Journal of Theoretical and Applied Information Technology
29th February 2024. Vol.102. No 4

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MULTI-LABEL INTENT CLASSIFICATION FOR EDUCATIONAL CHATBOT: A COMPARATIVE STUDY USING PROBLEM TRANSFORMATION, ADAPTED ALGORITHM AND ENSEMBLE METHOD

AZIZAN ISA[1], WAN MOHD AMIR FAZAMIN WAN HAMZAH[2], MOHD KAMIR YUSOF[3], ISMAHAFEZI ISMAIL[4], MOKHAIRI MAKHTAR[5]

[1]Computer Science Unit, Kelantan Matriculation College, Malaysia
[2,3,4,5]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Malaysia

E-mail: [1]azizan@kmkt.matrik.edu.my, [2*]amirfazamin@unisza.edu.my, [3]mohdkamir@unisza.edu.my, [4]ismahafezi@unisza.edu.my, [5]mokhairi@unisza.edu.my

ABSTRACT

This article presents a comparative study of multi-label intent classification for educational chatbots using three machine learning (ML) techniques: problem transformation, adapted algorithm and ensemble method. In the context of chatbots, user intent can be complex, potentially spanning multiple areas simultaneously. Current single-label intent classification techniques often fail to handle such intricate user intentions. Thus, an in-depth study of multi-label intent classification was conducted, critically analysing the performance of these techniques based on evaluation metrics such as accuracy, hamming loss, precision, recall and F1-score. The results highlighted the superiority of the problem transformation technique, particularly the label powerset method, over the other two methods across all evaluation metrics. Significantly, the label powerset methodology demonstrated remarkable performance with a substantial accuracy rate of 0.9669 and a minimal hamming loss of 0.0132, showcasing its efficacy in handling tasks associated with multi-label intent classification. The adapted algorithm and ensemble method displayed positive results but did not surpass the problem transformation technique. This study offers valuable insights for researchers and developers seeking to design an efficient and accurate intent classification for educational chatbots.

Keywords: *Educational chatbot, Classification, Problem Transformation, Adapted Algorithm, Ensemble Method.*

1. INTRODUCTION

Chatbots, also known as conversational agents, have emerged as powerful tools in the field of education, harnessing the capabilities of artificial intelligence (AI) and natural language processing (NLP) to effectively replicate human dialogues and engagements. An intent is a broad description of what a chatbot user is trying to say. Chatbots can generate initial replies using machine learning algorithms or employing various heuristic techniques to choose responses from a pre-existing library [1-2]. The user's intent influences the response generated by the chatbot during the interaction.

Intent classification in chatbots entails mapping user queries to predefined intents, enabling the system to provide appropriate responses or actions. However, conventional intent classification approaches primarily address single-label classification problems, which limits their efficacy in handling intricate educational scenarios where multiple intents may coexist. Several responses also often do not meet the user's intentions, making the chatbot unable to respond correctly. Consequently, chatbot responses may belong to the wrong intent label [3]. The issue with multi-label intent data is that instances could theoretically belong to more than one class. The overlapping of many labels causes the borders to become hazy [4]. Furthermore, the lack of knowledge on how multi-label intent classification tasks should be carried out adds to these problems.

Several techniques to implement multi-label intent classification have been studied to address this challenge: problem transformation, adapted algorithms and ensemble method [5-10]. However, a

comprehensive comparative study that systematically evaluates the performance of these techniques in the specific context of educational chatbots is yet to be conducted. Therefore, an application of problem transformation was explored, which involved converting the multi-label intent classification problem into a series of subproblems focused on single-label classification. Dynamically incorporating adapted algorithms would adjust their behaviour based on the data characteristics and learning processes. This algorithm can potentially adapt to evolving user patterns and improve the intent classification accuracy over time. In addition, an ensemble method that combines the predictions of multiple intent classifiers was explored to achieve superior performance. The effectiveness of this ensemble method in improving the accuracy of multi-label intent classification of chatbots was evaluated.

Through a comparative study, this study aims to provide valuable insights into the performance and suitability of various machine-learning techniques for multi-label intent classification in chatbots, such as problem transformation, adaptive algorithms and ensemble methods. These findings would contribute to the advancement of chatbots and offer practical guidance for developers and researchers in designing more accurate and efficient intent classification systems for educational chatbots. This solution describes multi-label intent classification methods based on machine learning (ML) algorithms. The best method is determined based on the analysis results, including accuracy, hamming loss, precision, recall, and F-1 score.

2. RELATED WORKS

This section provides a detailed description of the educational chatbot, classification and multi-label intent classification. The explanation also includes previous related studies conducted by researchers.

2.1 Educational Chatbot

Chatbots have been used in the educational field as dialogic teaching facilitators since the early 1970s [11]. There is a growing trend in utilising educational chatbots, mainly owing to their capacity to deliver a cost-efficient way of engaging learners and offering a tailored learning journey [12]. The relevance of chatbot integration becomes particularly significant in online education, where chatbots can be employed as virtual teaching assistants to handle simple queries from students and offer responses around the clock, even when human staff are unavailable. Chatbots have been increasingly used in education to revolutionise teaching and learning practices. They have the potential to offer personalised learning experiences, improve student engagement and reduce the workload of educators [13].

Chatbots are useful for delivering personalised content. They can adapt to each student's unique learning style, pace and knowledge level, making learning more efficient and enjoyable [14]. Using algorithms and AI, chatbots can track student progress, identify weak areas and adjust educational content accordingly [15-16]. Moreover, chatbots are instrumental in increasing students' engagement. They can provide instant feedback, answer queries at any time, and promote interactive learning through conversation, making students more involved in the learning process [17]. Furthermore, chatbots can significantly reduce teachers' workload. By automating routine tasks, such as answering frequently asked questions or providing feedback on straightforward tasks, chatbots allow free time for teachers to focus on more complex pedagogical activities.

Despite their potential, several difficulties were encountered when using chatbots in educational settings. The primary challenge is in guaranteeing accurate responses considering the user's context and intention. Chatbots must understand natural language and context to deliver valuable and relevant replies. A vital feature of chatbot operations is the precise determination and classification of the user's intent, enabling the chatbots to understand and answer user questions correctly. Usually, NLP and ML algorithms are used to overcome these challenges [13][17].

2.2 Classification

Classification represents an ML process wherein ML algorithms are utilised to designate a class label to examples derived from a particular problem domain. It represents a predictive modelling problem in which the class label results from an input [5-7]. The classification process assigns data elements to their respective classes and shows an optimal performance when the output value is finite and discrete. Classification problems frequently encounter situations involving multiple class labels. In the realm of multiclass classification, a sample can be exclusively categorised under a single label. However, within the scope of multi-label classification, a sample could potentially be linked with several labels simultaneously.

Single-label classification is associated with only a single class label. In a specific classification problem, labels are associated with a hierarchical

structure [5]. In single-label classification, a model is trained to predict one label (or class) for each instance from a set of mutually exclusive labels, L. It means that each instance in the dataset is associated with one label, and only one label is associated with L. The predictive function in a single-label classification involves assigning the most appropriate label from set L to a given instance, which is performed based on learning from the training dataset where each instance is already associated with a known label. The goal is to generate a model that can accurately predict the labels of new unseen instances.

In multi-label classification, each instance may be simultaneously linked to several labels. This classification model aims to develop a predictive function capable of assigning an accurate subset of labels to each instance [6-7]. The subset of labels, represented by Y, is extracted from the complete set of possible labels L and is mathematically expressed as $Y \subseteq L$. In other words, the algorithm's task for each instance is to identify the most suitable subset of labels from the overall set L. Such classification is critical in various real-world situations where a single instance must be identified or categorised under multiple classes or labels simultaneously.

The predictive function used in multi-label classification often relies on complex ML algorithms, including, but not limited to, problem transformation, adapted algorithm and ensemble methods. These algorithms are designed to handle the inherent complexity of instances belonging to multiple classes, thus allowing them to deliver more realistic and flexible solutions for such problems. Multi-label classification has received increased attention and has been applied to various domains, including text classification [7], image classification [8], scene and video classification [9] and bioinformatics [10]. Multi-label classification is advantageous compared to commonly used classification approaches as it associates each occurrence with one or more classes or labels. Multi-label classification is concerned with learning from a set of instances associated with a set of labels; one instance may be linked with numerous labels simultaneously.

2.3 Multi-label Intent Classification

The classification of intent is a type of text categorisation that fulfils the same role and is viewed as an issue in the realm of NLP. This involves assigning class labels to units, such as sentences, paragraphs or documents. Intent classification involves the automatic grouping of text data based on user objectives. ML and NLP automatically link words or phrases to a particular intent. Furthermore, intent classification refers to the process of determining the objective of any given utterance within a task-oriented dialogue system [18]. Within chatbot conversational systems, intent classification is often used to select responses. An intent classifier is expected to associate an input utterance with the correct intent and identify when the utterance is unrelated to any of the intents.

Multi-label intent classification allows datasets with more than one target variable to be categorised [19]. With multi-label intent classification, several labels become the outputs for a specific prediction. A particular input can be associated with multiple labels during the prediction process. The model must be evaluated based on ML techniques to construct the model for this study. Figure 1 presents the techniques used in multi-label intent classification, including problem transformation, adapted algorithms and ensemble methods. The most suitable techniques are chosen based on their accuracy and the quality of the results.

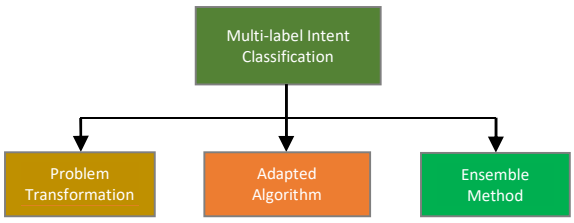

Figure 1: The Multi-Label Intent Classification Techniques

Problem transformation is a technique in ML that transforms the multi-label issue into one or multiple single-label problems. It functions through a set of transformations that reconfigure the initial problem. The three main methods under problem transformation are binary relevance, label powerset, and classifier chains. Binary relevance approaches each label as a binary problem, ignoring any potential relationships between the labels. However, the label powerset views each unique set of labels as a distinct class, considering label correlations, but often faces challenges with data sparsity. A classifier chain is a method that forms binary classifiers in a sequence; each new link in the chain uses the previous ones as additional features, thus considering label correlations [20].

The adapted algorithm technique modifies a traditional ML algorithm to deal with multi-label data directly, negating the need to transform the problem into single-label tasks. This adjustment is

generally specific to the algorithm, with multi-label k-nearest neighbours (MLkNN) and multi-label decision trees acting as the common examples. These methods preserve the structure of the original data and inherently consider label correlations; however, they are often more complex than problem-transformation techniques [21].

Ensemble methods combine multiple ML models, known as base learners, to improve the overall performance. These methods can be applied to single- and multi-label problems. In a typical ensemble, individual models make independent predictions, while the final prediction is made by a process of voting (for classification) or averaging (for regression). Ensemble methods include bagging, boosting and stacking. These methods are generally robust to overfitting and often perform better than the individual models [22][40].

3. METHOD

Multi-label intent classification involves accurately assigning a label to an input recognised as a natural language expression drawn from an established collection of intents. The ML model forecasts a classification corresponding to a specific intent in this process. Multi-label intent classification can be implemented using various methods or strategies. The choice of these methods generally depends on characteristics, including the nature of the dataset and the quantity of data. Figure 2 illustrates the systematic process for conducting multi-label intent classification using ML algorithms, which involves data collection, exploratory data analyses, data preprocessing, dataset splitting, model building and evaluation metrics [23-24].

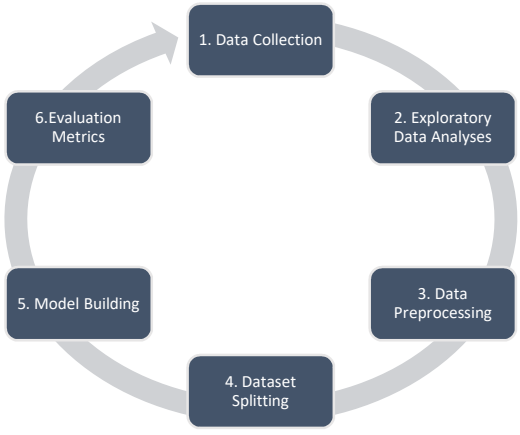

Figure 2: Systematic Process of Multi-Label Intent Classification using ML Algorithms

3.1 Data Collection

The dataset employed in this study was a predefined intent associated with the Web Application Development course chatbot. The intents encompassed topics, namely HTML, JavaScript, Jsp, servlet, and mysql. An intent dictionary object was formulated to represent the intent classes as numerical values.

3.2 Exploratory Data Analyses

Exploratory Data Analysis (EDA) is a crucial step in data analysis that entails examining and understanding data before applying any modelling or statistical techniques [25-26]. In the context of checking for missing values, calculating the number of intents under each label and determining the number of intents with multiple labels, EDA helps to uncover important insights. The first step, checking for missing values, involves identifying incomplete or null data points in the dataset to ensure data quality and inform subsequent data-handling strategies.

Second, calculating the number of intents under each label involves quantifying the frequencies of different labels or categories in the dataset. This information provides an overview of the distribution and relative importance of each intent. Finally, determining the number of intents with multiple labels helps identify instances where a single data point is associated with multiple labels. The analysis sheds light on the complexity and potential overlaps in the dataset, aiding subsequent data modelling or classification tasks. Overall, EDA plays a vital role in understanding data characteristics, identifying data issues, as well as informing further data processing and analysis decisions.

3.3 Data Preprocessing

Data preprocessing is a crucial stage in developing an ML model, where its effectiveness determines its success [27-28]. In this study, since the data analyses the user's intent in the form of text data from a chatbot, text preprocessing techniques were employed. Text preprocessing involves cleaning and preparing text data for NLP tasks to transform the text into a more understandable format that enhances the performance of ML algorithms. The specific steps involved in text preprocessing include intent-noise scanning and removing stop words.

3.3.1 Intent Noise

The intent noise scan removes punctuation marks, characters, digits, and pieces of text that can interfere with multi-label intent classification.

3.3.2 Removing Stop Words

Stop words are commonly occurring words in any language that contribute little to a text's overall meaning. These include conjunctions, pronouns and articles. Removing stop words enabled the model to focus on more relevant terms for training.

3.4 Dataset Splitting

Splitting the dataset is an essential step in ML for assessing the performance and generalisability of the model, which is done by dividing the available dataset into distinct subsets for training and testing purposes. This division is typically based on percentage allocation, which determines the proportion of data assigned to each subset. The customary approach involves creating two subsets: training and testing subsets. In this study, the model was trained using the training set, while its performance was evaluated using unseen data from the testing set. The percentage allocation may vary depending on factors such as dataset size, problem complexity and computational resources [29-30].

A commonly employed practice is to allocate a larger percentage of data to the training set, usually approximately 70-80%, with the remaining percentage assigned to the testing set [31-32] to ensure sufficient data for the model to learn from and achieve good generalisation. However, a specific percentage allocation can be adjusted based on the unique requirements and constraints of the problem. In the original data, the features (X) and targets (y) were split into a training set (70%) and a testing set (30%). As a rough guideline, reserving 30% of the dataset for testing is reasonable.

3.5 Model Building

At this stage, the model for multi-label intent classification is built using techniques such as problem transformation, adapted algorithms, and ensemble methods. A detailed description of each technique is provided as follows:

3.5.1 Problem Transformation

The problem transformation approach transforms the multi-label classification problem into a binary multi-classification problem. Each binary classifier is responsible for predicting the presence or absence of a particular intent label. The transformation of the problem uses three methods to develop a model: Binary Relevance, Classifier Chains and Label Powerset. All these methods use machine learning algorithm models, such as Decision Trees, k-nearest neighbours (KNN), Multinomial Naive Bayes, Neural Network Multilayer Perceptron (NNMLP) and Random Forest, to determine the best model for multi-label intent classification.

3.5.2 Adapted Algorithm

The multi-label k-nearest neighbours (MLkNN) algorithm was used as an adapted algorithm method for multi-label intent classification. MLkNN is an extension of the k-nearest neighbours (KNN) algorithm, adapted to work with multi-label datasets by treating each label as an independent binary classification problem. The algorithm uses the KNN approach to find the k-nearest neighbours for each data point in the feature space. Then, it estimates the probability of the presence of each label based on the labels of the k-nearest neighbours. The final classification is determined by selecting the most probable label for each instance. The MLkNN method can efficiently handle the task of multi-label intent classification, making it particularly useful when dealing with datasets where multiple intent labels may be associated with each input text.

3.5.3 Ensemble Method

Stacking is used in the ensemble method for multi-label intent classification, which involves the OneVsRest and random forest classifiers. A stacking framework is employed to combine the predictions of multiple models to improve classification performance. The OneVsRest Classifier is utilised as a base classifier, where multiple binary classifiers were trained independently for each intent label.

The Random Forest Classifier acts as the meta-classifier, which learns to combine predictions from the base classifiers. The stacking approach involves training base classifiers on the input data to generate predictions for each intent label. These predictions and the original features are input to the Random Forest classifier, producing the final predictions for the multi-label intents. By leveraging the strengths of the OneVsRest Classifier and Random Forest Classifier within the stacking framework, this ensemble method aims to enhance the accuracy and robustness of multi-label intent classification.

3.6 Evaluation Metrics

Evaluation metrics play a crucial role in assessing the performance of multi-label intent classification models [6][33]. A comparative analysis compares the results of the problem transformation, adapted algorithm and ensemble methods. For the problem transformation technique, a comparison is first made against the binary relevance, classifier chain and label powerset methods. The analysis includes accuracy, hamming loss, precision, recall and F1-score.

3.6.1 Accuracy Score

The accuracy score measures correctly predicted the proportion of labels to the total number of labels.

It provides an overall measure of how well the model correctly predicts all the labels. However, accuracy might not be the most informative metric for imbalanced multi-label datasets.

Accuracy = (number of correctly predicted labels) / (total number of labels) (1)

3.6.2 Hamming Loss

The hamming loss calculates the fraction of incorrectly predicted labels, that is, the average number of incorrect labels per instance. It considers false positives and false negatives, making it suitable for evaluating multi-label classification, where multiple labels can be assigned to each instance.

Hamming loss = (number of incorrectly predicted labels) / (total number of labels) (2)

3.6.3 Precision

Precision measures the proportion of correctly predicted positive labels relative to the total number of predicted positive labels. It provides insight into the model's ability to identify relevant labels among the predicted positive labels correctly.

Precision = (number of true positive labels) / (number of predicted positive labels) (3)

3.6.4 Recall

Recall, also known as the sensitivity or true positive rate, measures the proportion of correctly predicted positive labels to the total number of actual positive labels, representing the ability of the model to capture all relevant positive labels.

Recall = (number of true positive labels) / (number of actual positive labels) (4)

3.6.5 F1-Score

The F1-score is the harmonic mean of precision and recall. It provides a balanced evaluation metric by considering precision and recall. The F1-score gives equal importance to precision and recall and is useful when the dataset is imbalanced or when both metrics are equally important.

F1-score = 2 × (Precision × Recall) / (Precision + Recall) (5)

These evaluation metrics provide different perspectives on the performance of multi-label intent classification models. Accuracy assesses overall correctness, hamming loss measures label-level errors, precision focuses on positive predictions, recall evaluates positive instance coverage, whereas the F1-score combines precision and recall into a single value for a balanced assessment.

4. RESULTS AND DISCUSSION

A dataset containing 402 unique chatbot intents was explicitly collected for training and testing a multi-label intent classification model. Each intent represented a specific user intention or task that the model aims to predict based on the input text. The dataset comprised text samples or utterances labelled with one or more intent labels, indicating the presence of intents in each sample. These labels allowed the model to train and predict multiple intents simultaneously. The labels for each intent were identified as HTML, JavaScript, Jsp, servlet and mysql.

4.1 Descriptive Statistics

Table 1 presents descriptive statistics for the different labels (html, javascript, jsp, servlet, and mysql). The “Count” column indicates the number of intents associated with each label. There are 163 intents labelled as “html,” 66 intents labelled as “javascript,” 257 intents labelled as “jsp,” 112 intents labelled as “servlet,” and 109 intents labelled as “mysql.” The “Mean” column indicates the average presence of each label in the dataset. The mean presence of “html” is 0.405473, “javascript” is 0.164179, “jsp” is 0.639303, “servlet” is 0.278607, and “mysql” is 0.271144.

These values represent the average occurrence or likelihood of each label in a dataset. The "Std" column represents the standard deviation of label presence across the dataset. It provides a measure of the variation or spread of label values. The standard deviation for “html” is 0.491595, “javascript” is 0.370900, “jsp” is 0.480801, “servlet” is 0.448872, and “mysql” is 0.445104. Higher values indicate greater variability in the presence of labels among the instances.

Table 1: Descriptive Statistics

Label	Count	Mean	Std
html	163	0.405473	0.491595
javascript	66	0.164179	0.370900
jsp	257	0.639303	0.480801
servlet	112	0.278607	0.448872
Mysql	109	0.271144	0.445104

4.2 Exploratory Data Analysis and Data Preprocessing

Table 2 displays intents and labels in the obtained dataset used for multi-label classification. In this context, each row represents an "intent or a particular task that a user may want to perform. Each column

after the first represents a "label." These labels are likely topics related to the coding languages or techniques: html, javascript, jsp, servlets and mysql. Label = "1" indicates that the intent has the label, while label "0" indicates that the intent does not have the label. Each intention may have one or more labels.

Table 2: Intents and Labels

Intents	Label (label = 1, no label = 0)				
	html	javascript	jsp	servlet	mysql
code of jsp request implicit object	0	0	1	1	0
code example to create table in mysql and jsp	0	0	1	0	1
code example of jsp scriptlet tag	1	0	1	0	0
give me some sample codes to view users in jsp	0	0	1	1	0
give me some sample codes to define the set of ...	1	0	0	0	0
code of page directive in jsp	1	0	1	0	0
how to get secured against set cookies in the ...	0	0	1	1	0
code to define a table	1	0	0	0	0
give me some sample codes to add user in jsp	0	0	1	0	1
give me some sample codes from input group…	1	0	0	0	0

Table 3 lists a few intents in the first column, while the second column corresponds to a text noise. The noise scan can help to understand how the noise is distributed and how it might affect an NLP model's ability to identify intents correctly [34-35].

Table 3: Text Noise

No.	Text Noise
0	{'text_noise': 6.0606060606060606, 'text_lengt...
1	{'text_noise': 8.0, 'text_length': 25, 'noise_...
2	{'text_noise': 11.11111111111111, 'text_length...
3	{'text_noise': 5.88235294117647, 'text_length'...
4	{'text_noise': 7.6923076923076925, 'text_lengt...
……..	……..
397	{'text_noise': 8.771929824561402, 'text_length...
398	{'text_noise': 10.0, 'text_length': 40, 'noise...
399	{'text_noise': 4.166666666666666, 'text_length...
400	{'text_noise': 8.19672131147541, 'text_length'...
401	{'text_noise': 9.090909090909092, 'text_length...
……..	…….

Table 4 displays the intents after the removing stop words process was performed. The first column, "No", represents the identification number for each intent. The second column, "Intents", contains the actual text of intent. This text removed stop words, as it consisted mostly of content-rich words that could be key to understanding intent. By doing this, the machine learning model can focus on the most important words and is less likely to be distracted by common but uninformative words. This can make the model more efficient and improve its performance in multi-label intent classification [36].

Table 4: Intents after Removing the Stop Words

No.	Intents
0	code example paragraph html
1	code paragraph html
2	sample code paragraph html
3	code example line break html
4	code line break html
……..	.....
397	sample code implement login system jsp
398	implement login system jsp?
399	code implementing login system servlet

400	sample code implements login system servlet
401	implement login system servlet?
……	……..

4.3 Dataset Splitting and Model Building

The dataset was divided into training and test sets. The proportion used was 70% for the training set and 30% for the testing set [37-38]. Models for multi-label intent classification were built using techniques such as problem transformation, adapted algorithms and ensemble methods. Table 5 outlines the evaluation metrics of the problem transformation using various ML algorithms. The methods used were binary relevance (BR), classifier chains (CC), and label powerset (LP). Several evaluation metrics were reported for each algorithm and method, including the accuracy score, hamming loss, precision, recall and F1-score for each class labelled as follows: Class 0 (C0), Class 1 (C1), Class 2 (C2), Class 3 (C3) and Class 4 (C4).

The Decision Trees (DT) algorithm achieved the highest accuracy score of 0.9587 using the label powerset method, with the lowest hamming loss of 0.0165. The precision, recall, and F1- scores were the highest for classes 1 and 2 across all methods, indicating that the model performed best in these classes. K-nearest neighbours (KNN) performed slightly worse than DT, with the highest accuracy score of 0.8347 (label powerset) and a somewhat higher hamming loss of 0.0711. Notably, the precision, recall and F1-scores were all lower for KNN across all classes, suggesting that the algorithm has more difficulty distinguishing between classes than DT.

The performance of the Multinomial Naive Bayes (MNNB) algorithm was similar to that of KNN, with the highest accuracy score of 0.8512 (classifier chains) and a slightly higher hamming loss of 0.0661. Interestingly, despite the algorithm's generally lower performance, its precision for classes 0 and 4 was perfect when using the binary relevance and classifier chain methods. The Neural Network Multilayer Perceptron (NNMLP) performed quite well, with an accuracy score of 0.9256 (for classifier chains and label powerset) and a moderate hamming loss of 0.0298. The precision and recall for classes 0, 1, 2 and 4 were perfect across all methods, indicating the model's strong performance in these classes.

The Random Forest (RF) algorithm performed best for the problem transformation technique, with the highest accuracy score of 0.9669 using the label powerset and the lowest hamming loss of 0.0132. Meanwhile, the precision for classes 0, 1, 2 and 4 was perfect across all the methods. The recall was also high across all classes, suggesting that the algorithm effectively identified true-positive examples in the data. These results proved that the best results can be obtained using a combination of a power set of labels as a problem transformation technique, as in previous studies by Kumar *et al.* [41], Amr [42] and Khan [43].

Table 5: Evaluation Metrics of the Problem Transformation Techniques with Various ML Algorithms

ML	Evaluation Metrics		Problem Transformation		
			BR	CC	LP
DT	Accuracy Score		0.9091	0.9256	0.9587
	Hamming Loss		0.0182	0.0165	0.0165
	Precision	C0	0.9800	1.0000	0.9800
		C1	1.0000	1.0000	1.0000
		C2	1.0000	1.0000	1.0000
		C3	0.8100	0.8400	0.9100
		C4	1.0000	1.0000	0.9700
	Recall	C0	1.0000	1.0000	1.0000
		C1	1.0000	1.0000	1.0000
		C2	1.0000	1.0000	1.0000
		C3	0.8700	0.8700	0.9700
		C4	1.0000	0.9700	0.8900
	F-1 Score	C0	0.9900	1.0000	0.9900
		C1	1.0000	1.0000	1.0000
		C2	1.0000	1.0000	1.0000
		C3	0.8400	0.8500	0.9400
		C4	1.0000	0.9900	0.9300
KNN	Accuracy Score		0.7934	0.8017	0.8347
	Hamming Loss		0.0727	0.0826	0.0711
	Precision	C0	0.9800	0.9800	0.9600
		C1	0.6700	0.6500	0.6700
		C2	0.9700	0.9400	0.9700
		C3	0.8200	0.8000	0.8200
		C4	0.8900	0.8200	0.8900
	Recall	C0	0.8600	0.8600	0.9100
		C1	0.9200	1.0000	0.9200
		C2	0.9000	0.9000	0.8900
		C3	0.7700	0.8000	0.7700
		C4	0.9100	0.8900	0.9100
	F-1 Score	C0	0.9200	0.9200	0.9400
		C1	0.7700	0.7900	0.7700
		C2	0.9300	0.9200	0.9300
		C3	0.7900	0.8000	0.7900
		C4	0.9000	0.8500	0.9000
MNNB	Accuracy Score		0.8017	0.8512	0.8347
	Hamming Loss		0.0479	0.0446	0.0661
	Precision	C0	0.9800	0.9800	1.0000
		C1	1.0000	0.9200	0.8900
		C2	0.9400	0.9400	0.8600
		C3	0.8400	0.8400	0.8000
		C4	1.0000	1.0000	0.9700
	Recall	C0	0.9300	0.9300	0.9800
		C1	0.7700	0.8500	0.6200
		C2	0.9600	0.9300	0.9600
		C3	0.7000	0.9000	0.8000
		C4	0.9700	0.9400	0.8300
	F-1 Score	C0	0.9500	0.9500	0.9900

		C1	0.8700	0.8800	0.7300
		C2	0.9500	0.9400	0.9100
		C3	0.7600	0.8700	0.8000
		C4	0.9900	0.9700	0.8900
NN MLP	Accuracy Score		0.9174	0.9256	0.9256
	Hamming Loss		0.0264	0.0298	0.0298
	Precision	C0	1.0000	1.0000	1.0000
		C1	0.9300	0.9300	0.9300
		C2	1.0000	1.0000	1.0000
		C3	0.7800	0.7800	0.7800
		C4	1.0000	1.0000	1.0000
	Recall	C0	1.0000	1.0000	1.0000
		C1	1.0000	1.0000	1.0000
		C2	0.9300	0.9300	0.9300
		C3	0.9300	0.9700	0.9700
		C4	1.0000	0.9100	0.9100
	F-1 Score	C0	1.0000	1.0000	1.0000
		C1	0.9600	0.9600	0.9600
		C2	0.9600	0.9600	0.9600
		C3	0.8500	0.8700	0.8700
		C4	1.0000	0.9600	0.9600
RF	Accuracy Score		0.9421	0.9587	0.9669
	Hamming Loss		0.0132	0.0132	0.0132
	Precision	C0	1.0000	1.0000	1.0000
		C1	1.0000	1.0000	1.0000
		C2	1.0000	1.0000	1.0000
		C3	0.9700	0.9100	0.8800
		C4	1.0000	1.0000	1.0000
	Recall	C0	0.9500	0.9600	1.0000
		C1	1.0000	1.0000	1.0000
		C2	1.0000	1.0000	1.0000
		C3	0.9300	1.0000	1.0000
		C4	0.9400	0.9100	0.8900
	F-1 Score	C0	0.9700	0.9800	1.0000
		C1	1.0000	1.0000	1.0000
		C2	1.0000	1.0000	1.0000
		C3	0.9500	0.9500	0.9400
		C4	0.9700	0.9600	0.9400

The multi-label k-nearest neighbours (MLkNN) is an adapted algorithm for multi-label intent classification. Improved performance can be achieved by adapting the standard MLkNN algorithm to fit the task better [44]. Table 6 presents the evaluation metrics of the proposed algorithm. This table presents the performance of a machine learning model trained using an adapted algorithm technique for a multi-label intent classification problem. The accuracy score was 0.7686, indicating that this model correctly classified approximately 76.86% of the instances. It is a general measure of how often a model is correct. In contrast, the hamming loss was 0.0744. The hamming loss measures the fraction of incorrect labels to the total number of labels. Thus, this model incorrectly classified approximately 7.44% of instances.

The model demonstrated high precision across all classes, ranging from 0.86 to 0.96. This indicates that when the model predicts a class, it is usually correct. However, the recall values varied, with values ranging from 0.75 to 0.94, showing the model's ability to capture relevant instances. The F1-score, balancing precision and recall, also varied across classes, with values ranging from 0.82 to 0.94. Overall, the model appeared to perform well across classes, demonstrating strong precision and recall, though individual class performance may vary, suggesting potential areas for improvement, particularly in recall for class 0 and precision for class 1.

Table 6: Evaluation Metrics of the Adapted Algorithm

Technique	**Evaluation Metrics**		
Adapted Algorithm	Accuracy Score		0.7686
	Hamming Loss		0.0744
	Precision	C0	0.9600
		C1	0.8600
		C2	0.9300
		C3	0.8800
		C4	0.9100
	Recall	C0	0.7500
		C1	0.9200
		C2	0.9400
		C3	0.7700
		C4	0.8900
	F-1 Score	C0	0.8400
		C1	0.8900
		C2	0.9400
		C3	0.8200
		C4	0.9000

The OneVsRest Classifier and the Random Forest Classifier were employed in the ensemble method for multi-label intent classification [45]. A stacking framework was used to integrate the predictions of numerous models to improve the classification performance. Table 7 presents the evaluation metrics for the ensemble method. The table presents the evaluation results of a machine learning model that used an ensemble method for a multi-label intent classification problem. Starting with an accuracy score of 0.8760, it means that the model correctly predicted the class for approximately 87.6% of the instances. This value represents the overall success of the model predictions. The hamming loss was 0.0298. The hamming loss refers to the fraction of incorrect labels to the total number of labels. Thus, the model made incorrect predictions in approximately 2.98% of cases.

Precision measured the accuracy of positive predictions, showing high values of 1.0000 for class 0, class 1 and class 4, as well as strong values for class 2 (0.9700) and class 3 (0.9200). Recall, indicating the proportion of actual positives captured by the model, displayed high scores for most classes,

notably for class 1 (1.0000), class 2 (0.9900) and class 0 (0.9500). These scores demonstrated the model's ability to make accurate positive predictions and effectively identify true positives for most classes. The F1-scores, which combine precision and recall into a single metric, further confirmed the model's balanced performance across different classes, with scores ranging from 0.8600 to 1.0000, affirming the model's strong overall predictive capability and balanced trade-off between precision and recall for most classes.

Table 7: Evaluation Metrics of the Ensemble Method

Technique	Evaluation Metrics		
Ensemble Method	Accuracy Score		0.8760
	Hamming Loss		0.0298
	Precision	C0	1.0000
		C1	1.0000
		C2	0.9700
		C3	0.9200
		C4	1.0000
	Recall	C0	0.9500
		C1	1.0000
		C2	0.9900
		C3	0.8000
		C4	0.8900
	F-1 Score	C0	0.9700
		C1	1.0000
		C2	0.9800
		C3	0.8600
		C4	0.9400

Figure 3 compares the multi-label intent classification accuracy scores obtained using the problem transformation, adapted algorithm and ensemble method techniques.

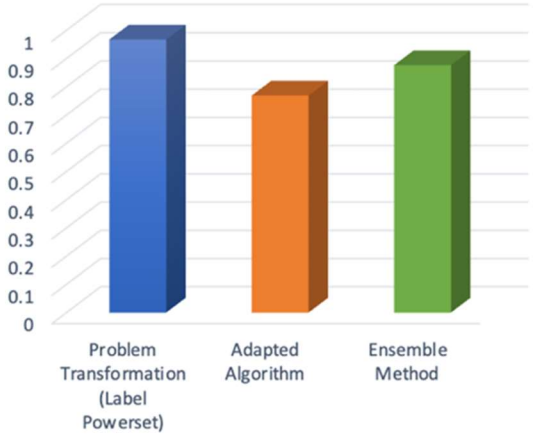

Figure 3: Accuracy Score

An accuracy score is the simplest way to measure the performance of a model. It denotes how often the prediction of the model is correct. In this case, the problem transformation (label powerset) performed best, with an accuracy score of 0.9669, followed by the ensemble method at 0.8760 and the adapted algorithm at 0.7686.

Meanwhile, the hamming loss is the fraction of wrong labels to the total number of labels. The lower the hamming loss, the better the performance of the model. As shown in Figure 4, the problem transformation (label powerset) model has the lowest hamming loss of 0.0132, suggesting that it made the least number of mistakes among the three. The ensemble method followed with 0.0298, while the adapted algorithm performed the worst in this metric, at 0.0744.

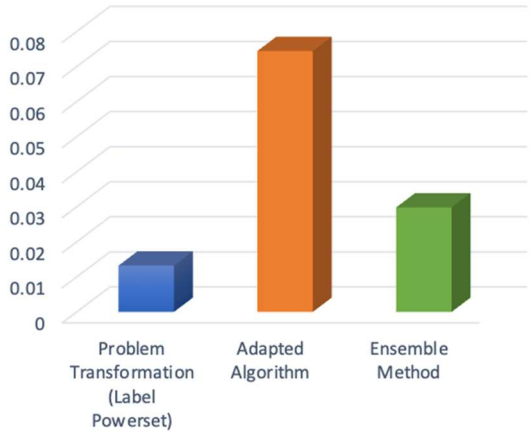

Figure 4: Hamming Loss

Figure 5 shows the problem transformation (label powerset) model with 1.0000 precision for classes 0, 1, 2 and 4 and 0.8800 for class 3, indicating a very high level of correctness in its identifications. Its recall rates were also excellent, standing at 1.0000 for classes 0, 1, 2 and 3, whereas slightly lower at 0.8900 for class 4. F1-scores for the same model were also the highest across all classes. The adapted algorithm performed less on the precision, recall and F1-scores than the other two models. The ensemble method performed better than the adapted algorithm but fell short compared with the problem transformation (label powerset) model.

Based on the evaluation metrics, the problem transformation (label powerset) model outperformed the other two models across all measures and was well-suited for datasets with a limited number of labels [21]. The ensemble method performed better than the adapted algorithm but still fell short of the problem transformation (label powerset) technique. Thus, it may require more resources [39].

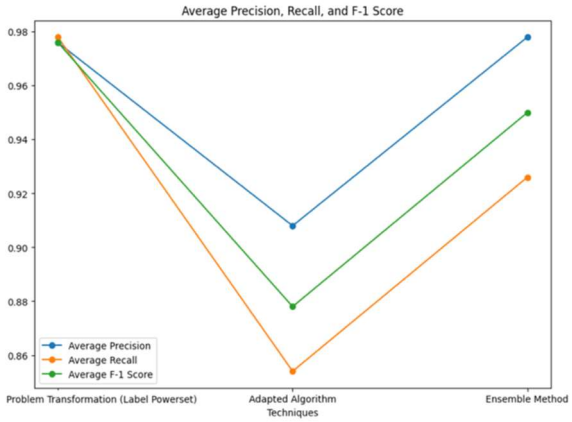

Figure 5: Average Precision, Recall, and F-1 Score

5. CONCLUSION

In this study on multi-label intent classification, three machine learning techniques, namely, problem transformation, adapted algorithm and ensemble method, were compared in terms of their performance across several evaluation metrics, including accuracy, hamming loss, precision, recall, and F1-score. The results indicated that the problem transformation technique, specifically the label powerset method, outperformed the other two methods in all metrics. Notably, the label powerset method using the Random Forest (RF) algorithm achieved a high-precision score of 0.9669 and a low hamming loss of 0.0132. Furthermore, it demonstrated excellent average precision, recall and F1-score values.

These findings emphasised the effectiveness of the problem transformation (label powerset) method [41-43] in addressing multi-label intent classification challenges. Although the adapted algorithm and ensemble method displayed promising results, they fell short compared to the problem transformation approach. Overall, this study highlighted the significance of the problem transformation technique and provided valuable insights for selecting appropriate methods for multi-label intent classification tasks.

However, this comparative study of multi-label intent classification is limited to the use of methods and algorithms that have been identified earlier. For problem transformation techniques, only machine learning algorithm models, such as Decision Trees, k-nearest neighbours (KNN), Multinomial Naive Bayes (MNNB), Neural Network Multilayer Perceptron (NNMLP) and Random Forest (RF), were used. The algorithm used for the adapted algorithm technique was the MLkNN. Likewise, only the stacking method was used for the Ensemble Method, which involved the OneVsRest and random forest classifiers.

Future research could explore a broader range of algorithms within the problem transformation method to identify potential alternatives or enhancements that may further improve multi-label intent classification. Conducting more extensive hyperparameter tuning for the adapted algorithm and ensemble methods could also help uncover their full potential and improve their performance.

ACKNOWLEDGEMENT

This work is supported by the Fundamental Research Grant Scheme (FRGS), Ministry of Higher Education Malaysia (MOHE), with the project reference code FRGS/1/2020/ICT06/UNISZA/02/3.

REFERENCES:

[1] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017.

[2] Y. Wu, W. Wu, C. Xing, C. Xu, Z. Li, and M. Zhou, "A sequential matching framework for multi-turn response selection in retrieval based chatbots," *Computational Linguistics*, *45*(1), 2019, pp.163-197.

[3] W. M. A. F. W. Hamzah, M. K. Yusof, I. Ismail, M. Makhtar, H. Nawang, and A. A. Aziz, "Multiclass intent classification for chatbot based on machine learning algorithm," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, 2022.

[4] M. U. Ghani, M. Rafi, and M. A. Tahir, "Discriminative Adaptive Sets for Multi-Label Classification," *IEEE Access*, vol. 8, 2020, pp. 227579–227595.

[5] S. Sharma and D. Mehrotra, "Comparative analysis of multi-label classification algorithms," in 2018 *First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018.

[6] G. Tsoumakas and I. Katakis, "Multi-Label Classification: An Overview," *International Journal of Data Warehousing and Mining (IJDWM),* 2007, vol. 3, no. 3, pp. 1–13.

[7] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multi-label classification via calibrated label ranking," *Mach. Learn*., 2008, vol. 73, no. 2, pp. 133–153.

[8] Z. Yan, W. Liu, S. Wen, and Y. Yang, "Multi-label image classification by feature attention

network," *IEEE Access*, 2019, vol. 7, pp. 98005–98013.

[9] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris, and I. Vlahavas, "An empirical study of multi-label learning methods for video annotation," in *2009 Seventh International Workshop on Content-Based Multimedia Indexing*, 2009.

[10] Y. Guo, F.-L. Chung, G. Li, and L. Zhang, "Multi-label bioinformatics data classification with ensemble embedded feature selection," *IEEE Access*, 2019, vol. 7, pp. 103863–103875.

[11] D. Laurillard, *Rethinking university teaching: A conversational framework for the effective use of learning technologies*, 2nd ed. Routledge, 2013.

[12] Benotti, Luciana, Mara Cecilia Martnez, and Fernando Schapachnik, "A tool for introducing computer science with automatic formative assessment," IEEE Transactions on Learning Technologies, 2017, vol 11(2), pp. 179–192.

[13] R. Winkler and M. Soellner, "Unleashing the potential of chatbots in education: A state-of-the-art analysis," *Academy of Management Proceedings.* Vol. 2018, no. 1, p. 15903, 2018.

[14] M. Alavi and D. E. Leidner, "Knowledge management and knowledge management systems: Conceptual foundations and research issues," *MIS Quarterly*, 2001, pp. 107-136.

[15] Gamboa-Cruzado Ja, Menendez-Morales Ch, Del Carpio Cf, López-Goycochea Je, Alva A, Arévalo Cr, "Use of Chatbots In E-Commerce: A Comprehensive Systematic Review," *Journal Of Theoretical and Applied Information Technology*. 2023 Feb 28;101(4).

[16] E. Weerasinghe, T. Kotuwegedara, R. Amarasena, P. Jayasinghe, and K. Manathunga, "Dynamic conversational chatbot for assessing primary students," in *Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium*, Cham: Springer International Publishing, 2022, pp. 444–448.

[17] A. Følstad and P. B. Brandtzæg, "Chatbots and the new world of HCI," *Interactions*, 2017, vol. 24, no. 4, pp. 38-42.

[18] J. Schuurmans and F. Frasincar, "Intent Classification for Dialogue Utterances*," IEEE Intelligence System*, 2020, vol. 35, no. 1, pp. 82–88.

[19] N. Endut, W. M. A. F. W. Hamzah, I. Ismail, M. Kamir Yusof, Y. Abu Baker, and H. Yusoff, "A Systematic Literature Review on Multi-Label Classification based on Machine Learning Algorithms," *TEM Journal*, 2022, pp. 658–666.

[20] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.

[21] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering,* 2014, vol. 26, no. 8, pp. 1819–1837.

[22] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, 2010, vol. 33, no. 1–2, pp. 1–39.

[23] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, "Heart disease identification method using machine learning classification in E-healthcare," *IEEE Access,* 2020, vol. 8, pp. 107562–107582.

[24] P. C. Sen, M. Hajra, and M. Ghosh, "Supervised classification algorithms in machine learning: A survey and review," in *Advances in Intelligent Systems and Computing, Singapore: Springer Singapore*, 2020, pp. 99–111.

[25] J. Tummers, C. Catal, H. Tobi, B. Tekinerdogan, and G. Leusink, "Coronaviruses and people with intellectual disability: an exploratory data analysis," *Journal of Intellectual Disability Research*, 2020, vol. 64, no. 7, pp. 475–481.

[26] Dr. Suma T, Megha C, M. S. Kumar, and M. Jadhav, "Empirical analysis for crime prediction and Forecasting using machine learning and deep learning techniques," *International Journal of Advanced Research in Science, Communication and Technology*, 2022, pp. 60–62.

[27] Azhar Sa, Hidayat F, Azfarezat Mh, "Efficiency of Fake News Detection with Text Classification Using Natural Language Processing," *Journal of Theoretical And Applied Information Technology*. 2023 Nov 30;101(22).

[28] T. Madeira, R. Melício, D. Valério, and L. Santos, "Machine learning and Natural Language Processing for prediction of human factors in aviation incident reports," *Aerospace*, 2021, vol. 8, no. 2, p. 47.

[29] Herlina B, Soeparno H, "Machine Learning Model To Improve Classification Performance In The Process Of Detecting Phishing Urls In Qr Codes," *Journal Of Theoretical And*

Applied Information Technology. 2023 Sep 30;101(18).

[30] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing, 2019.

[31] Nguyen, Q.H., Ly, H.B., Ho, L.S., Al-Ansari, N., Le, H.V., Tran, V.Q., Prakash, I. and Pham, B.T., "Influence of data splitting on performance of machine learning models in prediction of shear strength of soil," *Mathematical Problems in Engineering*, 2021, pp.1-15.

[32] A. M. U. D. Khanday, S. T. Rabani, Q. R. Khan, N. Rouf, and M. Mohi Ud Din, "Machine learning based approaches for detecting COVID-19 using clinical text data," *International Journal of Information Technology*, 2020, vol. 12, no. 3, pp. 731–739.

[33] M. Heydarian, T. E. Doyle, and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE Access*, 2022, vol. 10, pp. 19083–19095.

[34] Al Sharou, Khetam, Zhenhao Li, and Lucia Specia, "Towards a better understanding of noise in natural language processing," in *Proceedings of the Conference Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications*, 2021.

[35] S. Garg, G. Ramakrishnan, and V. Thumbe, "Towards robustness to label noise in text classification via noise modeling," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.

[36] V. Raina and S. Krishnamurthy, "Natural Language Processing," in *Building an Effective Data Science Practice*, Berkeley, CA: Apress, 2022, pp. 63–73.

[37] K. M. Kahloot and P. Ekler, "Algorithmic splitting: A method for dataset preparation," *IEEE Access*, 2021, vol. 9, pp. 125229–125237.

[38] N. M. Kebonye, "Exploring the novel support points-based split method on a soil dataset," *Measurement (Lond.)*, 2021, vol. 186, no. 110131, p. 110131.

[39] Read, J., Pfahringer, B., Holmes, G. and Frank, E., "Classifier chains for multi-label classification," *Machine learning*, 2011, 85, pp.333-359.

[40] Kavitha M, Prabhavathy P, "A Hybrid Approach for Text Classification," *Journal of Theoretical and Applied Information Technology*. 2023 Apr 30;101(8).

[41] Kumar, S., Kumar, N., Dev, A. and Naorem, S., "Movie genre classification using binary relevance, label powerset, and machine learning classifiers," *Multimedia Tools and Applications*, 2023, 82(1), pp.945-968.

[42] Amr Elsayed , "SVM transformations for Multi-labeled Topics," *TechRxiv*. 2022.

[43] Khan, T.T., Hassan, A., Ahamed, MF and Islam, S., "Multi-label Bengali Abusive Comments Classification using Problem Transformation Method," In *2023 20th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, IEEE, pp.1-6.

[44] Lerian, J.C. and Chenayan, G., "The Implementation of Multi Label K-Nearest Neighbor Algorithm to Classifying Essay Answers," *Journal of Information System, Technology and Engineering*, 2023, 1(3), pp.89-94.

[45] Alzanin, S.M., Gumaei, A., Haque, M.A. and Muaad, A.Y., "An optimised Arabic multi-label text classification approach using genetic algorithm and ensemble learning," *Applied Sciences*, 2023, 13(18), p.10264.