable and hidden knowledge. Advanced data mining techniques can be used to discover hidden patterns and relationships. Models developed from these techniques are useful for medical practitioners to make right decisions. The present research studied the application of data mining techniques to develop predictive models for breast cancer recurrence in patients who were followed-up for two years. Method: The patients were registered in the Iranian Center for Breast Cancer (ICBC) program from 1997 to 2008. The dataset contained 1189 records, 22 predictor variables, and one outcome variable. We implemented machine learning techniques, i.e., Decision Tree (C4.5), Support Vector Machine (SVM), and Artificial Neural Network (ANN) to develop the predictive models. The main goal of this paper is to compare the performance of these three well-known algorithms on our data through sensitivity, specificity, and accuracy. Results and Conclusion: Our analysis shows that accuracy of DT, ANN and SVM are 0.936, 0.947 and 0.957 respectively. The SVM classification model predicts breast cancer recurrence with least error rate and highest accuracy. The predicted accuracy of the DT model is the lowest of all. The results are achieved using 10-fold cross-validation for measuring the unbiased prediction accuracy of each model.

**Machine learning based performance development for diagnosis of breast cancer**

Breast cancer is prevalent among women and develops from breast tissue. Early diagnosis and accurate treatment is vital to increase the rate of survival. Identification of genetic factors with microarray technology can make significant contributions to diagnosis and treatment process. In this study, several machine learning algorithms are used for Diagnosis of Breast Cancer and their classification performances are compared with each other. In addition, the active genes in breast cancer are identified by attribute selection methods and the conducted study show success rate 90,72 % with 139 feature.

**A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis.**

Breast cancer is becoming a leading cause of death among women in the whole world, meanwhile, it is confirmed that the early detection and accurate diagnosis of this disease can ensure a long survival of the patients. Expert systems and machine learning techniques are gaining popularity in this field because of the effective classification and high diagnostic capability. In this paper, a rough set (RS) based supporting vector machine classifier (RS\_SVM) is proposed for breast cancer diagnosis. In the proposed method (RS\_SVM), RS reduction algorithm is employed as a feature selection tool to remove the redundant features and further improve the diagnostic accuracy by SVM. The effectiveness of the RS\_SVM is examined on Wisconsin Breast Cancer Dataset (WBCD) using classification accuracy, sensitivity, specificity, confusion matrix and receiver operating characteristic (ROC) curves. Experimental results demonstrate the proposed RS\_SVM can not only achieve very high classification accuracy but also detect a combination of five informative features, which can give an important clue to the physicians for breast diagnosis.

**SVM and SVM ensembles in breast cancer prediction**

Breast cancer is an all too common disease in women, making how to effectively predict it an active research problem. A number of statistical and machine learning techniques have been employed to develop various breast cancer prediction models. Among them, support vector machines (SVM) have been shown to outperform many related techniques. To construct the SVM classifier, it is first necessary to decide the kernel function, and different kernel functions can result in different prediction performance. However, there have been very few studies focused on examining the prediction performances of SVM based on different kernel functions. Moreover, it is unknown whether SVM classifier ensembles which have been proposed to improve the performance of single classifiers can outperform single SVM classifiers in terms of breast cancer prediction. Therefore, the aim of this paper is to fully assess the prediction performance of SVM and SVM ensembles over small and large scale breast cancer datasets. The classification accuracy, ROC, F-measure, and computational times of training SVM and SVM ensembles are compared. The experimental results show that linear kernel based SVM ensembles based on the bagging method and RBF kernel based SVM ensembles with the boosting method can be the better choices for a small scale dataset, where feature selection should be performed in the data pre-processing stage. For a large scale dataset, RBF kernel based SVM ensembles based on boosting perform better than the other classifiers.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

Breast Cancer is the most frequent disease as a cancer type for women. There is no existing systems can give a better performance on the available data.

**Disadvantages of Existing System:**

1. Performance is less.

**3.2 Proposed System**

In this paper, we have discussed two popular machine learning techniques for Wisconsin Breast Cancer classification. Artificial Neural Network and Support Vector Machine are used as ML techniques for the classification of WBC (Original) dataset in WEKA tool. The effectiveness of applied ML techniques is compared in term of key performance metrics such as accuracy, precision, recall and ROC area. Based on the performance metrics of the applied ML techniques, SVM (Sequential Minimal Optimization Algorithm) has showed the best performance in the accuracy of 96,9957 % for the diagnosis and prediction from WBC dataset.

**Advantages:**

1. Performance is best.

2. Accuracy is more.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loa ded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms *what* must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**Operating Environment**

Windows XP.

**HARDWARE REQUIREMENTS:**

# Processor - Pentium –IV

* Speed - 1.1 Ghz
* RAM - 256 MB(min)
* Hard Disk - 20 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows7/8
* Programming Language - Python

**4. SYSTEM DESIGN**

**UML Diagram:**

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

**Class Diagram:**



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

**Use case Diagram:**

Run VM Algorithm

Run ANN Algorithm

Upload test data & predict disease

User

Generate Train &Test Model

Accuracy Graph

Upload Wisconsin Dataset

**Sequence diagram:**

A **sequence diagram** is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behavior of a system.



**Component Diagram:**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.

**Component Diagram:**

User

Upload Wisconsin

Generate train & test Model

Run SVM

Algorithm

Run ANN

Algorithm

Upload test data &

Predict disease

Dataset

Accuracy Graph

**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

**Deployment Diagram:**

Upload Wisconsin

Dataset

Accuracy Graph

Patient

Generate train

& test model

Run SVM

Algorithm

Run ANN Algorithm

Upload test&

predict disease

**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

Activity Diagram:

Upload Wisconsin dataset

Generate train & test model

Run SVM Algorithm

Run ANN Algorithm

Upload test data & predict disease

Accuracy Graph

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

2. Dataset loaded

User

4. Split dataset into train& test

6. Training model generated

1. Upload Wisconsin dataset 8. Get accuracy

3. Generate train & test model 10. Get predict result

5. Run SVM Algorithm 12. Accuracy graph is displayed

7. Run ANN Algorithm

9. Upload test data & predict disease

11. Accuracy Graph

**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

**CancerPrediction.py:**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

import numpy as np

import pandas as pd

from sklearn import \*

from sklearn.metrics import confusion\_matrix

from sklearn.model\_selection import train\_test\_split

from sklearn import svm

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

from sklearn.ensemble import RandomForestClassifier

from keras.models import Sequential

from keras.layers import Dense

main = tkinter.Tk()

main.title("Machine Learning Methods Comparison") #designing main screen

main.geometry("1300x1200")

global filename

global cls

global X, Y, X\_train, X\_test, y\_train, y\_test

global svm\_acc, ann\_acc # all global variables names define in above lines

def traintest(train): #method to generate test and train data from dataset

X = train.values[:, 0:9]

Y = train.values[:, 10]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, Y, test\_size = 0.3, random\_state = 0)

return X, Y, X\_train, X\_test, y\_train, y\_test

def generateModel(): #method to read dataset values which contains all five features data

global X, Y, X\_train, X\_test, y\_train, y\_test

train = pd.read\_csv(filename)

X, Y, X\_train, X\_test, y\_train, y\_test = traintest(train)

text.insert(END,"Train & Test Model Generated\n\n")

text.insert(END,"Total Dataset Size : "+str(len(train))+"\n")

text.insert(END,"Split Training Size : "+str(len(X\_train))+"\n")

text.insert(END,"Split Test Size : "+str(len(X\_test))+"\n")

def upload(): #function to upload tweeter profile

global filename

filename = filedialog.askopenfilename(initialdir="dataset")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n");

def prediction(X\_test, cls): #prediction done here

y\_pred = cls.predict(X\_test)

for i in range(len(X\_test)):

print("X=%s, Predicted=%s" % (X\_test[i], y\_pred[i]))

return y\_pred

# Function to calculate accuracy

def cal\_accuracy(y\_test, y\_pred, details):

cm = confusion\_matrix(y\_test, y\_pred)

accuracy = accuracy\_score(y\_test,y\_pred)\*100

text.insert(END,details+"\n\n")

text.insert(END,"Accuracy : "+str(accuracy)+"\n\n")

text.insert(END,"Report : "+str(classification\_report(y\_test, y\_pred))+"\n")

text.insert(END,"Confusion Matrix : "+str(cm)+"\n\n\n\n\n")

return accuracy

def runSVM():

global svm\_acc

global cls

global X, Y, X\_train, X\_test, y\_train, y\_test

cls = svm.SVC(kernel='rbf',

class\_weight='balanced', # penalize

probability=True)

cls.fit(X\_train, y\_train)

text.insert(END,"Prediction Results\n\n")

prediction\_data = prediction(X\_test, cls)

svm\_acc = cal\_accuracy(y\_test, prediction\_data,'SVM Accuracy, Classification Report & Confusion Matrix')

def runANN():

global ann\_acc

model = Sequential()

model.add(Dense(12, input\_dim=9, activation='relu'))

model.add(Dense(9, activation='relu'))

model.add(Dense(1, activation='softmax'))

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(X\_train, y\_train, epochs=100, batch\_size=64)

\_, ann\_acc = model.evaluate(X\_train, y\_train)

ann\_acc = ann\_acc\*100

text.insert(END,"ANN Accuracy : "+str(ann\_acc)+"\n\n")

def predictDisease():

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="dataset")

test = pd.read\_csv(filename)

test = test.values[:, 0:9]

text.insert(END,filename+" test file loaded\n");

y\_pred = cls.predict(test)

for i in range(len(test)):

if str(y\_pred[i]) == '1':

text.insert(END,"X=%s, Predicted=%s" % (X\_test[i], 'Infected')+"\n")

else:

text.insert(END,"X=%s, Predicted=%s" % (X\_test[i], 'Not Infected')+"\n")

def graph():

height = [svm\_acc,ann\_acc]

bars = ('SVM Accuracy', 'ANN Accuracy')

y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.show()

font = ('times', 16, 'bold')

title = Label(main, text='Comparison of Machine Learning Methods for Breast Cancer Diagnosis')

title.config(bg='greenyellow', fg='dodger blue')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 14, 'bold')

uploadButton = Button(main, text="Upload Wisconsin Dataset", command=upload)

uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

modelButton = Button(main, text="Generate Train & Test Model", command=generateModel)

modelButton.place(x=350,y=550)

modelButton.config(font=font1)

runsvm = Button(main, text="Run SVM Algorithm", command=runSVM)

runsvm.place(x=750,y=550)

runsvm.config(font=font1)

runann = Button(main, text="Run ANN Algorithm", command=runANN)

runann.place(x=50,y=600)

runann.config(font=font1)

predict = Button(main, text="Upload Test Data & Predict Disease", command=predictDisease)

predict.place(x=350,y=600)

predict.config(font=font1)

graph = Button(main, text="Accuracy Graph", command=graph)

graph.place(x=750,y=600)

graph.config(font=font1)

main.config(bg='LightSkyBlue')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## Implementation

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

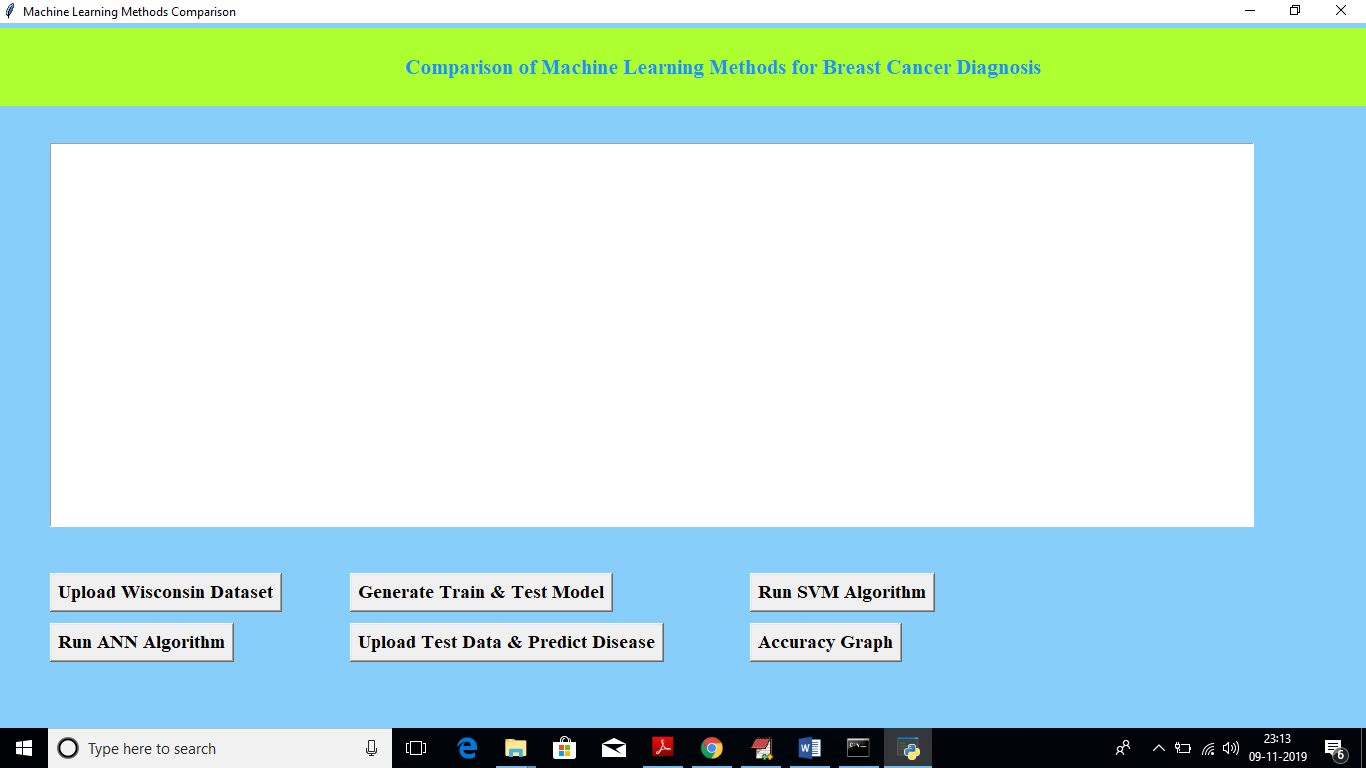
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

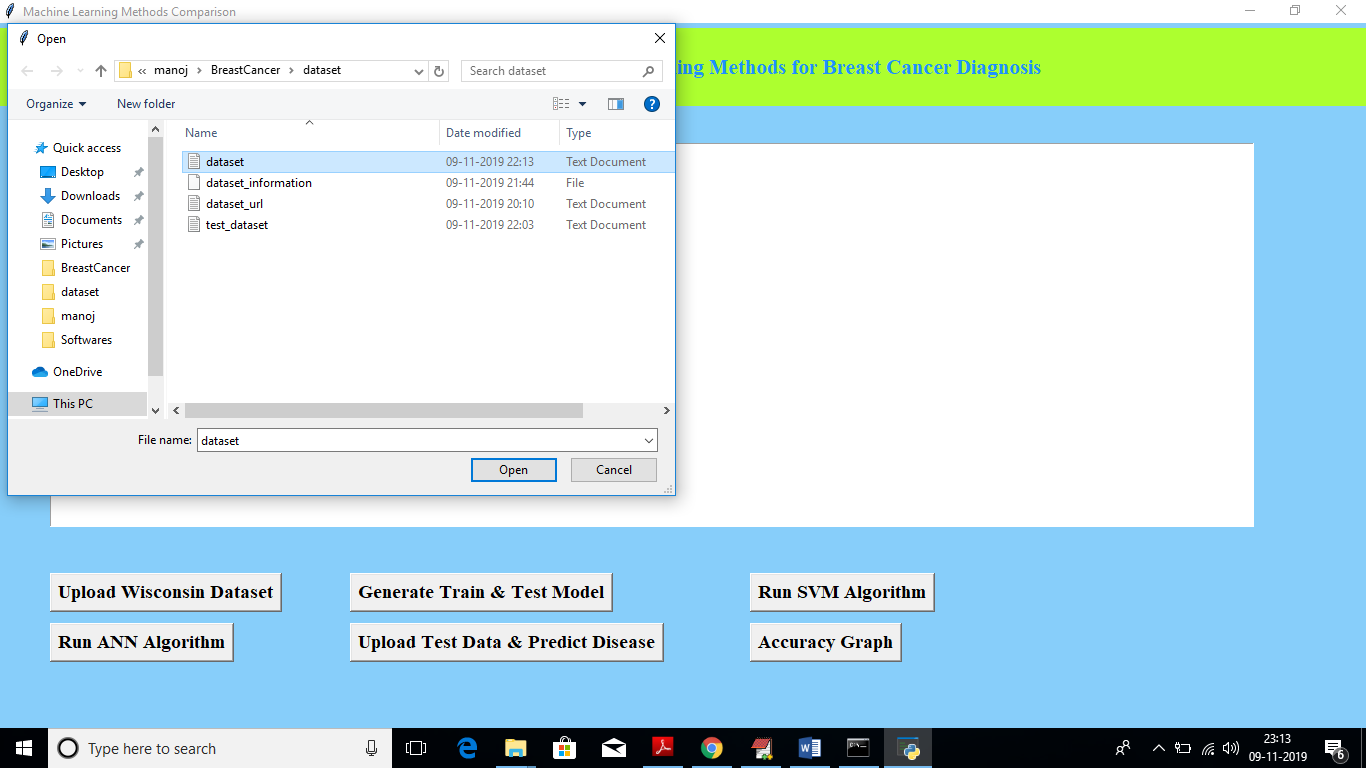
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | upload Wisconsin dataset | Test whether the dataset is uploaded or not | if the dataset may not loaded | we cannot do further operations | we can do further operations | High | High |
| 02 | generate train & test model | Verify the train and test model will generated or not | Without loading the dataset | we cannot split data into train & test | we can split data into train & test | High | High |
| 03 | run svm algorithm | Verify the SVM algorithm will run or not | Without spitting the data | we cannot generate training model | we can generate training model | High | High |
| 04 | run ann algorithm | Verify either ANN algorithm will run or not | Without spitting the data | we cannot run ANN algorithm to get accuracy | we can run ANN algorithm to get accuracy | High | High |
| 05 | upload test data & predict disease | Verify either the predicted results are displayed or not | without uploading of test data | we cannot get the predicted results | we can get the predicted results | High | High |
| 06 | accuracy graph | verify the accuracy graph is displayed or not | without saving the accuracy values | accuracy graph is not displayed | the accuracy graph is displayed successfully | High | High |

**7. SCREENSHOTS**

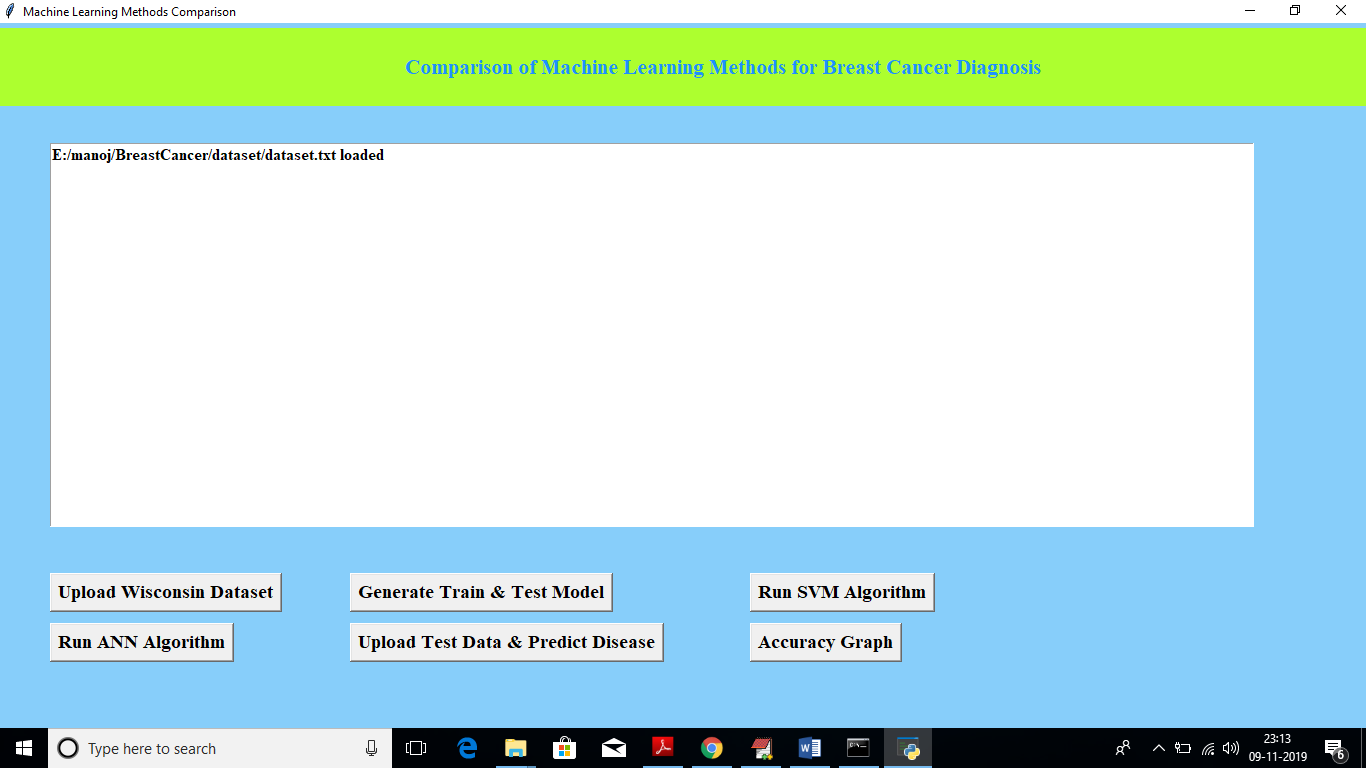
Double click on ‘run.bat’ file to get below screen.



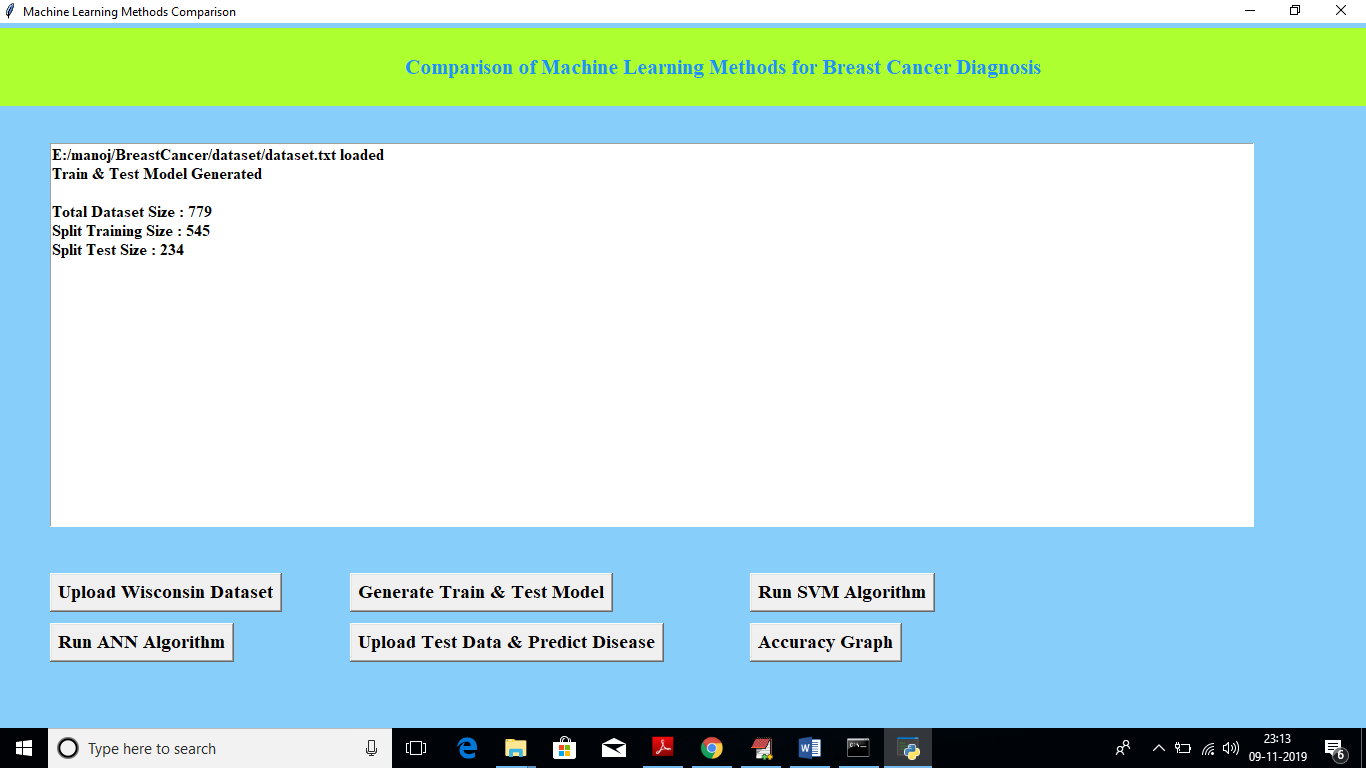
Now click on ‘Upload Wisconsin Dataset’ button to upload dataset



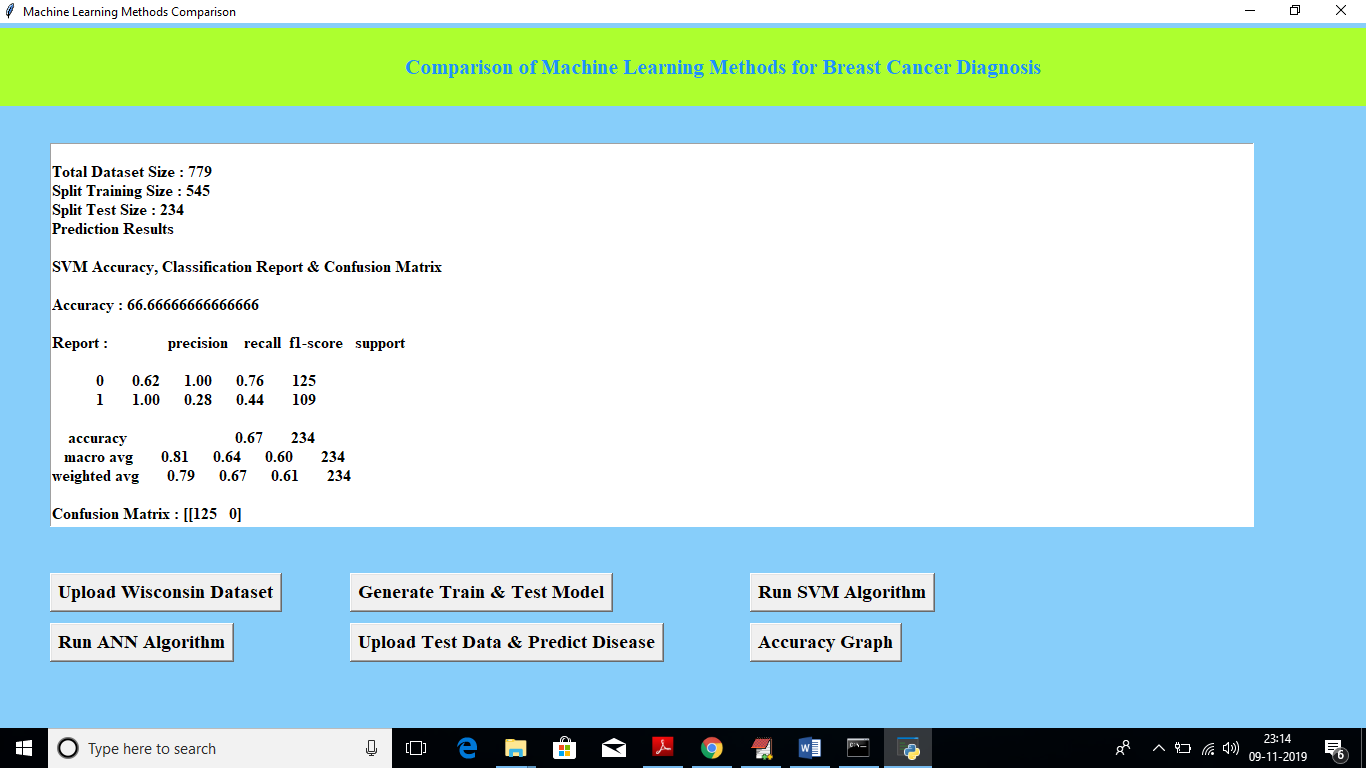
In above screen I am uploading ‘dataset.txt’ file and after upload will get below screen



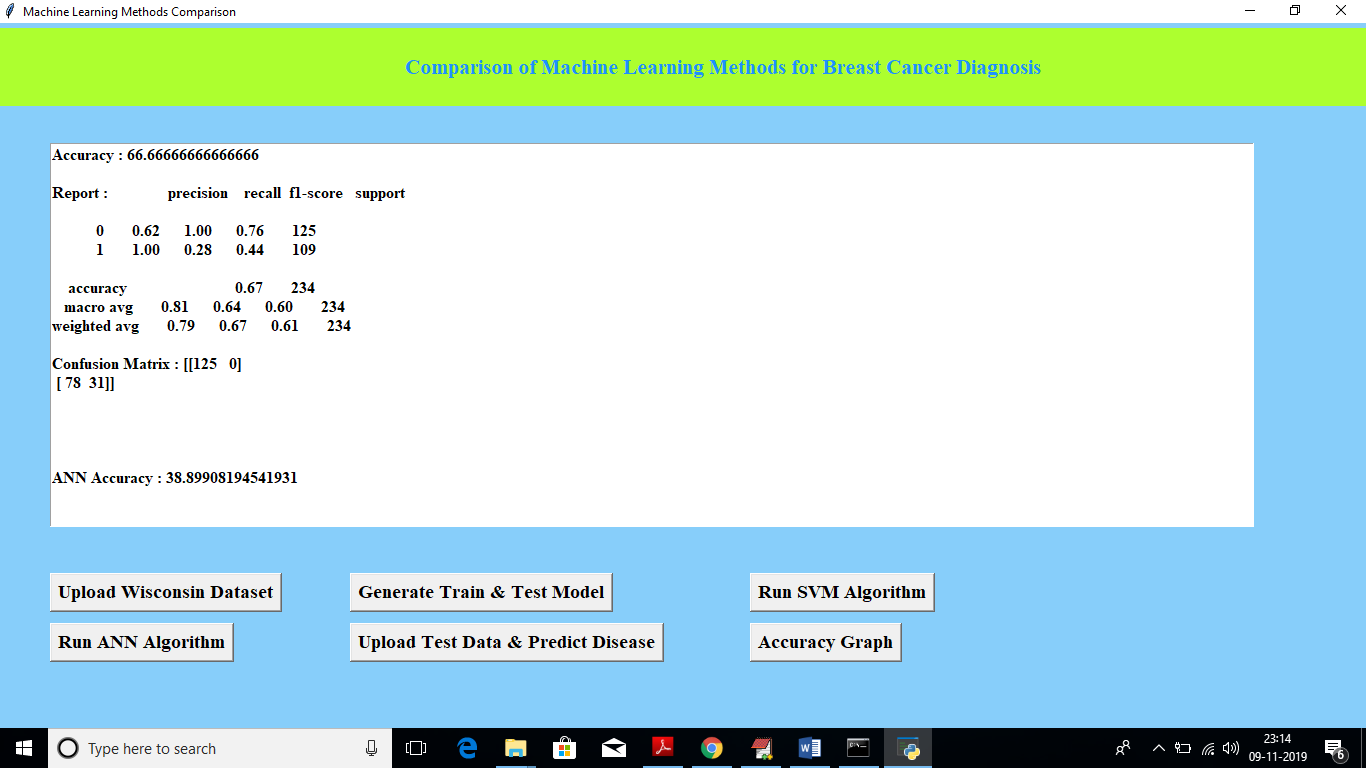
Now click on ‘Generate Train & Test Model’ button to split dataset into train and test



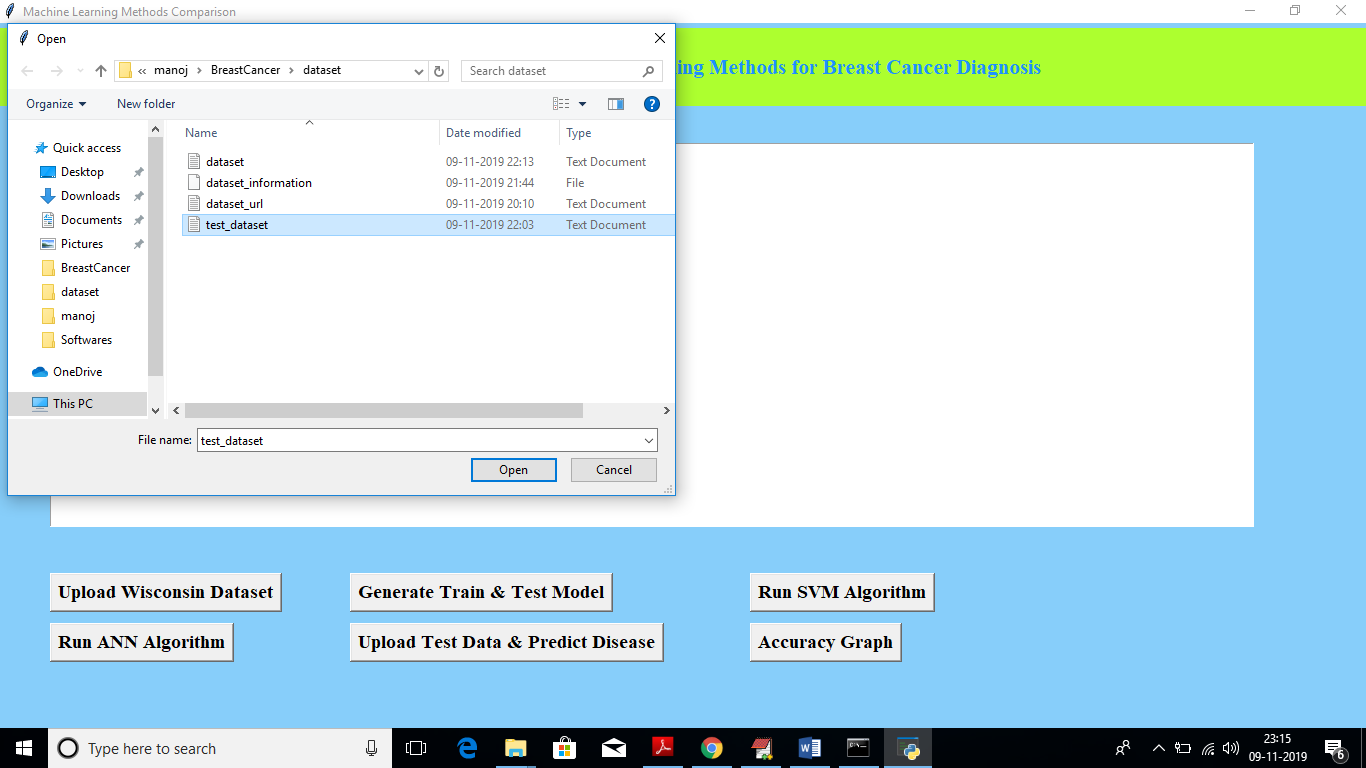
In above screen we can see dataset contains total 779 records and training size splits to 545 and test size splits to 234. Now click on ‘Run SVM Algorithm’ button to generate training model with SVM



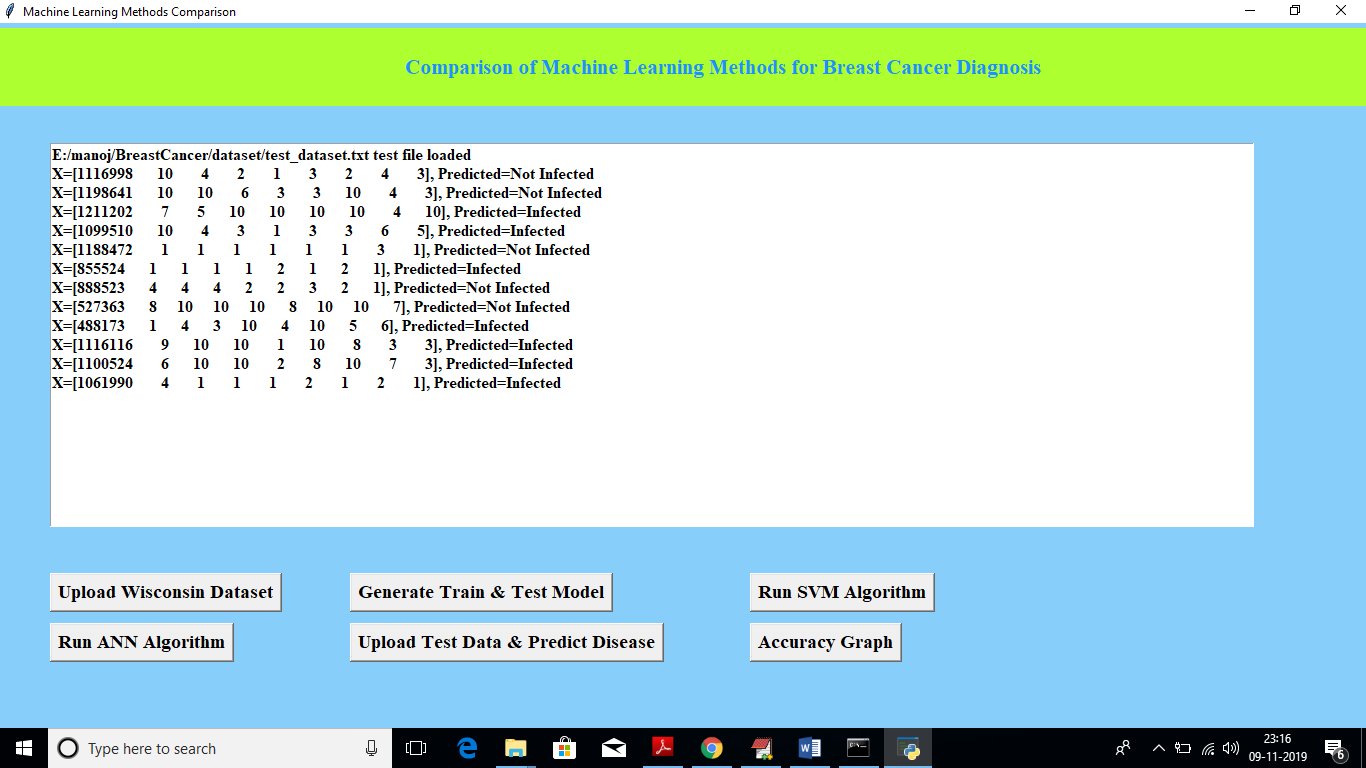
In above screen we can see Accuracy value, precision and recall obtained from SVM. SVM got 66.66% accuracy. Now click on ‘Run ANN Algorithm’ to get ANN Accuracy



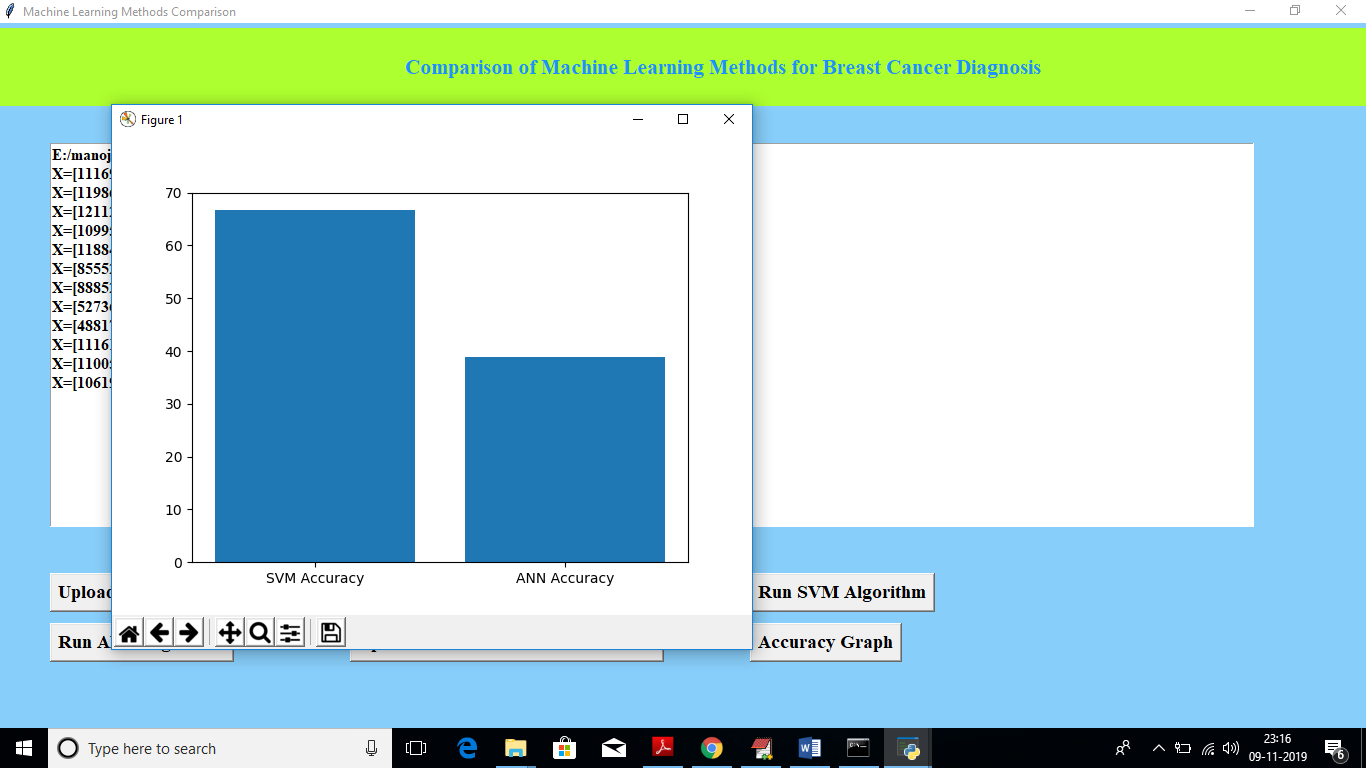
In above screen ANN got 38% accuracy and SVM got better accuracy than ANN. Now click on ‘Upload Test Data & Predict Disease’ button to upload test data and to predict disease



In above screen I am uploading test\_dataset file and below is predicted result



Now click on ‘Accuracy Graph’ button to get accuracy graph of both algorithms



In above graph x-axis represents algorithm name and y-axis represents accuracy of that algorithm. From above graph we can conclude that SVM is better than ANN

**8. CONCLUSION**

Breast Cancer is the most frequent disease as a cancer type for women. Therefore, any development for diagnosis and prediction of cancer disease is capital important for a healthy life. In this paper, we have discussed two popular machine learning techniques for Wisconsin Breast Cancer classification. Artificial Neural Network and Support Vector Machine are used as ML techniques for the classification of WBC (Original) dataset in WEKA tool. The effectiveness of applied ML techniques is compared in term of key performance metrics such as accuracy, precision, recall and ROC area. Based on the performance metrics of the applied ML techniques, SVM (Sequential Minimal Optimization Algorithm) has showed the best performance in the accuracy of 96,9957 % for the diagnosis and prediction from WBC dataset.

**9. REFERENCES**

[1] Cancer, https://www.who.int/en/news-room/fact-sheets/detail/cancer. Last Access: 25.01.2019.

[2] Siegel, R. L., Miller, K. D., & Jemal, A. (2018). Cancer statistics, Ca-a Cancer Journal for Clinicians, 68 (1), pp. 7-30.

[3] Maity, N. G., & Das, S. (2017). Machine learning for improved diagnosis and prognosis in healthcare. In 2017 IEEE Aerospace Conference, pp. 1-9.

[4] Asri, H., Mousannif, H., Al Moatassime, H., & Noel, T. (2016). Using machine learning algorithms for breast cancer risk prediction and diagnosis. Procedia Computer Science, 83, pp. 1064-1069.

[5] Bazazeh, D., & Shubair, R. (2016). Comparative study of machine learning algorithms for breast cancer detection and diagnosis. In 2016 5th International Conference on Electronic Devices, Systems and Applications, pp. 1-4.

[6] Ahmad, L. G., Eshlaghy, A. T., Poorebrahimi, A., Ebrahimi, M., & Razavi, A. R. (2013). Using three machine learning techniques for predicting breast cancer recurrence. J Health Med Inform, 4 (124).

[7] Bektas, B., & Babur, S. (2016). Machine learning based performance development for diagnosis of breast cancer, Medical Technologies National Congress, pp. 1-4.

[8] Chen, H. L., Yang, B., Liu, J., & Liu, D. Y. (2011). A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. Expert Systems with Applications, 38 (7), pp. 9014-9022.

[9] UCI Breast Cancer Wisconsin (Original) Dataset, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Origina l%29. Last Access: 30.01.2019.

[10] Huang, M. W., Chen, C. W., Lin, W. C., Ke, S. W., & Tsai, C. F. (2017). SVM and SVM ensembles in breast cancer prediction. PloS one, 12 (1).

[11] Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction, Computational and structural biotechnology Journal, 13, pp. 8-17.

[12] Umadevi, S., & Marseline, K. J. (2017). A survey on data mining classification algorithms. In 2017 International Conference on Signal Processing and Communication, pp. 264-268.

[13] Padmapriya S., Devika M., Meena V., Dheebikaa S. B., & Vinodhini R. (2016). Survey on Breast Cancer Detection Using Weka Tool. Imperial Journal of Interdisciplinary Research (IJIR), Vol 2, no. 4.

[14] S. Turgut, M. Dagtekin, T. Ensari, Microarray Breast Cancer Data Classification Using Machine Learning Methods, Int. Conf. on Electric Electronics, Computer Science, Biomedical Engineerings' Meeting, DOI: 10.1109/EBBT.2018.8391468, April 18-19, 2018.

[15] M. Amrane, S. Oukid, I. Gagaoua, T. Ensari, Breast Cancer Classification Using Machine Learning, Int. Conf. on Electric Electronics, Computer Science, Biomedical Engineerings' Meeting, DOI: 10.1109/EBBT.2018.8391453, April 18-19, 2018.