Journal of Informatics and Web Engineering
Vol. 4 No. 2 (June 2025) eISSN: 2821-370X

Dynamic Job Recommendation by Profiling Undergraduates Academic Performances

Bao-Ling Foo[1], Choo-YeeTing[2*], Hui-Ngo Goh[3], Albert Quek[4], Chin-Leei Cham[5]
[1,2,3,4] Faculty of Computing and Informatics, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Malaysia
[5] Faculty of Artificial Intelligence and Engineering, Multimedia University, Jalan Multimedia, 63100 Cyberjaya, Malaysia
**corresponding author: (cyting@mmu.edu.my; OCRCiD: 0000-0001-5667-2816)*

Abstract - Job-seeking tasks are always challenging. Often, job recommendation systems require human intervention in the job-seeking process. Therefore, the study focuses on recommendation of most relevant job sectors and prioritizing companies based on a student's profile. The objectives of this study are: (i) to identify important features that optimize job recommendation, (ii) to construct a predictive model that recommends most relevant job sectors, and (iii) to recommend companies by computing the similarity between student and job profiles. In this study, the dataset was collected from Graduate Tracer Study from a university. Additionally, a job dataset was collected to extend the training dataset. As a result, both students and job profiles are used in this study. To enhance the accuracy, several models have been utilized for classifying job sector. This includes both hierarchical and single level classification. In hierarchical classification, Random Forest and Categorical Boosting were utilized; while in single level classification, a total of 9 different machine learning models were utilized. To assess the model's performance, the metrics such as accuracy, weighted precision, weighted recall, and weighted f1-socre, were utilized. The finding shows that Hierarchical Classification outperforms Single Level Classification, with evaluation metrics ranging from approximately 72% to 76%, whereas Single Level Classification achieved around 58% to 62%. In conclusion, the integration of BorutaShap with Bidirectional Encoder Representation Transformers with 12 transformed layers enhances the performance of Hierarchical Classification, with the highest evaluation metrics around 75%. To recommend companies, a predefined rule is utilized to filter relevant companies, then, the similarity of the companies is measured using Cosine Similarity after transforming both student and company information using Bidirectional Encoder Representation Transformers with 12 transformed layers.

Keywords— Job Recommendation, Company Recommendation, Transformers, Classification, Random Forest, Categorical Boosting, Hybrid Filtering

Received: 16 October 2024; Accepted: 26 March 2025; Published: 16 June 2025

This is an open access article under the CC BY-NC-ND 4.0 license.



Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2025.4.2.12
© Universiti Telekom Sdn Bhd.
Published by MMU Press. URL: https://journals.mmupress.com/jiwe

1. INTRODUCTION

Recommendation System (RS) consists of software tools to provide recommendation for users [1], [2], [3]. In the context of RS, the application is commonly categorized into eight key fields, such as e-business, e-commerce, e-government, e-group, e-learning, e-library, e-resource service, and e-tourism [4], [5], [6]. Among these, e-business category is commonly integrated with e-recruitment platforms, which often provide recommendations of the job to users. Therefore, Job Recommendation System (JRS) plays a key role in making job searches easier for students and employers. However, challenges arise in matching student and company profiles, resulting in time-consuming in manual searches for a job and lack of industry-specific insights. Additionally, limited work experiences of fresh graduates also often as the challenges in existing systems. Furthermore, JRS often lacks sufficient customization, placing excessive emphasis on formal qualifications, and limiting their ability to adapt to the evolving job market. Therefore, this study focused on enhancing job recommendations by identifying the most relevant job sectors and prioritizing companies based on student profiles. Therefore, the main objectives are: (i) to identify the important features that optimize job recommendation, (ii) to construct a predictive model that recommends most relevant job sectors, and (iii) to recommend companies by computing the similarity between student and job profile. In this study, the structure of the report is as follows: Part 2 explores related work on JRS, while Part 3 explains the methodology for the proposed solution. Next, Part 4 presents the result and discussion. Finally, Part 5 concludes the results and states the possible future improvements.

2. RELATED WORKS

2.1 Job Recommendation System

In the field of JRS, researchers employed different methodologies to enhance system performance. Various JRS primarily focused on the variables within user (student) and job profiles. In addition, recommendation techniques are often employed by researchers to improve the accuracy of recommendation. Furthermore, AI-based techniques, such as machine and deep learning, are often used to refine the matching process. However, there are challenges associated with binary classification in AI-based techniques. Therefore, to address these limitations, multi-class classification is utilized, which allows data to be categorized into multiple groups rather than just 2, as seen in binary classification. As a result, a detailed discussion of these techniques will be provided in the next section.

2.2 Variables in Job Recommendation System

In several studies of JRS, researchers focused on different aspects of student profiles, including demographic, education, experiences, skills, interests, expectations, and interactions with JRS, as shown in Table 1. Several studies have highlighted different variables within student profiles to enhance job matching accuracy. For instance, the researcher [7] explored a wide range of demographic variables, including address, age, gender, and marital status, alongside personality traits, education, work experience, skills, and expectations. Similarly, the researcher [8] placed emphasis on demographic details such as gender and address while also considering educational background, skills, and expectations. In contrast, the researcher [9] focused on identifying skills required for the job roles. Furthermore, the researcher [10] explored both demographic and educational aspects, analysing email addresses, phone numbers, universities attended, qualification levels, skills, and expectations. This ongoing research trend underscores the multifaceted nature of student profiles in job recommendations, with each study offering different insights into the crucial factors influencing job matching effectiveness.

However, student profiles alone do not provide sufficient information. To enhance the effectiveness of JRS, it is equally important to include job profiles, which provide insights into job roles and company details. As a result, job profile variables are analysed as part of the related work. Existing research highlights that job profiles consist of various elements, including both job and company information, as shown in Table 2. Various studies have explored different aspects to job profiles to refine recommendation accuracy. For example, the researchers [7] focused on job offers, categories, working hours, and skill/experience requirements, while another researcher [9] analysed job titles, positions, descriptions, responsibilities, and requirements.

Table 1. Variables in Student Profiles (Summary)

Category	Variables
Demographic Information	Age [7], [8], [11], [12], Marital Status [7], Address [7], [11-15], Full Name [15], Gender [7], [8], [11], [12], [15], [16], Personality [7], [17-20], Community Convenience [21], Hobbies [22], Health [21], Personal Improvement [21], Family Economic Condition [12]
Educational Background [7]	University [11], [15], Grades [11], [12], [16], Qualification Level [8], [10], [11], [22], [23], Passing Year [11], Course [20], [24], Course ID [24], Course Description [24], Number of Failed Courses [12], Field of Study [8], [13], [14], Main Subject [17], [22], Major Subject [11], [12], [14], [15], Academic Performance [20], [25], Extracurricular Activities [18]
Work Experiences	Experience [7], [11], [21], [23], [26], [27], Position [11], Salary [11], Project [27], Internship [27], Work Duration [22]
Skills [7], [8], [9], [10], [11], [13], [23], [25], [27], [28]	Technical [22], [24], [29], Non-Technical [22], [24], Language [14], [22], [23], [29], Ability [30], Style [30], Knowledge [30], Tools [30], [31], Desired Skills [23], Certificate Skills [26], Technology [30], Artistic Skills [18], Computer-Related Skills [18], Sportive Skills [18], Community Services [18], Volunteering [18], Other [31]
Interest Areas [16], [19], [25]	Personal Interest [38], Research Direction [15], Job Intention, Area of Expertise [13], Domain Interest [13]
Expectations [19]	Job Type [12], Salary [15], [23], Location [12], [15], [23], Job Position [8], [9], [15], Working Environment [15], [21], Area with High Economic [12], Area with High Familiarity [12], Job Score [12], Working Intensity [21], Working Hours [21], Schedule Flexibility [21]
Interactions [16]	Apply [11], Like [10], [11], View [11], [32], Click [11], Revisited [11], Read [11], Email [11], Login [11], Register [11], Rating [10], [32], Reviews [10], Search [32], Behavior, Feedback

A more extensive perspective was adopted by the researcher [10], which investigated job titles, descriptions, positions, salaries, vacancies, and ratings. Meanwhile, the researchers [11] focused on job descriptions, required skills, and ratings. Additional investigations conducted by the researchers [13], [19], [24], [29], [31] delved into job fields, titles, descriptions, and associated details. Subsequent research, including studies by the researchers [14], [15], [18], [30], [34], consistently highlighted job titles, descriptions, and related profiles. Meanwhile, researchers [16], [21], [22], [25], [26], [28], [33] analysed job titles, descriptions, positions, skills, qualifications, ratings, and requirements. By combining student and job profiles, these studies present a comprehensive view of the factors influencing job recommendation. As a result, this multidimensional approach strengthens JRS functionality, improving both the accuracy and relevance of job matches. For further details on the key variables influencing job matching, Tables 1 and 2 provide a structured overview.

2.3 Recommendation Techniques

Table 3 provides a summary of the recommendation techniques used in existing work. The recommendation techniques, such as Content Based Filtering (CBF), Collaborative Filtering (CF), Semantic Filtering (SF), Rule Based Filtering (RBF), and Hybrid Filtering (HF), are commonly employed to narrow down large sets of items into smaller based on specific criteria.

Table 2. Variables in Job Profiles (Summary)

Category	Variables
Job Information [20]	Job Title [8], [10], [14], [16], [17], [18], [24], [25], [27], [31], [32], Job Description [10], [11], [22], [25], [26], [34], Job Category [7], [22], Job Vacancies [8], [10], [27], Job Position [8], [10], [12], [14], [22], [28], [32], [33], Job Responsibility [15], [28]
Career Details	Job Field [13], [14], [31], Job Sector [14], [18], [29], Job Type [12], [27]
Posting Details	Tags [10], Post Time [10]
Work Conditions	Working Hours [7], [21], Flexibility [21], Environment [15], [21], Intensity [21], Salary [8], [10], [21], [23], [25]
Employee Benefits	Welfare
Company Profiles	Company Name [8], [10], [12], [15], [16], [24], [27], Company Address [7], [8], [10], [12], [13], [15], [23], [24], [25], [27]
Company Requirement [16]	Skills [7], [8], [10], [11], [13], [14], [15], [22-25], [27], [28], Experiences [7], [21], [22], [23], [26], [27], Qualification Level [10], [15], [21], [22], [23], [26], Language [15], [23], [26], Major [15], Health [21], Commuting Convenience [21]
Company Rating	Ratings [10], [11], [13], [27], [33], Likes [10], Reviews [10]

Table 3. Recommendation Techniques Used by Researchers

Author	CBF	CF	SF	RBF	HF
[1]	✓				✓
[7], [12], [16], [31], [33], [35]		✓			
[8], [19], [34], [36]	✓				
[9]	✓	✓		✓	✓
[11], [13], [25], [27]	✓	✓			✓
[14]	✓	✓	✓		✓

First, CBF recommends the items to users by computing the similarity of the items features with user's preferences [1], [8], [9], [11], [13], [14], [19], [25], [27], [34], [36]. This similarity is typically computed using vector-based metrics such as Cosine Similarity, Pearson Correlation Coefficient, Jaccard Coefficient, and Euclidean Distance. Second, CF recommends items based on the preferences or behaviours of similar users [7], [9], [11-14], [16], [25], [27], [31], [33], [35]. The similarity metrics are identical to CBF, but in CF, the similarity metrics is used to identify the user or item relationship. Third, SF focused on understanding the meaning of items to enhance the accuracy of the recommendation. Unlike simple keyword matching, SF often employed advanced transformer techniques to understand the context and meaning of items and user preferences [14]. Furthermore, RBF relies on predefined rules or conditions to filter out material, thereby making recommendations [9].

Lastly, HF is often used to improve the recommendation systems by combining multiple techniques, effectively addressing the limitations of individual techniques [1], [9], [14], [25]. The researcher [14] combined CBF, CF, and SF, utilizing each for specific variables like skills, job sectors, and related skills. Additionally, researchers [25], [27] employed hybrid approaches by combining different recommendation techniques like CBF and CF. The concept of hybridization in recommendation systems involves the use of various techniques to successfully integrate different recommendation methods. There are several hybridization techniques such as weighted, switching, mixed, feature combination, cascade, feature augmentation, and meta-level.

a. **Weighted hybridization** assigns weights to recommend from different methods, either statically or dynamically adjusting based on the performances.
b. **Switching hybridization** dynamically selects the best-performing techniques based on specific conditions or user contexts.
c. **Mixed hybridization** combines recommendations from different techniques, providing users with a mix of output without explicitly considering the strengths and weaknesses of the individual techniques.

d. **Feature combination** integrates the attributes from several recommendation techniques into a single model.
e. **Cascade hybridization** uses the output of one recommendation technique as the input for another.
f. **Feature augmentation** integrates additional user demographics or external data.
g. **Meta-level hybridization** introduces a meta-learning layer that pre-select or combines different recommendation techniques based on user characteristics. This hybridization potentially employs combination techniques using Machine Learning techniques.

2.4 Artificial Intelligence (AI) Techniques

AI has revolutionized multiple fields by allowing systems to simulate human intelligence, particularly in decision-making, pattern recognition, and data processing. Within AI, Machine Learning (ML) plays a crucial role in improving the accuracy of recommendation systems. As shown in Table 4, several ML techniques are often employed by research such as Natural Language Processing (NLP), Naive Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Decision Tree (DT), Logistic Regression (LR). In the context of recommendation system, NLP is used to understand, interpret, and generate human language [7], [10], [22], [34]; whereas other ML techniques are often used for classification tasks. RF is an ensemble learning method that builds multiple decision trees on random data subsets [7], [32], [34]. Additionally, XGBoost is a gradient-boosting algorithm that sequentially corrects errors from previous trees [12], [16], [26]. Lastly, NN or Multi-Layer Perceptron (MLP), is a model that with multiple layers of neurons that learn complex patterns [1], [2], [7], [8], [11], [26], [30].

Table 4. Machine Learning Techniques Used by Researchers

Author	NLP	NB	SVM	KNN	RF	XGB	DT	LR	NN
[1, 3]									✓
[7]	✓	✓	✓	✓	✓		✓		✓
[8]		✓					✓		✓
[10], [22]	✓								
[11]		✓							✓
[12], [14]						✓			
[26]						✓			✓
[30]				✓					✓
[31]				✓					
[32]	✓	✓	✓		✓				
[34]	✓	✓	✓	✓	✓		✓		
[35]			✓					✓	

Deep Learning (DL), a subset of ML, takes recommendation systems to a new level by enabling more complex feature extraction and sequential pattern recognition. As shown in Table 5, Hybrid Convolutional Neural Networks (HCNN), Deep Semantic Similarity Models (DSSM), Autoencoder (AE), Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Deep Neural Network (DNN), are commonly employed by researchers. HCNN combines feature extraction techniques with Convolutional Neural Networks (CNN), to enhance the robustness for the application of the job recommendation [16]; on the other hand, DSSM computing the semantic similarity of user and item descriptions [26]. Additionally, AE enables dimensionality reduction and feature learning, which discovers patterns in user-item interactions [26]. Furthermore, sequential models like RNN and LSTM networks excel in modelling temporal dynamics, which enable the system to track user behaviours over time [22], [33]. Lastly, DNN learn complex relationships in recommendation systems, which provide an advanced framework [1], [2].

Table 5. Deep Learning Techniques Used by Researchers

Author	HCNN	DSSM	AE	LSTM	RNN	DNN
[1], [2]						✓
[16]	✓					
[22], [33]				✓	✓	
[26]		✓	✓			
[37], [38], [39], [40]					✓	
[41-46]				✓		

2.5 Multi-Class Classification Techniques

In machine learning, multi-class classification is crucial for enabling models to differentiate between multiple categories. Table 6 presents various multi-class classification techniques, such as One-vs-One (OvO), One-vs-Rest (OvR), and One-vs-One-vs-Rest (OvOvR) [54], [55], [56], are commonly employed by researchers. OvO constructs binary classifiers for every pair of classes, while OvR trains separate classifiers for each class against all others. The hybrid OvOvR approach integrates both strategies, improving classification performance in complex scenarios. Researchers employ diverse methodologies within these frameworks, such as conventional techniques like SVM and DT, as well as advanced methods like CNN, ensemble learning, and multi-level classification. SVM has been widely utilized, with [50] enhancing multi-class document classification using non-linear kernels, and the researcher [56] exploring similar kernel-based approaches. Neural networks also play a significant role, as seen in the study [52], where a DNN is transformed using Generative Pre-Trained Transformer (GPT) or Bi-directional Encoder Representation Transformer (BERT) for improved classification accuracy. Additionally, boosting techniques such as CatBoost and Logit Boost have been applied to tackle class imbalances [58], while the researcher [57] adapts AdaBoost for CNN-based classification. Researchers have also leveraged hybrid approaches, with the researcher [49] combining KNN and SVM for enhanced performance.

Table 6. Multi Class Classification Techniques Used by Researchers

Author	OvOvR	OvO	OvR	ENSEMBLE	DL	OTHER
[47]					✓	✓
[48], [49], [50]						✓
[51]					✓	
[52], [53]					✓	
[54]		✓	✓		✓	✓
[55]	✓	✓	✓			
[56]			✓			
[57], [58]				✓		

Beyond traditional multi-class classification, researchers have explored hierarchical classification structures to refine classification accuracy. As illustrated in Figure 1, Multi-Class Single-Level Classification directly assigns each data point into a category, such as Healthy, Benign, Malignant, or Eczema. These straightforward techniques did not consider hierarchical relationships among classes. In contrast, Multi-Class Multi-Level Classification, shown in Figure 2, introduces a hierarchical structure where decisions are made in stages. For instance, an initial classification might distinguish between Healthy and Unhealthy conditions. If categorized as Unhealthy, a second level could further classify conditions into Melanoma or Eczema, followed by a third level differentiating between Malignant and Benign cases. This structured approach, explored by [47] in CNN-based image classification, is particularly useful in medical and complex domains where progressive classification leads to more precise and meaningful predictions. By

integrating machine learning and deep learning techniques, multi-class classification continues to evolve, offering more refined and adaptive solutions to diverse classification challenges.

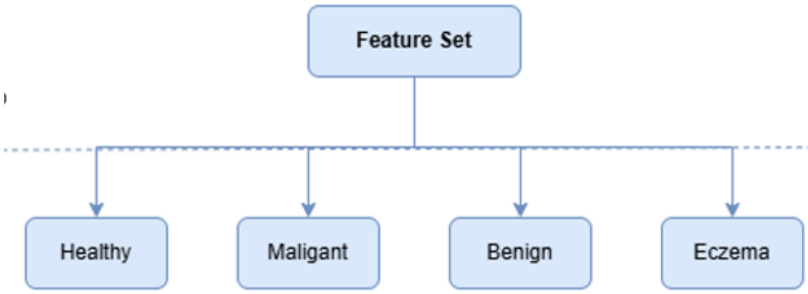


Figure 1. Multi-class Single Level Classification

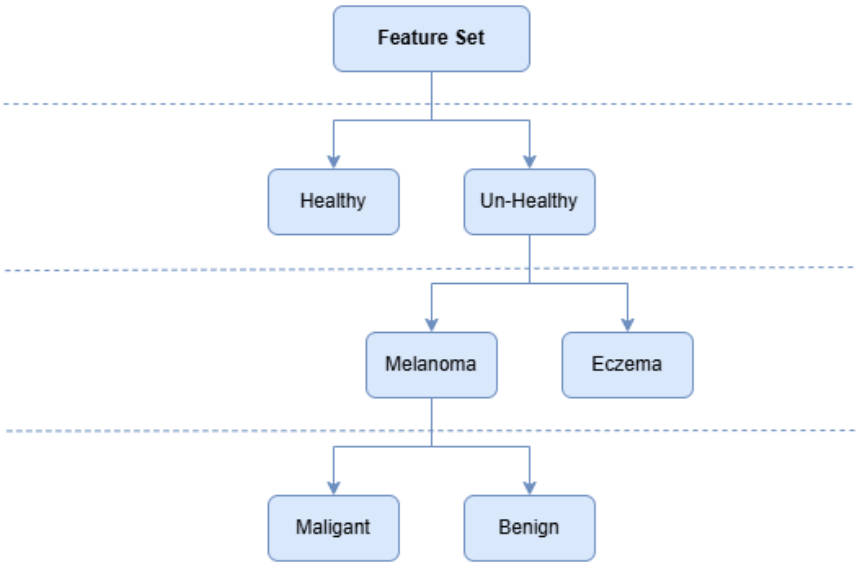


Figure 2. Multi-class Multi Level Classification (Hierarchical Classification)

3. METHOD

3.1 Introduction

Figure 3 illustrates the overall framework of the JRS, which involves several key stages, starting from data preparation, followed by feature selection, class imbalance treatment, model construction and optimization, and finally company filtering and ranking. Future enhancements to the company filtering techniques within the JRS are anticipated. The details of each stage will delve in the subsequent section.

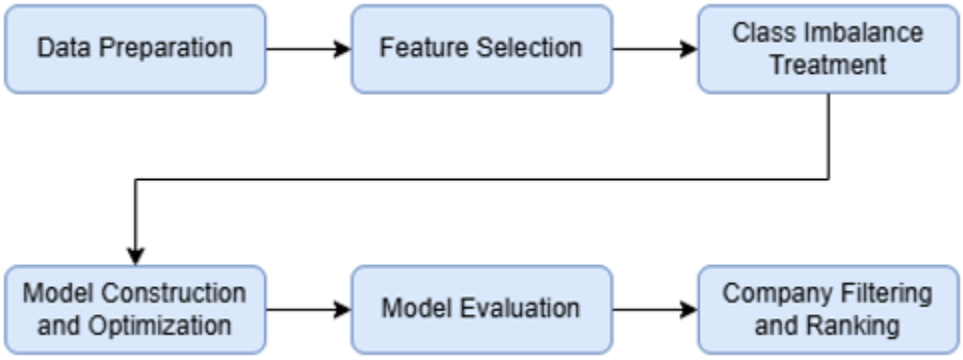


Figure 3. Overall Framework of JRS

3.2 Data Preparation

As shown in Figure 4, data preparation includes several key steps: data collection, data preprocessing, data splitting, and data transformation.

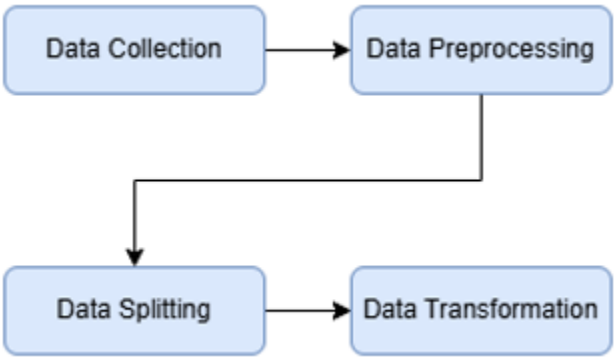


Figure 4. Pipelines in Data Preparation

3.2.1 Data Collection

The dataset was sourced from Graduate Tracer Study (GTS) from the university and job dataset, which includes information about both students and companies. Student profiles include demographics information, academic performance, education information, and work experiences, while company profiles include details like job listings, company information, requirements, and ratings.

The GTS, also known as "Sistem Kajian Pengesanan Graduan", is an annual study conducted by the Ministry of Higher Education (MOHE). This is to determine employment status after graduation. GTS dataset consists of two main files corresponding to the year 2021 and 2022. GTS dataset consists of 2374 records and 433 features in 2021; and 2096 records and 433 features in 2022. The GTS dataset is used to recommend the job sector.

On the other hand, the job dataset consists of three main files, including company information, requirement, and rating. The job database consists of 249,500 records and 15 features in the company information file; 66318 records and 168 features in the company requirement file; and 146925 records and 28 features in the company rating file. Due to the limited company information available in the GTS dataset, therefore, the job dataset is used to extend the functionality of recommending Top-N companies.

3.2.2 Data Preprocessing

In the data preprocessing process, it involves data integration, data cleaning, feature engineering, and finally missing value handling. In the data integration process, multiple files such as dataset from GTS in both 2021 and 2022 are integrated into a single file.

Next, data cleaning step includes the text data cleanup, filtering out unwanted data, standardizing values, and grouping similar values. Firstly, text data cleanup is to identify and eliminate irrelevant or noisy data. Secondly, filtering unwanted data such as removing rows of the overseas data such as overseas students and companies. This is to ensure that this study is within the scope of recommending the local or Malaysia companies based on local students. Thirdly, standardizing value to aligns there are correct and appropriate names to ensure consistency across the dataset. Finally, grouping similar values such as “Pulau Pinang” and “Penang” into “Penang” to simplify and streamline the data for improved analysis and interpretation.

In addition, feature engineering is crucial because it helps to improve model performance by transforming raw data into meaningful features. Therefore, in this study, new columns such as major, age, sum GPA, count of attempted exams, and average GPA, are created. This step is to enhance the dataset with relevant information for classification tasks.

Furthermore, missing value handling is a crucial step to ensure the accuracy and reliability of the mode. Therefore, this study addressed the missing values for both categorical and numerical variables. For categorical variables, “No_Data” is filled to represent the missing data, whereas, for numerical variables, “0” is filled to ensure consistency in calculations.

3.2.3 Data Splitting

In this study, the dataset is divided into train and test set using stratified sampling, with 80% allocated for train and 20% for test. Stratified sampling ensures that the class distribution in both train and test sets reflects the overall class distribution in the original dataset. By preserving the class distribution, the performance of the model can be more accurately assessed on unseen data.

3.2.4 Data Transformation

As shown in Figure 5, this study employed several techniques to transform the dataset. First, embedding is one of the techniques employed in this study. Embedding is a technique that is used to represent the text in numerical format while preserving its semantic meaning. As shown in Table 7, the dataset is given in a structured format, while embeddings are primarily used for unstructured data like text. Therefore, as shown in Figure 6, paragraph generation is utilized to assign the paragraph that consists of variables and its values. Once the paragraph is generated, embedding techniques are utilized to capture and represent semantic information using two different techniques: (i) BERT, and (ii) Universal Sentence Encoder (USE). In BERT embedding techniques, there are two key models: BERT-L6 and BERT-L12, corresponding to “all-MiniLM-L6-v2” and “all-MiniLM-L12-v2”, respectively. BERT-L6 consists of 6 transformed layers, therefore, it is known as BERT with 6 transformed layers. Similarly, BERT-L12 consists of 12 transformed layers, therefore, it is known as BERT with 12 transformed layers.

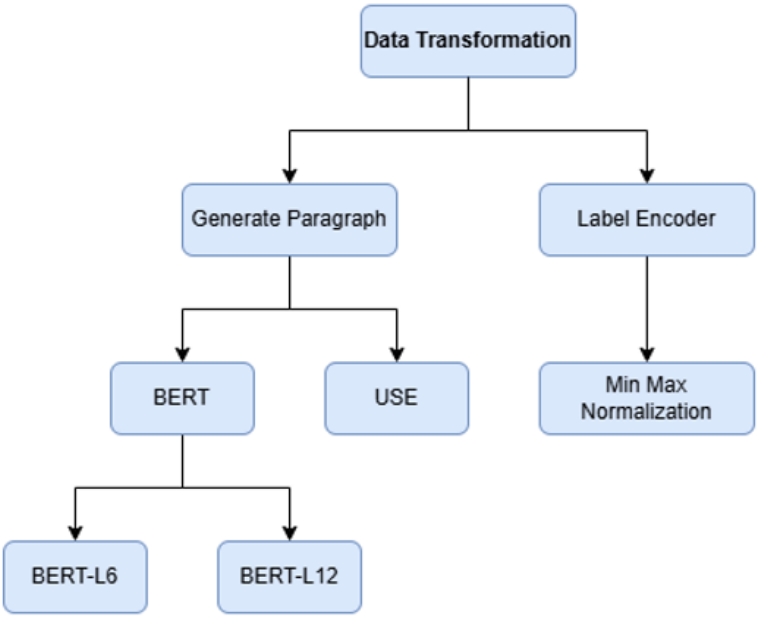


Figure 5. Data Transformation Step

Table 7. Structured Data

T1_GPA	T2_GPA	T3_GPA	...	T9_GPA
2.00-2.49	2.00-2.49	3.00-3.49	...	No_Data

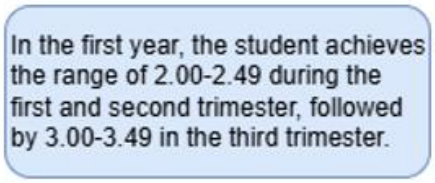


Figure 6. Generated Paragraph

3.3 Feature Selection

In this study, feature selection techniques are used to identify a subset of relevant features from a larger dataset to improve model performance and interpretability [61]. Two primary categories of feature selection techniques include wrapper methods and dimensionality reduction techniques [59]. Among wrapper methods, BorutaShap is employed to determine significant features in predictive modelling by iteratively evaluating feature importance and interactions. It classifies features into three categories: confirmed important, unimportant, and tentative features. In this study, only the confirmed important ones being selected for this study [60-64]. Additionally, dimensionality reduction techniques like Principal Component Analysis (PCA) help optimize the number of components while retaining at least 95% cumulative explained variance. This step is to ensure balance between reducing dataset dimensionality and preserving critical information [65-69]. Both encoding and embedding data employed the feature selection techniques to refine the set of relevant attributes, enhancing model efficiency and accuracy.

3.4 Class Imbalance Treatment

SMOTE, also known as Synthetic Minority Over-Sampling Technique, is one of the techniques that address the class imbalance issues by generating synthetic records for the minority class [70-75]. In this study, the treatment for class imbalance is applied for Single Level Classification tasks where there is a significant imbalance between the classes. However, for Hierarchical Classification tasks, the class distribution tends to become more balanced after breaking it down into multiple levels. Therefore, class imbalance treatment is not applied to Hierarchical Classification because the imbalance issue is mitigated through the hierarchical structure, which inherently balances the classes at each level.

3.5 Model Construction and Optimization

Figure 7 shows the construction of the model for the different combinations of transformation and feature selection techniques. The step uses two different types of classification: Single Level and Hierarchical Classification. Table 8 provides a summary of experiment setups for classification techniques. The Table lists 12 experiments that explore the effects of applying no feature selection, PCA or BorutaShap, paired with different transformation techniques like BERT-L6, BERT-L12, USE, and Label Encoder. Table 9 acts as a reference for ML models used throughout the study, with each model assigned a unique identifier for easy reference.

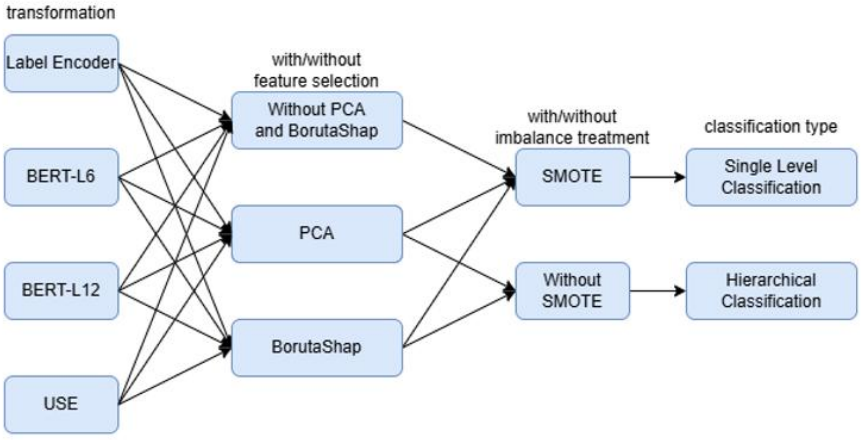


Figure 7. Model Construction (Different Experimental Cases)

Table 8. Overview of Experiment Setups for Classification Techniques

Experiment	Description
E1	Without Feature Selection + BERT-L6
E2	Without Feature Selection + BERT-L12
E3	Without Feature Selection + USE
E4	Without Feature Selection + Label Encoder
E5	PCA + BERT-L6
E6	PCA + BERT-L12
E7	PCA + USE
E8	PCA + Label Encoder
E9	BorutaShap + BERT-L6
E10	BorutaShap + BERT-L12
E11	BorutaShap + USE
E12	BorutaShap + Label Encoder

Table 9. Machine Learning Models Used in Classification Experiments

Model	Description
M1	K-Nearest Neighbors (KNN)
M2	Decision Tree (DT)
M3	Support Vector Machine (SVM)
M4	Logistic Regression (LR)
M5	Random Forest (RF)
M6	Adaptive Boosting (AdaBoost)
M7	Categorical Boosting (CatBoost)
M8	Extreme Gradient Boosting (XGB)
M9	Multi Layer Perceptron (MLP)

In Single Level Classification, a total of 12 experiments (E1-E12) are conducted to evaluate how different preprocessing and feature selection techniques impact model performances. Each model (M1-M9) is applied to every experimental setup to comprehensively understand the best practices in preprocessing for improving classification outcomes. Hierarchical Classification in this study is more focused, using only 8 experiments (E5-E12). These two models (M5&M7) are chosen for Hierarchical Classification due to their proficiency in managing complex data structures and their effectiveness in ensemble learning, which can significantly enhance performance in multi-level classification tasks.

Figure 8 illustrates a comprehensive tree diagram that visually represents the classification process across various levels. In the hierarchical classification context, classes are structured in a hierarchical manner, consisting primary classes (PC), secondary classes (SC), tertiary classes (TC), quaternary classes (4C), quinary classes (5C), and senary classes (6C). In this hierarchical classification approach, the organization of classes aims to balance their distribution, typically targeting an approximate 50% distribution. The classification involves utilizing a total of 9 different models to classify the output, with the final output classes or labels typically represented by the leaf nodes in the hierarchy.

To optimize the model performance, grid-based hyperparameter tuning, or GridSearchCV is utilized with 5-fold cross-validation. In this step, not all combinations of the method applied the optimization process. In single level classification, only the combination of Label Encoding and PCA (E4) are applied due to extended training time. In hierarchical classification, all the combinations of data transformation and feature selection techniques (E5-E12) are applied.

3.6 Model Evaluation

In this study, evaluation metrics such as accuracy, precision, recall, and f1-score, are used. When dealing with multi-class classification, weighted metrics for precision, recall, and f1-score are used. This is to address class imbalance by assigning the weight proportional with the frequency of the classes. Classes with fewer records achieved the higher weights, while classes with large records achieved the lower weights. In addition, for neural network classification, the loss function, Sparse Categorical Crossentropy, are employed to assess the model performance.

3.7 Company Filtering and Ranking

Several models are first used in the process to identify the job sector, then evaluation metrics are used to determine the best model. Following that, company filtering and ranking are used to prioritize companies using cascade hybridization, as shown in Figure 9.

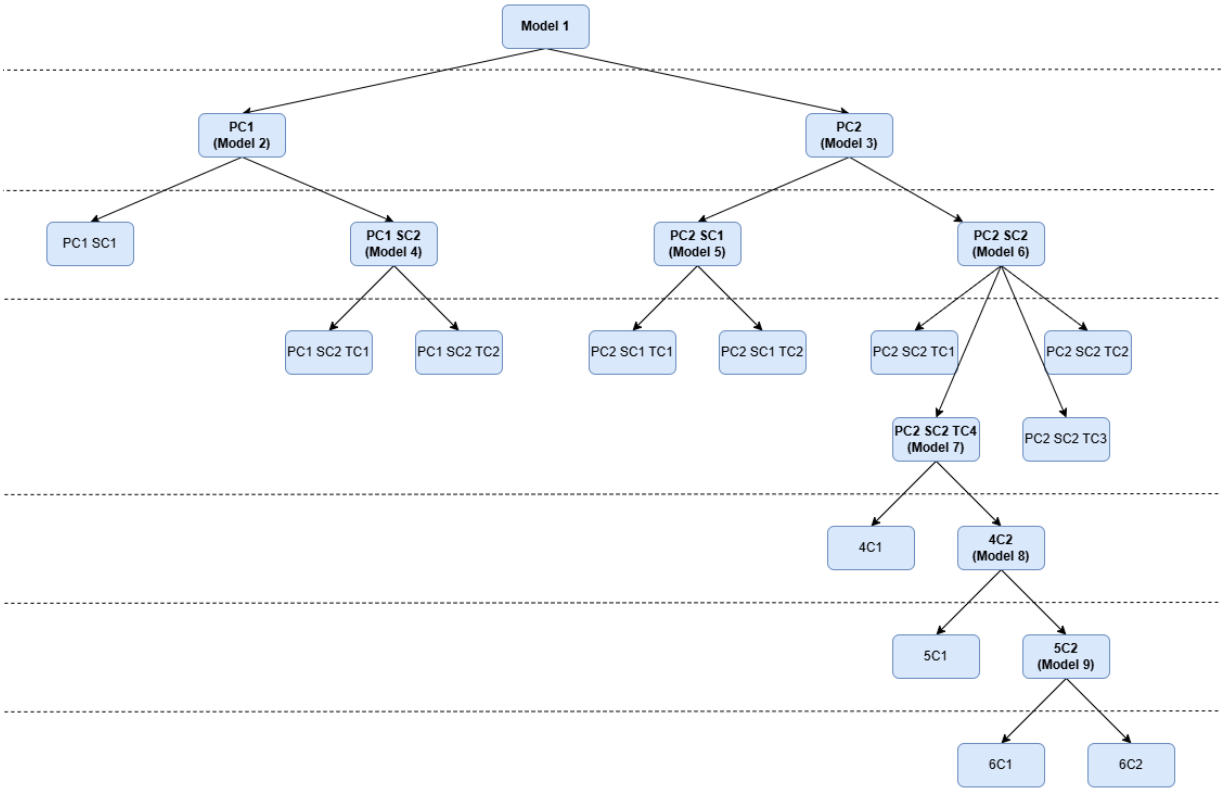


Figure 8. Hierarchical Classification

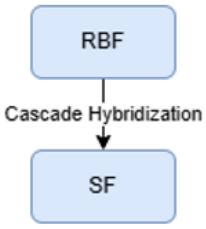


Figure 9. Company Filtering and Ranking to Recommend Companies

Firstly, RBF is used to filter based on pre-defined rules or conditions. The rule is defined as "If a student is classified into job sector X, recommend companies that offer job opportunities for job sector X". This rule explicitly links the student's job sector classification to the recommendation of relevant companies. The output of RBF is then used as input for the next step through cascade hybridization techniques. Following RBF, Semantic Filtering (SF) is applied to both student and company profiles. This means that the companies filtered by RBF become the input for SF, so that forming a cascaded workflow. SF using embedding technique (BERT-L12) to capture semantic relationships within textual data. The objective is to find the similarities between student profiles and company profiles based on the captured semantic information. Once both student and company profiles have been embedded, similarity calculations are performed using cosine similarity. The cosine similarity formula is expressed as Equation (1).

$$Co\sin e\ \ Similarity\ (A,B) = \frac{A \times B}{\|A\| \times \|B\|} \quad (1)$$

where:

- A and B represent the vectors of the student and company profiles respectively
- $\|A\|$ and $\|B\|$ denote their respective magnitude.

This calculation computes the similarity between student and company profiles. Based on the results of SF and similarity calculations, a list of recommended companies is generated. These recommendations take into account both the job sector classification and the semantic similarity between student and company profiles. By combining RBF with SF and similarity calculations, top companies are identified by matching both student and company profiles. Therefore, Top-k companies are recommended to the user or job-seeking students.

4. RESULTS AND DISCUSSIONS

The study investigated three key findings: feature selection using BorutaShap and dimensionality reduction via PCA, model performance in both single-level and hierarchical classification, and findings in company filtering and ranking. First, the findings on feature selection, such as where features selected by BorutaShap and the number of principal components retained after PCA, were discussed. Additionally, model performance for both single-level and hierarchical classification approaches, were discussed. Lastly, the analysis of company filtering and ranking such as the output of RBF, Semantic Similarity Computation, and the ranking of companies based on similarity scores, were discussed.

4.1 Feature Selection using BorutaShap and Dimensionality Reduction via PCA

The findings on PCA and BorutaShap highlight significant differences in the outcomes based on the methodology employed for data transformation. PCA determines the number of components needed to capture a specified amount of data variance, typically set at 95%. In the case of different transformation techniques, BERT-L6 retained 84 components, BERT-L12 retained 80 components, USE required 91 components, and Label Encoder retained 44 components. Meanwhile, BorutaShap identifies important features within the dataset. However, the number of attributes identified, and the specific features chosen by BorutaShap analysis vary significantly depending on the data transformation techniques. For instance, BERT-L6 and BERT-L12 each identified 156 attributes, USE selected 125 attributes, and Label Encoder identified 21 attributes, including variables such as current GPA, program description, faculty, and spent term. Moreover, the choice of data transformation technique influences the interpretability of the selected features. Sentence embeddings like BERT and USE provide numerical values but do not preserve the original feature names, making interpretation more challenging. Conversely, methods like Label Encoder maintain the original feature names, simplifying the interpretation of the selected features.

4.2 Model Performance

4.2.1 Single Level Classification

For single-level classification, model performance is evaluated based on different transformation techniques without feature selection, PCA, or BorutaShap. Additionally, the loss function utilized in the MLP is analysed. Table 10 presents the model performance without feature selection, where RF, AdaBoost, XGBoost, and MLP emerge as the top-performing models in terms of accuracy, precision, recall, and F1-score. Among the transformation techniques, BERT-L12 and Label Encoder consistently provide better performance across different models and evaluation metrics.

Table 10. Performance in Single Level Classification (E1-E4) & (M1-M9)

Experiment	Model	Accuracy	Precision	Recall	F1-Score
E1	M1	39.30%	48.18%	39.30%	42.24%
	M2	31.34%	35.42%	31.34%	33.02%
	M3	57.05%	54.67%	57.05%	54.95%
	M4	55.72%	56.51%	55.72%	55.54%

	M5	58.37%	52.62%	58.37%	54.64%
	M6	57.55%	51.91%	57.55%	53.95%
	M7	54.39%	50.92%	54.39%	52.15%
	M8	58.04%	53.54%	58.04%	55.21%
	M9	56.38%	57.07%	56.38%	55.73%
E2	M1	42.79%	50.40%	42.79%	45.54%
	M2	37.48%	42.45%	37.48%	39.65%
	M3	58.37%	57.38%	58.37%	56.89%
	M4	54.23%	55.99%	54.23%	54.41%
	M5	60.36%	53.86%	60.36%	56.38%
	M6	62.02%	57.96%	62.02%	58.62%
	M7	56.22%	52.88%	56.22%	54.13%
	M8	59.87%	54.73%	59.87%	56.66%
	M9	56.72%	56.42%	56.72%	55.29%
E3	M1	41.46%	48.20%	41.46%	43.78%
	M2	39.64%	43.14%	39.64%	41.17%
	M3	40.13%	52.59%	40.13%	43.05%
	M4	47.26%	48.83%	47.26%	47.68%
	M5	58.04%	52.38%	58.04%	54.69%
	M6	58.54%	51.81%	58.54%	54.70%
	M7	54.73%	51.63%	54.73%	52.91%
	M8	55.89%	50.65%	55.89%	52.78%
	M9	32.67%	53.88%	32.67%	38.07%
E4	M1	44.44%	46.78%	44.44%	45.35%
	M2	29.19%	37.28%	29.19%	31.93%
	M3	52.90%	51.41%	52.90%	51.69%
	M4	46.60%	51.98%	46.60%	48.30%
	M5	61.53%	57.59%	61.53%	58.10%
	M6	58.21%	52.94%	58.21%	54.87%
	M7	58.37%	54.24%	58.37%	55.96%
	M8	62.35%	57.18%	62.35%	59.00%
	M9	41.29%	54.17%	41.29%	44.73%

When PCA was incorporated, as shown in Table 11, SVM, LR, RF, and XGBoost demonstrated the best performance in accuracy, precision, recall, and F1-score. Similarly, BERT-L12 remained the most effective transformation technique, providing consistently better results across all models and evaluation criteria.

Table 11. Performance in Single Level Classification (E5-E8) & (M1-M9)

Experiment	Model	Accuracy	Precision	Recall	F1-Score
E5	M1	38.14%	47.37%	38.14%	41.19%

	M2	28.19%	32.89%	28.19%	30.18%
	M3	54.23%	54.81%	54.23%	53.61%
	M4	51.24%	55.34%	51.24%	52.51%
	M5	57.88%	52.91%	57.88%	54.50%
	M6	30.02%	36.47%	30.02%	32.47%
	M7	56.55%	52.89%	56.55%	54.39%
	M8	57.55%	52.73%	57.55%	54.52%
	M9	51.24%	55.49%	51.24%	52.57%
E6	M1	42.79%	50.61%	42.79%	45.84%
	M2	35.32%	39.44%	35.32%	36.93%
	M3	56.38%	56.58%	56.38%	55.42%
	M4	51.41%	55.94%	51.41%	52.61%
	M5	61.86%	56.16%	61.86%	57.59%
	M6	60.86%	55.29%	60.86%	56.66%
	M7	56.38%	52.47%	56.38%	53.98%
	M8	57.71%	52.92%	57.71%	54.72%
	M9	46.60%	54.60%	46.60%	49.21%
E7	M1	32.84%	44.37%	32.84%	36.32%
	M2	30.18%	34.41%	30.18%	31.85%
	M3	36.48%	53.09%	36.48%	39.78%
	M4	17.25%	41.12%	17.25%	21.54%
	M5	49.59%	45.42%	49.59%	47.07%
	M6	38.81%	38.98%	38.81%	38.61%
	M7	45.94%	44.81%	45.94%	45.10%
	M8	50.25%	48.66%	50.25%	49.08%
	M9	17.91%	36.71%	17.91%	19.96%
E8	M1	33.67%	41.78%	33.67%	36.44%
	M2	25.21%	29.25%	25.21%	26.59%
	M3	45.77%	44.97%	45.77%	45.12%
	M4	38.14%	51.48%	38.14%	42.34%
	M5	51.24%	46.70%	51.24%	48.32%
	M6	50.58%	46.86%	50.58%	47.72%
	M7	47.76%	46.08%	47.76%	46.75%
	M8	48.42%	46.57%	48.42%	47.32%
	M9	41.79%	49.82%	41.79%	44.64%

Furthermore, Table 12 presents the results when BorutaShap was applied. In this case, SVM, LR, RF, and AdaBoost emerged as the best-performing models, while BERT-L12 continued to outperform other transformation techniques, reinforcing its effectiveness in enhancing classification performance.

Table 12. Performance in Single Level Classification (E9-E12) & (M1-M9)

Experiment	Model	Accuracy	Precision	Recall	F1-Score
E9	M1	38.47%	48.49%	38.47%	41.64%
	M2	27.86%	32.59%	27.86%	29.76%
	M3	54.23%	55.09%	54.23%	53.63%
	M4	54.06%	56.91%	54.06%	54.60%
	M5	59.04%	54.21%	59.04%	55.58%
	M6	58.21%	52.52%	58.21%	54.37%
	M7	55.06%	52.50%	55.06%	53.27%
	M8	57.88%	53.50%	57.88%	54.95%
	M9	48.76%	53.89%	48.76%	49.18%
E10	M1	39.64%	48.65%	39.64%	42.88%
	M2	37.65%	42.48%	37.65%	39.55%
	M3	57.71%	57.64%	57.71%	56.42%
	M4	54.23%	57.14%	54.23%	54.59%
	M5	61.86%	56.32%	61.86%	57.89%
	M6	61.86%	56.72%	61.86%	58.14%
	M7	55.56%	52.49%	55.56%	53.69%
	M8	58.87%	53.91%	58.87%	55.76%
	M9	50.58%	57.51%	50.58%	52.70%
E11	M1	40.96%	48.88%	40.96%	43.67%
	M2	37.48%	42.42%	37.48%	39.53%
	M3	46.27%	47.55%	46.27%	42.38%
	M4	44.61%	50.18%	44.61%	46.44%
	M5	58.37%	53.53%	58.37%	55.14%
	M6	57.38%	51.71%	57.38%	53.90%
	M7	52.57%	50.07%	52.57%	50.86%
	M8	55.22%	51.31%	55.22%	52.85%
	M9	46.60%	50.11%	46.60%	47.38%
E12	M1	49.25%	48.89%	49.25%	48.92%
	M2	35.82%	43.16%	35.82%	38.82%
	M3	49.59%	52.25%	49.59%	49.51%
	M4	47.60%	54.00%	47.60%	48.67%
	M5	52.07%	52.64%	52.07%	51.88%
	M6	54.89%	51.15%	54.89%	52.71%
	M7	53.23%	51.54%	53.23%	51.95%
	M8	47.76%	48.10%	47.76%	47.29%
	M9	44.11%	53.74%	44.11%	45.80%

Beyond accuracy metrics, the loss function in MLP was also analysed. As indicated in Table 13, the loss function varied across different data transformation techniques and feature selection methods. Notably, BERT-based transformation techniques resulted in lower loss values, whereas USE produced higher loss values, indicating relatively weaker performance. Among feature selection methods, BorutaShap proved to be the most effective, consistently reducing loss compared to PCA.

Table 13. Loss Function in Model (M9) - Experiment (E1-E12)

Experiment	Loss
E1	1.68
E2	1.83
E3	1.68
E4	1.69
E5	1.98
E6	1.7
E7	1.97
E8	2.5
E9	1.98
E10	1.93
E11	1.88
E12	1.75

Overall, in single-level classification, RF emerged as the top-performing model across all evaluation metrics, including accuracy, precision, recall, and F1-score. Additionally, BorutaShap stood out as the best feature selection technique, consistently improving model performance. Meanwhile, BERT-L12 was the most effective transformation technique, offering richer and more informative representations that contributed to enhanced classification outcomes.

4.2.2 Hierarchical Classification

For hierarchical classification, an in-depth analysis was conducted using RF and CatBoost, incorporating various feature selection techniques such as PCA and BorutaShap, along with different transformation techniques including BERT-L6, BERT-L12, USE, and Label Encoder. Table 14 compares the performance of RF and CatBoost models across different feature selection and transformation techniques. The analysis reveals that RF models achieved the highest accuracy and recall (both at 75.97%) when using Label Encoder. However, in terms of precision (74.38%) and F1-score (74.44%), BERT-L12 transformation performs slightly better. On the other hand, CatBoost models with BERT-L12 transformation and PCA show the highest accuracy (75.48%), precision (75.02%), and recall (75.48%), while the combination of CatBoost and BorutaShap yields the highest F1-score (73.81%).

Table 14. Overall Performance in Hierarchical Classification (Average) - Experiment (E5-E12) & Model (M5&M7)

Model	Experiment	Accuracy	Precision	Recall	F1-score
M5	E5	74.04%	71.67%	74.04%	71.92%
	E6	73.99%	71.97%	73.99%	72.07%
	E7	67.87%	64.00%	67.87%	65.33%
	E8	73.14%	71.42%	73.14%	71.09%
	E9	74.37%	72.49%	74.37%	72.78%

	E10	75.59%	74.38%	75.59%	74.44%
	E11	72.93%	71.73%	72.93%	71.68%
	E12	75.97%	74.21%	75.97%	74.24%
M7	E5	73.94%	72.00%	73.94%	72.40%
	E6	75.48%	75.02%	75.48%	73.08%
	E7	67.87%	64.00%	67.87%	65.33%
	E8	71.34%	69.12%	71.34%	69.18%
	E9	73.56%	71.42%	73.56%	71.80%
	E10	75.11%	73.78%	75.11%	73.81%
	E11	72.89%	68.72%	72.89%	69.86%
	E12	75.14%	72.91%	75.14%	72.64%

To further explore model performance, Table 15 presents a comparison between RF and CatBoost, considering all feature selection and transformation techniques. The results indicate that RF outperforms CatBoost across all evaluation metrics, achieved the highest accuracy (73.49%), precision (71.48%), recall (73.49%), and F1-score (71.69%). Next, Table 16 examines the impact of feature selection techniques, showing that BorutaShap consistently outperforms PCA with accuracy (74.45%), precision (72.45%), recall (74.45%), and F1-score (72.66%). Similarly, Table 17 evaluates transformation techniques, revealing that BERT-L12 achieved the best performance across all metrics, with accuracy (75.04%), precision (73.79%), recall (75.04%), and F1-score (73.35%). Conversely, USE exhibits the lowest performance, with accuracy (70.39%), precision (67.11%), recall (70.39%), and F1-score (68.05%).

Table 15. Performance – Model (Average)

Model	Accuracy	Precision	Recall	F1-score
RF	73.49%	71.48%	73.49%	71.69%
CatBoost	73.17%	70.87%	73.17%	71.01%

Table 16. Performance – Feature Selection Techniques (Average)

Feature Selection	Accuracy	Precision	Recall	F1-score
PCA	72.21%	69.90%	72.21%	70.05%
BorutaShap	74.45%	72.45%	74.45%	72.66%

Table 17. Performance – Transformation Techniques (Average)

Transformation	Accuracy	Precision	Recall	F1-score
BERT-L6	73.98%	71.89%	73.98%	72.23%
BERT-L12	75.04%	73.79%	75.04%	73.35%
USE	70.39%	67.11%	70.39%	68.05%
Label Encoder	73.90%	71.91%	73.90%	71.79%

Overall, the findings highlight that RF emerges as the top-performing model, consistently outperforming CatBoost. Among feature selection techniques, BorutaShap proves to be the most effective, while BERT-L12 stands out as the best transformation technique. The combination of RF with BorutaShap feature selection and BERT-L12 transformation yields the highest overall performance, achieved an accuracy of 75.59%, precision of 74.38%, recall

of 75.59%, and F1-score of 74.44%. Therefore, using this combination for hierarchical classification tasks ensures optimal performance across all evaluation metrics.

4.2.3 Comparison of Single Level Classification and Hierarchical Classification

Hierarchical classification demonstrated superior performance over single-level classification across accuracy, precision, recall, and F1-score, with the highest evaluation metrics ranging from approximately 72% to 76%, compared to about 58% to 62% for single-level classification. Additionally, in both classification approaches, models incorporating RF, BorutaShap for feature selection, and BERT-L12 transformation consistently achieved the best results.

4.3 Company Filtering and Ranking

For the result of company filtering and ranking, this involves filtering companies based on job sectors and then ranking them based on similarity between student and company profiles. In RBF, companies are categorized by job sectors, then ranked using cosine similarity. Next, top companies are identified based on specific student profiles. The filtering is performed using the predicted "Job_Sector".

Table 18 shows the findings of RBF, detailing the total number of companies filtered by each job sector and their respective company ID. Once the RBF is applied, the filtered companies are ranked based on cosine similarity between the student profiles and the company profiles. In an experiment targeting a student with the ID "U2022_1883" and a predicted job sector of "Computer/Information Technology", the Top-5 recommended companies are as follows: "Hostel Hunting Sdn Bhd", "Signature", "S Ecosystems", "Es Connect Sdn Bhd", and "Majikan". As shown in Table 19, these companies were ranking in descending order based on the similarity scores (highest similarity to lowest similarity).

Table 18. Findings in Rule Based Filtering

Job Sector	Number of Companies	Company IDs
Computer/Information Technology	559	1180, 2115, 1210, etc.
Accounting/Finance	380	2341, 1224, 697, etc.
Services	315	1202, 1521, 967, etc.
Sales/Marketing	285	952, 13, 1013, etc.
Admin/Human Resources	281	95, 2130, 2069, etc.
Engineering	157	571, 583, 470, etc.
Arts/Media/Communications	135	1631, 2274, 1956, etc.
Education/Training	117	2443, 215, 423, etc.
Others	60	351, 1620, 497, etc.
Manufacturing	44	2371, 628, 2260, etc.
Hotel/Restaurant	34	1840, 461, 2437, etc.
Building/Construction	13	2173, 1539, 457, etc.

Table 19. The Top-5 Recommended Companies

Company ID	Company Name	Similarity Score
1902	Hostel Hunting Sdn Bhd	0.6863
2383	Signature	0.6722
1418	S Ecosystems	0.6718

2325	Es Connect Sdn Bhd	0.6712
809	Majikan	0.6675

5. CONCLUSION

In conclusion, this study demonstrated the significant impact of data transformation techniques on identifying essential features for job recommendations. The PCA results indicate that sentence embeddings such as BERT and USE require more components to capture data variance compared to the Label Encoder. Meanwhile, BorutaShap analysis shows that although BERT and USE identify a larger number of features, these features are more difficult to interpret due to the loss of original feature names. In contrast, the Label Encoder selects fewer but more interpretable features, including variables like current GPA, program description, faculty, and spent term. Furthermore, the study aimed to enhance job recommendation by identifying key features and recommending the most suitable job sectors. By applying BorutaShap, the most relevant features for effective job recommendations were determined. Moreover, the integration of RF in Hierarchical Classification, combined with BorutaShap and BERT-L12, led to superior performance in predicting the most relevant job sectors, achieved an accuracy of 75.59%, precision of 74.38%, recall of 75.59%, and an F1-score of 74.44%. Finally, the company selection and ranking process leveraged RBF to filter relevant companies based on predefined criteria, followed by cosine similarity to compute semantic similarity scores. This ranking method allowed for the prioritization of the most compatible job opportunities for job-seeking students. For future improvements, this study involves experimenting with different approaches, including generating paragraphs using list templates and natural language. For instance, a list template might include "Current GPA = 3.50, Major = Data Science, and more," while converting it to natural language would involve transforming "3.00-3.49" to "between three and three point four nine." Additionally, enhancing the prediction of job sectors involves testing various embedding engines. The study will explore the Voyage embedding techniques, utilizing the "voyage-2" and "voyage-large-2" models to improve prediction accuracy.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for the careful review of our manuscript.

FUNDING STATEMENT

This project is funded and supported by TM R&D Fund (Telekom Malaysia Research and Development Fund) from Telekom Malaysia, Malaysia.

AUTHOR CONTRIBUTIONS

Bao-Ling Foo: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;
Choo-Yee Ting: Project Administration, Supervision, Writing – Review & Editing;
Hui-Ngo Goh: Project Administration, Writing – Review & Editing;
Albert Quek: Project Administration, Writing – Review & Editing;
Chin-Leei Cham: Project Administration, Writing – Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests was disclosed.

ETHICS STATEMENTS

Our publication ethics follow The Committee of Publication Ethics (COPE) guideline. https://publicationethics.org/

REFERENCES

[1] B. Dellal-Hedjazi and Z. Alimazighi, “Deep learning for recommendation systems,” 2020 6th IEEE Congress on Information Science and Technology (CiSt), pp. 90–97, Jun. 2020. doi: 10.1109/cist49399.2021.9357241.

[2] Z. Y. Poo, C. Y. Ting, Y. P. Loh, and K. I. Ghauth, “Multi-Label Classification with Deep Learning for Retail Recommendation,” Journal of Informatics and Web Engineering, vol. 2, no. 2, pp. 218–232, Sep. 2023. doi: 10.33093/jiwe.2023.2.2.16.

[3] J. Omana, P. N. Jeipratha, K. Devi, S. Benila, and K. Revathi, “Personalized Drug Recommendation System Using Wasserstein Auto-encoders and Adverse Drug Reaction Detection with Weighted Feed Forward Neural Network (WAES-ADR) in Healthcare,” Journal of Informatics and Web Engineering, vol. 4, no. 1, pp. 332–347, Feb. 2025. doi: 10.33093/jiwe.2025.4.1.24.

[4] M. N. Freire and L. N. De Castro, “e-Recruitment recommender systems: a systematic review,” Knowledge and Information Systems, vol. 63, no. 1, pp. 1–20, Nov. 2020. doi: 10.1007/s10115-020-01522-8.

[5] B. V. Santhosh Krishna, B. Rajalakshmi, I. Dsouza, J. Dsouza, X. Jeferson and K. Ashok, “E-commerce Trend Prediction Software,” 2024 IEEE 9th International Conference for Convergence in Technology (I2CT), Pune, India, 2024, pp. 1-4. doi: 10.1109/I2CT61223.2024.10543423.

[6] M. B. Savadatti and B. V. Santhosh Krishna, "Implementation of Collaborative Filtering in E-Commerce Recommender Systems: An Elementary Review," 2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN), Indore, India, 2024, pp. 917-922. doi: 10.1109/CICN63059.2024.10847463.

[7] K. Appadoo, M. B. Soonnoo and Z. Mungloo-Dilmohamud, "Job Recommendation System, Machine Learning, Regression, Classification, Natural Language Processing," 2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), Gold Coast, Australia, 2020, pp. 1-6. doi: 10.1109/CSDE50874.2020.9411584.

[8] H. Jain and M. Kakkar, “Job Recommendation System based on Machine Learning and Data Mining Techniques using RESTful API and Android IDE,” 2022 12th International Conference on Cloud Computing, Data Science &Amp; Engineering (Confluence), pp. 416–421, Jan. 2019. doi: 10.1109/confluence.2019.8776964.

[9] D. Mhamdi, R. Moulouki, M. Y. E. Ghoumari, M. Azzouazi, and L. Moussaid, “Job Recommendation based on Job Profile Clustering and Job Seeker Behavior,” Procedia Computer Science, vol. 175, pp. 695–699, Jan. 2020. doi: 10.1016/j.procs.2020.07.102.

[10] S. Azizi *et al.*, “Job recommendation system using machine learning and natural language processing,” Dublin Business School, 2020. doi: 10.1007/s43621-024-00292-5.

[11] J. Dhameliya and N. Desai, “Job Recommender Systems: a survey,” 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1–5, Mar. 2019. doi: 10.1109/i-pact44901.2019.8960231.

[12] Q. Zhou, F. Liao, L. Ge, and J. Sun, “Personalized Preference Collaborative Filtering: Job Recommendation for Graduates,” IEEE Conference Publication, pp. 1055–1062, Aug. 2019. doi: 10.1109/smartworld-uic-atc-scalcom-iop-sci.2019.00203.

[13] D. Punitavathi, V. Shinu, S. Siva Kumar, and S. P. Vidhya Priya, “Online job and Candidate recommendation system,” International Research Journal of Multidisciplinary Technovation, pp. 84–89, Mar. 2019. doi: 10.34256/irjmt19212.

[14] M. C. Urdaneta-Ponte, A. Mendez-Zorrilla, and I. Oleagordia-Ruiz, “Lifelong Learning Courses Recommendation System to improve professional skills using ontology and machine learning,” Applied Sciences, vol. 11, no. 9, p. 3839, Apr. 2021. doi: 10.3390/app11093839.

[15] J. Yao, Y. Xu, and J. Gao, “A study of reciprocal job recommendation for college graduates Integrating semantic keyword matching and social networking,” Applied Sciences, vol. 13, no. 22, p. 12305, Nov. 2023. doi: 10.3390/app132212305.

[16] Q. Wan and L. Ye, "Career recommendation for college students based on deep learning and machine learning," Scientific Programming, vol. 2022, pp. 1–10, Feb. 2022. doi: 10.1155/2022/3437139.

[17] B. E. V. Comendador, W. F. C. Becbec, and J. R. P. De Guzman, "Implementation of fuzzy logic technique in a decision support tool: Basis for choosing appropriate career path," International Journal of Machine Learning and Computing, vol. 10, no. 2, pp. 339–345, Feb. 2020. doi: 10.18178/ijmlc.2020.10.2.940.

[18] M. Qamhieh, H. Sammaneh, and M. N. Demaidi, "PCRS: Personalized Career-Path Recommender System for Engineering students," IEEE Access, vol. 8, pp. 214039–214049, Jan. 2020. doi: 10.1109/access.2020.3040338.

[19] A. H. A. Rashid, M. Mohamad, S. Masrom, and A. Selamat, "Student Career Recommendation System Using Content-Based Filtering Method," 2022 3rd International Conference on Artificial Intelligence and Data Sciences (AiDAS), Sep. 2022. doi: 10.1109/aidas56890.2022.9918766.

[20] P. C. Siswipraptini, H. L. H. S. Warnars, A. Ramadhan, and W. Budiharto, "Trends and Characteristics of Career Recommendation Systems for Fresh Graduated Students," 2022 10th International Conference on Information and Education Technology (ICIET), Apr. 2022. doi: 10.1109/iciet55102.2022.9779037.

[21] P. Feng, C. J. Jiang, J. Wang, S. Yeung, and X. Li, "Job Recommendation System Based on Analytic Hierarchy Process and K-means Clustering," 13th International Conference on Computer Modeling and Simulation, pp. 104–113, Jun. 2021. doi: 10.1145/3474963.3474978.

[22] D. Mhamdi, S. Ounacer, M. Msalek, M. Y. E. Ghoumari, and M. Azzouazi, "Job recommendation based on recurrent neural network approach," Procedia Computer Science, vol. 220, pp. 1039–1043, Jan. 2023. doi: 10.1016/j.procs.2023.03.145.

[23] A. Rivas, P. Chamoso, A. Gonzalez-Briones, R. Casado-Vara, and J. M. Corchado, "Hybrid job offer recommender system in a social network," Expert Systems, vol. 36, no. 4, May 2019. doi: 10.1111/exsy.12416.

[24] G. Zhu, N. A. Kopalle, Y. Wang, X. Liu, K. Jona, and K. Borner, "Community-based data integration of course and job data in support of personalized career-education recommendations," Proceedings of the Association for Information Science and Technology, vol. 57, no. 1, Oct. 2020. doi: 10.1002/pra2.324.

[25] Z. Zhalgassova, A. Shaikym, U. Sadyk, A. Kutzhan, M. Amirkumar, and B. Assangali, "MBTI-based recommendation system for extracurricular activities for high school students," 2023 17th International Conference on Electronics Computer and Computation (ICECCO), pp. 1–4, Jun. 2023. doi: 10.1109/icecco58239.2023.10147138.

[26] R. Mishra and S. Rathi, "Enhanced DSSM (deep semantic structure modelling) technique for job recommendation," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 7790–7802, Jul. 2021. doi: 10.1016/j.jksuci.2021.07.018.

[27] A. Mulay, S. Sutar, J. Patel, A. Chhabria, and S. Mumbaikar, "Job recommendation system using hybrid filtering," ITM Web of Conferences, vol. 44, p. 02002, Jan. 2022. doi: 10.1051/itmconf/20224402002.

[28] A. Kara, F. S. Danis, G. K. Orman, S. N. Turhan, and O. A. Ozlu, "Job recommendation based on extracted skill embeddings," in Lecture notes in networks and systems, 2022, pp. 497–507. doi: 10.1007/978-3-031-16075-2_35.

[29] E. Pang, M. Wong, C. H. Leung, and J. Coombes, "Competencies for fresh graduates' success at work: Perspectives of employers," Industry and Higher Education, vol. 33, no. 1, pp. 55–65, Aug. 2018. doi: 10.1177/0950422218792333.

[30] T. Siskos, "Career recommendations using supervised latent Dirichlet allocation," 2020. doi: 10.18452/21341.

[31] L. D. Kumalasari and A. Susanto, "Recommendation System of Information Technology Jobs using Collaborative Filtering Method Based on LinkedIn Skills Endorsement," SISFORMA, vol. 6, no. 2, pp. 63–72, Feb. 2020. doi: 10.24167/sisforma.v6i2.2240.

[32] A. Mughaid, I. Obeidat, B. Hawashin, S. AlZu'bi, and D. Aqel, "A smart Geo-Location job recommender system based on social media posts," 2019 6th International Conference on Social Networks Analysis, Management, and Security (SNAMS), pp. 505–510, Oct. 2019. doi: 10.1109/snams.2019.8931854.

[33] Y. Mao, Y. Cheng, and C. Shi, "A job recommendation method based on attention layer scoring characteristics and tensor decomposition," Applied Sciences, vol. 13, no. 16, p. 9464, Aug. 2023. doi: 10.3390/app13169464.

[34] P. K. Roy, S. S. Chowdhary, and R. Bhatia, "A Machine Learning approach for automation of Resume Recommendation system," Procedia Computer Science, vol. 167, pp. 2318–2327, Jan. 2020. doi: 10.1016/j.procs.2020.03.284.

[35] Md. S. Hossain and M. S. Arefin, "Development of an Intelligent Job Recommender System for Freelancers using Client's Feedback Classification and Association Rule Mining Techniques," Journal of Software, pp. 312–339, Jul. 2019. doi: 10.17706/jsw.14.7.312-339.

[36] S. Gadegaonkar, D. Lakhwani, S. Marwaha, and A. Salunke, "Job Recommendation System using Machine Learning," 2023 3rd International Conference on Artificial Intelligence and Smart Energy (ICAIS), pp. 596–603, Feb. 2023. doi: 10.1109/icais56108.2023.10073757.

[37] L. Tian, "Next career recommendation in Mississippi with artificial intelligence," Journal of Computational and Applied Mathematics, vol. 437, p. 115458, Jul. 2023. doi: 10.1016/j.cam.2023.115458.

[38] R. N. Ravikumar, S. Jain, and M. Sarkar, "AdaptiLearn: real-time personalized course recommendation system using whale optimized recurrent neural network," International Journal of Systems Assurance Engineering and Management, Apr. 2024. doi: 10.1007/s13198-024-02301-2.

[39] C. S. Kumar, M. P. Deeraj, K. N. H. Vardhan, K. Amulya, and K. Govardhan, "Fashionista a Personalized Fashion and Style Recommendation System with Machine Learning Insights," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1898–1904, Mar. 2024. doi: 10.1109/icaccs60874.2024.10717000.

[40] J. Hao, Z. Liu, Q. Sun, C. Zhang, and J. Wang, "A Static-Dynamic Hypergraph neural network framework based on residual learning for stock recommendation," Complexity, vol. 2024, pp. 1–12, Jan. 2024. doi: 10.1155/2024/5791802.

[41] M. Kayed, F. Azzam, H. Ali, and A. Ali, "Temporal dynamics of user activities: deep learning strategies and mathematical modeling for long-term and short-term profiling," Scientific Reports, vol. 14, no. 1, Jun. 2024. doi: 10.1038/s41598-024-64120-6.

[42] S. Suman, S. J. Kaur, A. Sharma, and S. Kumar, "Machine Learning-Based System for Admission and Jobs Prediction in Engineering and Technology Sector," 2024 IEEE International Conference on Computing, Power, and Communication Technologies (IC2PCT), pp. 463–468, Feb. 2024. doi: 10.1109/ic2pct60090.2024.10486533.

[43] N. B. Muddangala, T. S. Senthilkumar, S. S, A. S, A. M. Tadesse, and S. S, "A Privacy-Preserving, explainable job recommendation system using Transformer-Based NLP and career path prediction," SSRN Electronic Journal, Jan. 2025. doi: 10.2139/ssrn.5085468.

[44] Z. Yu and B. Li, "Reinforced concrete beam full response prediction with hybrid feature-orientation transformer-LSTM model," Engineering Structures, vol. 332, p. 120040, Mar. 2025. doi: 10.1016/j.engstruct.2025.120040.

[45] Y. Mahale et al., "Crop recommendation and forecasting system for Maharashtra using machine learning with LSTM: a novel expectation-maximization technique," Discover Sustainability, vol. 5, no. 1, Jun. 2024. doi: 10.1007/s43621-024-00292-5.

[46] X. Xiao, C. Li, X. Wang, and A. Zeng, "Personalized tourism recommendation model based on temporal multilayer sequential neural network," Scientific Reports, vol. 15, no. 1, Jan. 2025. doi: 10.1038/s41598-024-84581-z.

[47] N. Hameed, A. M. Shabut, M. K. Ghosh, and M. A. Hossain, "Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques," Expert Systems with Applications, vol. 141, p. 112961, Sep. 2019. doi: 10.1016/j.eswa.2019.112961.

[48] M. R. Hassan, S. Huda, M. M. Hassan, J. Abawajy, A. Alsanad, and G. Fortino, "Early detection of cardiovascular autonomic neuropathy: A multi-class classification model based on feature selection and deep learning feature fusion," Information Fusion, vol. 77, pp. 70–80, Aug. 2021. doi: 10.1016/j.inffus.2021.07.010.

[49] K. N. Prafajar, H. Vallyan, N. L. P. A. Candradewi, I. S. Edbert, and D. Suhartono, "Multiclass job recommendation system in the IT field between classification and prediction method," 2022 International Conference on Green Energy, Computing, and Sustainable Technology (GECOST), pp. 181–186, Oct. 2022. doi: 10.1109/gecost55694.2022.10010659.

[50] H. T. Sueno, "Multi-class Document Classification using Support Vector Machine (SVM) Based on Improved Naive Bayes Vectorization Technique," International Journal of Advanced Trends in Computer Science and Engineering, vol. 9, no. 3, pp. 3937–3944, Jun. 2020. doi: 10.30534/ijatcse/2020/216932020.

[51] I. Iqbal, M. Younus, K. Walayat, M. U. Kakar, and J. Ma, "Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images," Computerized Medical Imaging and Graphics, vol. 88, p. 101843, Dec. 2020. doi: 10.1016/j.compmedimag.2020.101843.

[52] F. Jourdan, T. T. Kaninku, N. Asher, J.-M. Loubes, and L. Risser, "How optimal transport can tackle gender biases in Multi-Class Neural Network Classifiers for job recommendations," Algorithms, vol. 16, no. 3, p. 174, Mar. 2023. doi: 10.3390/a16030174.

[53] E. Rendon, R. Alejo, C. Castorena, F. J. Isidro-Ortega, and E. E. Granda-Gutierrez, "Data sampling methods to deal with the big data Multi-Class imbalance problem," Applied Sciences, vol. 10, no. 4, p. 1276, Feb. 2020. doi: 10.3390/app10041276.

[54] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," Computers and Electronics in Agriculture, vol. 174, p. 105507, May 2020. doi: 10.1016/j.compag.2020.105507.

[55] W. Qiang, J. Zhang, L. Zhen, and L. Jing, "Robust weighted linear loss twin multi-class support vector regression for large-scale classification," Signal Processing, vol. 170, p. 107449, Dec. 2019. doi: 10.1016/j.sigpro.2019.107449.

[56] L. Sunitha and M. B. Raju, "Multi-class classification for large datasets with optimized svm by non-linear kernel function," Journal of Physics: Conference Series, vol. 2089, p. 012015, 2021. doi:1016/j.pcs.2021.012015

[57] A. Taherkhani, G. Cosma, and T. M. McGinnity, "AdaBoost-CNN: An adaptive boosting algorithm for convolutional neural networks to classify multi-class imbalanced datasets using transfer learning," Neurocomputing, vol. 404, pp. 351–366, May 2020. doi: 10.1016/j.neucom.2020.03.064.

[58] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," Journal of Big Data, vol. 7, no. 1, Sep. 2020. doi: 10.1186/s40537-020-00349-y.

[59] A. Bishnoi, F. Jaison, and D. Wadhwa, "Analyzing feature selection and dimensionality reduction for big data," 2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC), pp. 1–7, Jan. 2024. doi: 10.1109/icocwc60930.2024.10470927.

[60] C. J. Ejiyi et al., "Comparative performance analysis of Boruta, SHAP, and Borutashap for disease diagnosis: A study with multiple machine learning algorithms," Network Computation in Neural Systems, pp. 1–38, Mar. 2024. doi: 10.1080/0954898x.2024.2331506.

[61] G. Yue, "Screening of lung cancer serum biomarkers based on Boruta-shap and RFC-RFECV algorithms," Journal of Proteomics, vol. 301, p. 105180, Apr. 2024. doi: 10.1016/j.jprot.2024.105180.

[62] E. Akkur and A. C. Ozturk, "Predicting Lung Cancer Using Explainable Artificial Intelligence and Boruta-Shap Methods," Kahramanmaras Sutcu Imam Universitesi Muhendislik Bilimleri Dergisi, vol. 27, no. 3, pp. 792–803, Sep. 2024. doi: 10.17780/ksujes.1425483.

[63] T. Li, Y. Wu, F. Ren, and M. Li, "Estimation of unrealized forest carbon potential in China using time-varying Boruta-SHAP-random forest model and climate vegetation productivity index," Journal of Environmental Management, vol. 377, p. 124649, Feb. 2025. doi: 10.1016/j.jenvman.2025.124649.

[64] M. Jamei et al., "Monthly sodium adsorption ratio forecasting in rivers using a dual interpretable glass-box complementary intelligent system: Hybridization of ensemble TVF-EMD-VMD, Boruta-SHAP, and eXplainable GPR," Expert Systems with Applications, vol. 237, p. 121512, Sep. 2023. doi: 10.1016/j.eswa.2023.121512.

[65] C. A. Sequeira and E. M. Borges, "Enhancing statistical education in chemistry and STEAM using JAMOVI. Part 2. Comparing dependent Groups and Principal Component Analysis (PCA)," Journal of Chemical Education, Jun. 2024. doi: 10.1021/acs.jchemed.4c00342.

[66] A. Feraco et al., "Gender differences in dietary patterns and physical activity: an insight with principal component analysis (PCA)," Journal of Translational Medicine, vol. 22, no. 1, Dec. 2024. doi: 10.1186/s12967-024-05965-3.

[67] D. Hammoumi et al., "Seasonal variations and assessment of surface water quality using Water Quality Index (WQI) and Principal Component Analysis (PCA): a case study," Sustainability, vol. 16, no. 13, p. 5644, Jul. 2024. doi: 10.3390/su16135644.

[68] R. N. Bashir, O. Mzoughi, M. A. Shahid, N. Alturki, and O. Saidani, "Principal Component Analysis (PCA) and feature importance-based dimension reduction for Reference Evapotranspiration (ET0) predictions of Taif, Saudi Arabia," Computers and Electronics in Agriculture, vol. 222, p. 109036, May 2024. doi: 10.1016/j.compag.2024.109036.

[69] A. Coccato and M. C. Caggiani, "An overview of Principal Components Analysis approaches in Raman studies of cultural heritage materials," Journal of Raman Spectroscopy, vol. 55, no. 2, pp. 125–147, Nov. 2023. doi: 10.1002/jrs.6621.

[70] H. Hairani, T. Widiyaningtyas, and D. D. Prasetya, "Addressing class imbalance of health data: A Systematic Literature Review on Modified Synthetic Minority Oversampling Technique (SMOTE) strategies," JOIV International Journal on Informatics Visualization, vol. 8, no. 3, p. 1310, Sep. 2024. doi: 10.62527/joiv.8.3.2283.

[71] R. Bounab, K. Zarour, B. Guelib, and N. Khlifa, "Enhancing Medicare fraud detection through Machine Learning: Addressing class imbalance with SMOTE-ENN," IEEE Access, vol. 12, pp. 54382–54396, Jan. 2024. doi: 10.1109/access.2024.3385781.

[72] G. Husain et al., "SMOTE vs. SMOTEENN: A Study on the Performance of Resampling Algorithms for Addressing Class Imbalance in Regression Models," Algorithms, vol. 18, no. 1, p. 37, Jan. 2025. doi: 10.3390/a18010037.

[73] S. A. Gamel, S. S. M. Ghoneim, and Y. A. Sultan, "Improving the accuracy of diagnostic predictions for power transformers by employing a hybrid approach combining SMOTE and DNN," Computers & Electrical Engineering, vol. 117, p. 109232, Apr. 2024. doi: 10.1016/j.compeleceng.2024.109232.

[74] R. Nithya, T. Kokilavani, and T. L. A. Beena, "Balancing cerebrovascular disease data with integrated ensemble learning and SVM-SMOTE," Network Modeling Analysis in Health Informatics and Bioinformatics, vol. 13, no. 1, Mar. 2024. doi: 10.1007/s13721-024-00447-4.

[75] M. Y. Cakir and Y. Sirin, "Enhanced autoencoder-based fraud detection: a novel approach with noise factor encoding and SMOTE," Knowledge and Information Systems, vol. 66, no. 1, pp. 635–652, Nov. 2023. doi: 10.1007/s10115-023-02016-z.

[76] K. N. Lajis and H. Hashim, "Comparison of rule-based chatbot versus AI chatbot: The case of restaurant recommendations," in Information Science and Digital Society, S. M. Tan, T. W. Liew, and H. F. Neo, Eds. MMU Press, 2023, pp. 209–233.

BIOGRAPHIES OF AUTHORS

	**Bao-Ling Foo** is a student at the Faculty of Computing and Informatics (FCI), Multimedia University, Cyberjaya, Malaysia. She specializes in data science, with research interests in natural language processing (NLP), computer vision, machine learning, and recommendation systems. She can be contacted at fbaoling01@gmail.com
	**Choo-Yee Ting** is a Professor at the Faculty of Computing and Informatics (FCI), Multimedia University, Cyberjaya, Malaysia. He was awarded the Fellow of Microsoft Research in 2002 and has led research in predictive analytics and Big Data, funded by MOE, MOSTI, Telekom Malaysia, and MDeC. Additionally, he has won national Big Data competitions and serves as a trainer, consultant, and accessor. Furthermore, he supports Malaysia's National Immunisation Programme for COVID-19 and works on AI-driven projects for Telekom Malaysia and AirAsia. He can be contacted at cyting@mmu.edu.m
	**Hui-Ngo Goh** is an Assistant Professor at Faculty of Computing and Informatics (FCI), Multimedia University, Cyberjaya, Malaysia. With over 20 years of teaching experience in Computer Science and Information Technology, she has been actively involved in supervising undergraduate and postgraduate students, leading government-funded research projects, designing up-to-date syllabi, and guiding students in competitions such as Hackathon. Her extensive experience and dedication contribute significantly to the academic community. She can be reached at hngoh@mmu.edu.my
	**Albert Quek** is a Lecturer at the Faculty of Computing and Informatics (FCI), Multimedia University, Cyberjaya, Malaysia. He also holds the position of Deputy Director of Alumni Engagement, Career, and Entrepreneurship Development. His role involves engaging with alumni, fostering career development, and promoting entrepreneurial initiatives among students. He can be contacted at quek.albert@mmu.edu.my
	**Chin-Leei Cham** is a Lecturer at the Faculty of Artificial Intelligence and Engineering (FAIE), Multimedia University, Cyberjaya, Malaysia. His research areas include artificial intelligence, robotics, and automation systems. He is interested in AI-driven robotics for industrial and assistive applications, focusing on machine learning integration in automation. He has published several papers on AI algorithms and robotic intelligence. He is also involved in interdisciplinary research projects. He can be reached at clcham@mmu.edu.my