UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

Classification of Breast Cancer Symptom and Treatment from Biomedical Document

Nursyamimi Binti Subni
Faculty of Computing
Universiti Teknologi Malaysia
Skudai, Johor, Malaysia
nursyamimi.s@graduate.utm.my

***Abstract*—This study aims to develop a classification model for the symptoms and treatments of Breast Cancer (BC) using machine learning approaches based on the abstracts of BC-related journal articles. A systematic research methodology is designed to achieve the aim of research. The dataset is collected from PubMed website using Entrez in Biopython and pre-processed by splitting abstracts into sentences and performing text-cleaning techniques, and labelling data. TF-IDF method is used for feature extraction and vectorizes the pre-processed data into numerical data for fitting in the machine learning models such as SVM, Decision Tree, and Naïve Bayes. The performance of these models is evaluated with several performance metrics based on confusion matrix, such as accuracy, precision, and recall. The performance of applying different hyperparameters and different sets of training and testing data sets will be compared. Most of the classifiers obtained the best performance when using 90:10 splits. The SVM model outperforms the other classifiers in terms of performance metrics and confusion matrix. The research findings can provide a solution for identifying and classifying the symptoms and treatments of BC from multiple biomedical journal articles more efficiently and facilitate the development of more effective interventions for BC management.**

Keywords — Breast Cancer, Document, Text Classification, Text Processing, Symptom and Treatment, Machine Learning.

I. Introduction

Breast cancer remains a significant health issue worldwide, affecting millions of women each year. Early detection and effective treatment are critical for improving survival rates. In recent years, the volume of biomedical literature on breast cancer has grown exponentially, making it challenging for healthcare professionals to stay updated with the latest research findings. This study aims to address this challenge by developing automated methods for classifying biomedical documents related to breast cancer.

Natural language processing (NLP) and machine learning techniques offer promising solutions for the automatic classification of text documents. These methods can process large volumes of text data, extract relevant information, and categorize documents based on their content. By implementing these techniques, we can enhance the accessibility and organization of biomedical literature, ultimately supporting better decision-making in clinical practice and research.

II. Literature Review

A. Text Classification

Text classification is the process of automatically categorizing a collection of documents into one or more labeled or predefined groups based on their content [5]. This process is fundamental in natural language processing and has applications in various domains such as information retrieval, sentiment analysis, spam detection, and more [3]. The key steps of the text classification process include data collection, text preprocessing, feature extraction, dimensionality reduction, classification approaches, and performance evaluation [5].

B. Feature Extraction

Feature extraction is a critical step in text classification, aimed at reducing the dimensionality of the text data while preserving its relevant information [1]. Techniques such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Linear Discriminant Analysis (LDA) are commonly used [9]. Additionally, weighted word

strategies like Bag-of-Words (BoW) and Term Frequency-Inverse Document Frequency (TF-IDF), as well as word embedding methods like Word2Vec and GloVe, are prevalent in natural language processing [10]. Feature extraction is vital as it simplifies the computational complexity and enhances the performance of machine learning models [6].

C. Multilabel Classification

Multilabel Classification (MLC) is essential for scenarios where each instance can belong to multiple labels simultaneously, which is common in text classification, tag recommendation, and image labeling [11]. Unlike single-label classification, MLC handles multiple labels per instance, providing a more flexible and comprehensive categorization system [7]. The two primary approaches in MLC are problem transformation and algorithm adaptation. Problem transformation methods, such as Binary Relevance, Classifier Chain, and Label Powerset, convert the multi-label problem into several single-label problems [2]. Algorithm adaptation methods modify existing single-label algorithms to directly handle multi-label data, enhancing their ability to manage label dependencies effectively [2].

D. Machine Learning Approaches

Supervised learning approaches are widely used in text classification due to their ability to learn from labeled data and make accurate predictions on new, unseen data. Techniques like Decision Trees, Support Vector Machines (SVM), and Neural Networks are commonly employed. Decision Trees are interpretable and can capture non-linear relationships, making them useful for complex classification tasks [4]. SVMs, based on the Structural Risk Minimization principle, aim to find decision boundaries with maximum margins, reducing overfitting and improving generalization [8]. These methods are powerful and effective, particularly when combined with robust feature extraction techniques.

III. METHODOLOGY

The design and implementation of the research were carried out in a structured manner to ensure the accuracy and reliability of the results. This section details the steps involved in preparing the dataset, developing the classification models, and evaluating their performance. The flowchart of research methodology is shown in Fig. 1.

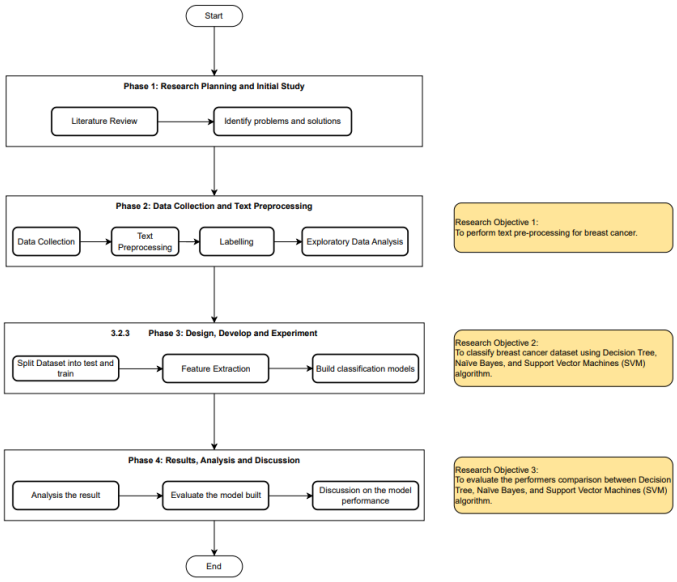

Fig. 1. Workflow of Research Methodology

A. Data Preparation

Effective data preparation is crucial for successful text classification. This phase involves collecting and preprocessing textual data to make it suitable for analysis. Text preprocessing includes steps such as tokenization, stop-word removal, stemming, and lemmatization to clean and normalize the text [5]. Ensuring the dataset is well-labeled is also essential, as the quality of the labels directly impacts the performance of the classification models [3]. Data preparation can be time-consuming but is necessary to ensure the accuracy and reliability of the text classification system.

B. Model Building

The balanced dataset containing biomedical documents related to breast cancer from NCBI was loaded as a CSV file. To prepare the training and testing data, the dataset was divided into ‘X’ and ‘y’ variables, where ‘X’ represents the text data and ‘y’ represents the labels. The dataset was split into three sets of ratios: 70% for training data and 30% for testing data, 80% for training data and 20% for testing data, and 90% for training data and 10% for testing data.

Before fitting the text data into the model, it needs to be transformed into numerical values that can be understood by the classifiers. The TfidfVectorizer class was imported to convert the text data to a matrix of TF-IDF features. The ‘ngram_range’ parameter was used to specify a range of n-values for different n-grams to be extracted. In this research, a ‘ngram_range’ of (1, 5) means the creation of the TF-IDF matrix will include unigrams (single words), bigrams (two-word combinations), and up to 5-grams (five-word combinations). The training data was fitted into the vectorizer to learn the vocabulary and the inverse document frequency (IDF), followed by the transformation of training and testing data into a sparse matrix of TF-IDF features.

Three algorithms were utilized for classification: Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. The dataset was trained by tuning the hyperparameters according to each classifier to identify the best parameter for classification.

All results obtained and their performance measurements were further analyzed, compared, and discussed. Fig. 2. show the experimental design of the classification for three models.

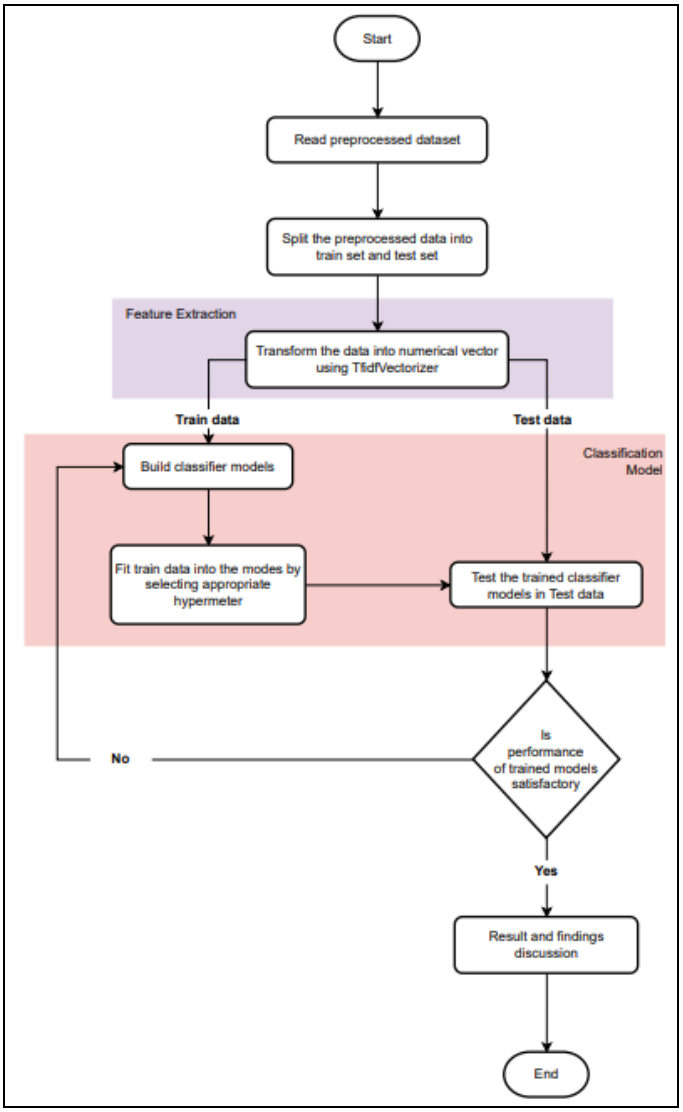

Fig. 2. Experimental Design of Classification Model

C. Performance Evaluation

The measurements such as accuracy, precision, recall and F1-score are based on the confusion matrix. The confusion matrix comprises of four evaluation outputs which are true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). TP denotes a positive class that is correctly predicted by the model, TN denotes a negative class that is correctly predicted by the model, FP denotes a positive class that is incorrectly predicted by the model and FN denotes a negative class that is incorrectly predicted by the model.

Accuracy is defined as the ratio of the correctly classified results to the total number of predictions. It can be calculated by using the following formula:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision is the proportion of correctly identified outcomes that were positively identified. It can be calculated by using the following formula:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall indicates the percentage of data that was projected to be positive but turned out to be negative. It is the percentage of TP in the collection of all genuinely positive data.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

F1-score is calculated from the combination of Precision and Recall by getting their harmonic mean.

$$\text{F1-score} = 2 \times \frac{Precision \times Recall}{(Precision+Recall)} \quad (4)$$

IV. Results

In this section, the results of the three machine learning methods using performance metrics based on confusion matrix will be displayed and analyzed.

A. Performance Measure of Support Vector Machine

Fig. 3. displays the performance of the SVM model trained with different values of the regularization parameter C, evaluated on three different train-test splits (70:30, 80:20, 90:10). The results cover accuracy, precision, recall, and F1-score for three classes: "Both Not Exists (0)", "Symptom(s) Exist (1)", and "Treatment(s) Exist (2)". The first table in Figure 5.1 shows the results of the 70:30 train-test ratio, the second table shows the results of the 80:20 train-test ratio, and the third table shows the results of the 90:10 train-test ratio.

Train test ratio: 70:30

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
C=0.8	97.95	0.98	1.00	0.99	1.00	0.23	0.38	0.99	0.86	0.92
C=0.9	97.99	0.98	1.00	0.99	1.00	0.23	0.38	0.99	0.87	0.92
C=1.0	97.99	0.98	1.00	0.99	1.00	0.23	0.38	0.99	0.87	0.92
C=1.1	98.23	0.98	1.00	0.99	1.00	0.37	0.54	0.98	0.89	0.93
C=1.2	98.47	0.99	1.00	0.99	1.00	0.40	0.57	0.98	0.92	0.95
C=1.3	98.57	0.99	1.00	0.99	1.00	0.40	0.57	0.97	0.94	0.95

Train test ratio: 80:20

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
C=0.8	98.28	0.98	1.00	0.99	1.00	0.25	0.40	1.00	0.89	0.94
C=0.9	98.35	0.98	1.00	0.99	1.00	0.25	0.40	1.00	0.90	0.95
C=1.0	98.64	0.99	1.00	0.99	1.00	0.40	0.57	1.00	0.91	0.95
C=1.1	98.78	0.99	1.00	0.99	1.00	0.40	0.57	1.00	0.94	0.97
C=1.2	98.93	0.99	1.00	0.99	1.00	0.45	0.62	1.00	0.95	0.67
C=1.3	99.00	0.99	1.00	0.99	1.00	0.50	0.67	1.00	0.95	0.97

Train test ratio: 90:10

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
C=0.8	98.28	0.98	1.00	0.99	1.00	0.31	0.47	1.00	0.93	0.96
C=0.9	98.42	0.98	1.00	0.99	1.00	0.38	0.56	1.00	0.93	0.96
C=1.0	98.57	0.98	1.00	0.99	1.00	0.46	0.63	1.00	0.93	0.96
C=1.1	98.85	0.99	1.00	0.99	1.00	0.54	0.70	1.00	0.95	0.98
C=1.2	99.28	0.99	1.00	1.00	1.00	0.62	0.76	1.00	1.00	1.00
C=1.3	99.43	0.99	1.00	1.00	1.00	0.69	0.82	1.00	1.00	1.00

Fig. 3. Results of Support Vector Machine

Based on Figure 3, the parameter value that yields the best results across all train-test splits is C=1.3. For the 70:30 split, the accuracy peaks at 98.57%. For class 0, precision and F1-score are both high at 0.99, with recall consistently at 1.00. For class 1, the best F1-score is 0.57 with a recall of 0.40. Class 2 achieves the highest precision of 0.97 and an F1-score of 0.95.

For the 80:20 train-test ratio, the highest accuracy is 99.00% with C=1.3. Class 0 shows perfect precision and F1-scores of 0.99 and 1.00 respectively. For class 1, the F1-score is 0.67 with a recall of 0.50. Class 2 maintains high performance with a precision of 1.00 and an F1-score of 0.97.

In the 90:10 train-test ratio, the accuracy reaches 99.43% with C=1.3. Class 0 continues to exhibit excellent precision and F1-scores of 1.00. For class 1, the F1-score is 0.67, matching the recall of 0.50. Class 2's precision and F1-scores are 0.99 and 0.97 respectively.

In conclusion, C=1.3 demonstrates the highest accuracy across all train-test splits, providing a balanced performance for all metrics, particularly for the "Symptom(s) Exist" and "Treatment(s) Exist" classes. The SVM model with this parameter shows its best results with the 90:10 train-test ratio, achieving the highest accuracy of 99.43%.

B. Performance Measure of Decision Tree

Based on the performance of the Decision Tree model evaluated with 10,000 features and various maximum depths, the results were analyzed across three different train-test splits (70:30, 80:20, 90:10). The evaluation included metrics such as accuracy, precision, recall, and F1-score for three classes: "Both Not Exists (0)", "Symptom(s) Exist (1)", and "Treatment(s) Exist (2)".

Train test ratio: 70:30

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
max_depth=21 max_features=1000	97.71	0.98	1.00	0.99	1.00	0.73	0.85	0.97	0.72	0.83
max_depth=22 max_features=1000	97.28	0.98	1.00	0.99	0.88	0.73	0.80	0.94	0.70	0.80
max_depth=22 max_features=1000	96.99	0.97	1.00	0.98	1.00	0.77	0.87	0.93	0.63	0.75
max_depth=22 max_features=1000	95.56	0.96	0.99	0.98	1.00	0.27	0.42	0.87	0.55	0.68
max_depth=22 max_features=1000	94.56	0.95	1.00	0.97	1.00	0.27	0.42	0.85	0.38	0.53

Train test ratio: 80:20

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
max_depth=21 max_features=1000	97.99	0.98	1.00	0.99	1.00	0.75	0.86	0.97	0.74	0.84
max_depth=22 max_features=1000	97.35	0.97	1.00	0.99	1.00	0.15	0.26	0.98	0.77	0.86
max_depth=22 max_features=1000	98.35	0.98	1.00	0.99	0.95	0.90	0.92	0.98	0.77	0.86
max_depth=22 max_features=1000	94.70	0.95	1.00	0.97	0.33	0.05	0.09	0.97	0.36	0.52
max_depth=22 max_features=1000	98.57	0.99	1.00	0.99	1.00	0.80	0.89	0.96	0.84	0.89

Fig. 4. Results of Decision Tree (Part 1)

Train test ratio: 90:10

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
max_depth=21 max_features=1000	97.71	0.98	1.00	0.99	1.00	0.31	0.47	0.95	0.86	0.90
max_depth=22 max_features=1000	99.43	0.99	1.00	1.00	1.00	0.77	0.87	1.00	0.98	0.99
max_depth=23 max_features=1000	99.00	0.99	1.00	0.99	1.00	0.92	0.96	0.95	0.91	0.93
max_depth=24 max_features=1000	98.42	0.98	1.00	0.99	1.00	0.85	0.92	0.97	0.81	0.89
max_depth=25 max_features=1000	97.56	0.98	1.00	0.99	0.92	0.85	0.88	0.97	0.70	0.81

Fig. 5. Results of Decision Tree (Part 2)

Fig. 4. and Fig. 5. show the table for three different train-test ratios. For the 70:30 train-test ratio, the highest accuracy of 97.71% was achieved with max_depth=21 and max_features=1000. For class 0, the precision and F1-score peaked at 0.98 and 0.99 respectively. The recall for class 0 was consistently high across different depths, ranging from 0.97 to 1.00. For class 1, the best F1-score was 0.87 with a recall of 0.73 at max_depth=22. Class 2 showed the highest precision of 0.97 at max_depth=21, with an F1-score of 0.83.

With an 80:20 train-test ratio, the highest accuracy of 98.57% was recorded at max_depth=25 and max_features=1000. Class 0 exhibit excellent high precision and F1-scores of 0.99 respectively. Class 1 achieved its best F1-score of 0.92 at max_depth=23 with a recall of 0.90 For class 2, the best precision of 0.98 and F1-score of 0.89 were observed at max_depth=25.

For the 90:10 train-test ratio, the model reached an accuracy peak of 99.43% at max_depth=22 with max_features=1000. Class 0 achieved a perfect precision and F1-score of 0.99 and 1.00. Class 1's best F1-score was 0.96 at max_depth=23 with a recall of 0.92. Class 2 had the highest precision of 1.00 at max_depth=22, with an F1-score of 0.99.

In conclusion, max_depth=22 with 10,000 features consistently provided the best results across all train-test splits, with the highest accuracy of 99.43% observed in the 90:10 split. This configuration showed a strong balance across all metrics, especially for the "Symptom(s) Exist" and "Treatment(s) Exist" classes.

C. Performance Measure of K-Nearest Neighbor

Fig. 6. shows the performance of the Naive Bayes model trained with different alpha values, evaluated on three different train-test splits (70:30, 80:20, 90:10). The results include accuracy, precision, recall, and F1-score for three classes: "Both Not Exist (0)", "Symptom(s) Exist (1)", and "Treatment(s) Exist (2)".

Train test ratio: 70:30

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
alpha=0.1	92.88	0.93	1.00	0.96	0.00	0.00	0.00	0.94	0.12	0.21
alpha=0.2	92.45	0.92	1.00	0.96	0.00	0.00	0.00	0.88	0.05	0.10
alpha=0.3	92.41	0.92	1.00	0.96	0.00	0.00	0.00	0.86	0.04	0.09

Train test ratio: 80:20

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
alpha=0.1	93.70	0.94	1.00	0.97	0.00	0.00	0.00	1.00	0.16	0.28
alpha=0.2	93.27	0.90	1.00	0.96	0.00	0.00	0.00	1.00	0.09	0.16
alpha=0.3	93.27	0.93	1.00	0.96	0.00	0.00	0.00	1.00	0.09	0.16

Train test ratio: 90:10

Parameter	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
alpha=0.1	92.98	0.93	1.00	0.96	0.00	0.00	0.00	1.00	0.16	0.28
alpha=0.2	92.41	0.92	1.00	0.96	0.00	0.00	0.00	1.00	0.07	0.13
alpha=0.3	92.41	0.92	1.00	0.96	0.00	0.00	0.00	1.00	0.07	0.13

Fig. 6. Results of Naïve Bayes

The tables in Fig. 6. present the results for each train-test ratio, where the first table shows the results of a 70:30 train-test split, the second table shows the results of an 80:20 train-test split and the third table shows the results of a 90:10 train-test split. For the 70:30 split, the highest accuracy achieved is 92.88% with alpha=0.1. For class 0, the precision is 0.93, recall is 1.00, and F1-score is 0.96. However, for class 1, the precision, recall, and F1-score are all 0.00, indicating that the model is not able to effectively predict this class. For class 2, the precision is 0.94, recall is 0.12, and F1-score is 0.21.

For the 80:20 split, the highest accuracy is 93.70%, also with alpha=0.1. For class 0, the precision is 0.94, recall is 1.00, and F1-score is 0.97. Similar to the 70:30 split, class 1 has a precision, recall, and F1-score of 0.00. For class 2, the precision is 1.00, recall is 0.16, and F1-score is 0.28.

For the 90:10 split, the highest accuracy is 92.98% with alpha=0.1. For class 0, the precision is 0.93, recall is 1.00, and F1-score is 0.96. For class 1, the precision, recall, and F1-score

remain at 0.00. For class 2, the precision is 1.00, recall is 0.16, and F1-score is 0.28.

Based on the results presented in Figure 5.4, the parameter that gives the best results across all train-test splits is alpha=0.1. The highest accuracy achieved is 93.70% for the 80:20 train-test split. For class 0, the model consistently shows high precision and recall, resulting in high F1-scores. However, the model struggles to predict class 1, with precision, recall, and F1-scores all at 0.00. For class 2, the model shows relatively better performance with a precision of 1.00 but has lower recall and F1-scores.

In conclusion, alpha=0.1 shows the highest accuracy across all train-test splits. This alpha value provides a good balance between precision and recall for class 0, but the model needs improvement in predicting classes 1 and 2. The best performance is observed with the 80:20 train-test ratio, achieving the highest accuracy of 93.70%.

V. Discussion

According to the performance of the three different machine learning models using different train-test splits described in IV, two models such as Support Vector Machine and Decision Tree performed best with train test ratio of 90:10 which is 90% of training data and 10% of testing data. Therefore, the comparison and the justification between all models will be made based on 90:10 split's results

A. Comparison on Overall Performance

Fig. 7. illustrates the tabulated results for the three machine learning classifiers.

Model	Accuracy (%)	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Support Vector Machine	99.43	0.99	1.00	1.00	1.00	0.69	0.82	1.00	1.00	1.00
Decision Tree	99.43	0.99	1.00	1.00	1.00	0.77	0.87	1.00	0.98	0.99
Naïve Bayes	92.98	0.93	1.00	0.96	0.00	0.00	0.00	1.00	0.16	0.28

Fig. 7. Results of all models with 9:10 split

Based on the results illustrated in Fig. 7. , we can evaluate the performance of three machine learning classifiers: Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. These models have been assessed based on their accuracy and other performance metrics for three different classes (labels): "Both Not Exist (0)", "Symptom(s) Exist (1)", and "Treatment(s) Exist (2)".

The Decision Tree and SVM both have the highest accuracy, each at 99.43%, demonstrating their effectiveness in handling the dataset. On the other hand, Naïve Bayes shows a lower accuracy of 92.98%, indicating it struggles more with the given data.

For "Both Not Exist (0)". Both SVM and Decision Tree exhibit perfect performance for this class, each achieving an F1-score of 1.00. This indicates that they can accurately identify instances where neither symptoms nor treatments exist, reflecting their ability to handle well-defined boundaries in the data. Naïve Bayes also performs well in this category, achieving a slightly lower F1-score of 0.96, which suggests it is capable but not as precise as the other two models.

For "Symptom(s) Exist (1)". In this category, the SVM shows moderate performance with an F1-score of 0.82. Despite its high precision of 1.00, its recall is lower at 0.69, indicating that while it correctly identifies most of the true positive cases, it misses some instances. The Decision Tree performs slightly better in this class with an F1-score of 0.87, showcasing balanced precision (1.00) and recall (0.77). This indicates it handles this class more effectively than the SVM. However, Naïve Bayes performs poorly, with an F1-score of 0.00. This highlights its significant struggle to classify instances where symptoms exist, possibly due to the complexity and variability in this category.

For "Treatment(s) Exist (2)". For this class, SVM again demonstrates perfect performance with an F1-score of 1.00, showing it can accurately classify instances where treatments exist. The Decision Tree is almost as effective, with an F1-score of 0.99, slightly falling short in recall (0.98) compared to its perfect precision (1.00). Naïve Bayes, however, shows significantly lower performance with an F1-score of 0.28. Despite having perfect precision (1.00), its recall is very low (0.16), indicating it fails to identify many true positive cases in this category.

SVM, with its high accuracy of 99.43% and balanced performance across most classes, shows strong capability in handling the dataset. Its perfect F1-scores for the "Both Not Exist (0)" and "Treatment(s) Exist (2)" classes suggest it effectively separates these data points, though it faces slight challenges with the "Symptom(s) Exist (1)" class. The Decision Tree, also with an accuracy of 99.43%, demonstrates consistent performance across all classes, particularly excelling in the "Symptom(s) Exist (1)" class compared to SVM. Naïve Bayes, with the lowest accuracy of 92.98%, struggles significantly with the "Symptom(s) Exist (1)" and "Treatment(s) Exist (2)" classes, indicating difficulties in handling the complexities and possible noise in the dataset.

SVM and Decision Tree are the top performers, demonstrating high accuracy and balanced performance across all classes. Naïve Bayes, on the other hand, underperforms relative to the other models, indicating it may not be well-suited for this dataset due to its lower accuracy and poor handling of certain classes. The dataset itself likely has well-defined boundaries for the "Both Not Exist (0)" and "Treatment(s) Exist (2)" classes, making them easier to classify with high precision and recall. However, the "Symptom(s) Exist (1)" class might be more complex or less frequent, posing challenges for accurate classification.

In terms of model strengths, SVM and Decision Tree excel due to their ability to capture complex patterns. SVM's strong precision and recall, especially in the "Both Not Exist (0)" and "Treatment(s) Exist (2)" classes, indicate its robustness in finding optimal margins. The Decision Tree's consistent high performance across all classes reflects its effectiveness in capturing the intricacies of the dataset. Naïve Bayes, while generally effective for simpler tasks, struggles with this dataset, likely due to high dimensionality and noise affecting its distance calculations and classification performance.

Ultimately, the choice of model depends on the specific requirements and constraints of the application. However, based on these results, SVM and Decision Tree show the best

adaptability and performance for this dataset, making them the preferable choices for accurately classifying the given data.

B. Comparison on Confusion Matrix

Fig. 8. illustrates the tabulated confusion matrix results for all three machine learning classifiers: Support Vector Machine (SVM), Decision Tree, and Naïve Bayes. These models are assessed based on their predictions for three different classes (labels): "Both Not Exist (0)", "Symptom(s) Exist (1)", and "Treatment(s) Exist (2)".

Model	Both Not Exist (0)			Symptom(s) Exist (1)			Treatment(s) Exist (2)		
	Correctly Predicted	Incorrectly Predicted		Correctly Predicted	Incorrectly Predicted		Correctly Predicted	Incorrectly Predicted	
		(1)	(2)		(0)	(2)		(0)	(1)
Support Vector Machine	642	0	0	9	4	0	43	0	0
Decision Tree	642	0	0	10	2	1	41	2	0
Naïve Bayes	642	0	0	0	13	0	7	36	0

Fig. 8. Confusion Matrix of all models with 80:20 split

SVM demonstrates high accuracy in predicting class 0, with 642 instances correctly classified. For class 1, it correctly predicts 9 instances but misclassifies 4 instances as "Both Not Exist (0)". Notably, SVM shows strong performance in predicting class 2 with 43 instances correctly classified and no misclassifications into class 1.

The Decision Tree model also performs well in predicting class 0 with 642 correctly classified instances. It predicts 10 instances correctly for class 1 but misclassifies 2 instances as "Both Not Exist (0)". For class 2, it correctly predicts 41 instances but has 2 misclassifications into class 0. Overall, the Decision Tree shows fewer misclassifications for class 0 compared to SVM.

Naïve Bayes has an equal number of correct predictions for class 0, with 642 instances. However, it struggles significantly with class 1, correctly predicting none and misclassifying 13 instances as "Both Not Exist (0)". For class 2, it correctly predicts only 7 instances and misclassifies 36 instances as "Symptom(s) Exist (1)". This indicates that Naïve Bayes struggles with the complexity of the dataset.

The SVM and Decision Tree models show the best overall performance, with high accuracy in predicting all labels and minimal misclassifications. SVM performs exceptionally well for class 2 with no misclassifications, while Decision Tree has slightly more balanced predictions across all classes. Naïve Bayes, on the other hand, has the highest number of misclassifications, particularly struggling with class 1 and class 2, indicating it may not be the best model for this dataset.

In conclusion, based on the confusion matrix results, SVM and Decision Tree are the top-performing models, demonstrating high accuracy and balanced predictions across all classes. Naïve Bayes underperforms significantly, suggesting it is less suited for this particular dataset. The strengths of SVM and Decision Tree in handling well-defined boundaries and capturing complex patterns make them the preferable choices for accurate classification in this scenario.

VI. Conclusion

In conclusion, the comparative evaluation highlighted that SVM, with appropriate hyperparameter tuning, provided the best performance in the text classification of breast cancer symptoms and treatments using journal articles from the PubMed website. The best results were observed when the data was split 90% for training and 10% for testing. SVM demonstrated the highest overall accuracy and balanced performance across all metrics and classes, making it the most effective classifier for this dataset. This was followed by Decision Tree, showing strong performance but slightly less effective than SVM. The worst performing model was Naïve Bayes. Although most of the proposed machine learning algorithms have proven capable of classifying text data based on the keywords defined for breast cancer, there are areas for future improvement. These include exploring different feature extraction methods, implementing hyperparameter tuning using techniques like GridSearchCV and RandomizedSearchCV, and addressing data imbalance using techniques such as MLSMOTE.

Acknowledgement

The author would like to express gratitude and appreciation to Faculty of Computing, Universiti Teknologi Malaysia (UTM) for their support in this research.

References

[1] Afolabi, I. (2008). Knowledge Discovery in Online Repositories: A TextMining approach. https://www.semanticscholar.org/paper/Knowledge-Discovery-in-Online-Repositories%3A-A-Afolabi-Musa/7c1e3e9690b080516dffc5655b1f77b0fe83e795.

[2] Bogatinovski, J., Todorovski, L., Džeroski, S. and Kocev, D. (2022) 'Comprehensive comparative study of multi-label classification methods', Expert Systems with Applications, 203. doi:10.1016/j.eswa.2022.117215.

[3] Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., and Ijaz, M. F. (2022). A complete process of text classification system using state-of-the-art NLP models. *Computational Intelligence and Neuroscience*, pp. 1–26.

[4] Guia, M., Silva, R. R., & Bernardino, J. (2019). Comparison of naïve bayes, support vector machine, decision trees and random forest on sentiment analysis. https://doi.org/10.5220/0008364105250531.

[5] Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification. *Artificial Intelligent Review*, 52, pp. 273–292.

[6] Nasa, D. (2012). Text Mining Techniques- A Survey. https://www.semanticscholar.org/paper/Text-Mining-Techniques-A-Survey-Nasa/ec73a3cc95143dc7935ce0272174e210fa64122b.

[7] Shaikh, R., Rafi, M., Mahoto, N. A., Sulaiman, A. and Shaikh, A. (2023) 'A filter-based feature selection approach in multilabel classification', Machine Learning: Science and Technology, 4(4). doi:10.1088/2632-2153/ad035d.

[8] Smith, L. C., & Yu, B. (2008). An evaluation of text classification methods for literary study. Literary and Linguistic Computing, 23(3), 327–343. https://doi.org/10.1093/llc/fqn015

[9] Tandel, S. S., Jamadar, A., & Dudugu, S. (2019). A survey on text mining techniques. https://doi.org/10.1109/icaccs.2019.8728547

[10] Thakur, K., & Kumar, V. (2021). Application of Text Mining Techniques on Scholarly Research Articles: Methods and Tools. New Review of Academic Librarianship, 28(3), 279–302. https://doi.org/10.1080/13614533.2021.1918190

[11] Zhang, M. -L. and Zhou, Z. -H. (2014). A Review on Multi-Label Learning Algorithms. *EEE Transactions on Knowledge and Data Engineering*, 26(8), pp.1819-1837.