CTU Journal of Innovation and Sustainable Development
ISSN 2615-9422 | e-ISSN 2815-5602

DOI:10.22144/ctujoisd.2023.036

Career path prediction using XGBoost Model and students' academic results

Hong Quan Nguyen, Duc Dang Khoi Nguyen, Tan Duy Le, An Mai, and Kha Tu Huynh*
International University, Viet Nam National University, Ho Chi Minh City, Viet Nam
**Corresponding author (hktu@hcmiu.edu.vn)*

Article info.

Received 30 Jul 2023
Revised 10 Sep 2023
Accepted 27 Sep 2023

Keywords

Career path prediction, decision making, deep learning, tree classification, XGBoost decision

ABSTRACT

This paper proposes an approach for constructing a system for career prediction by applying the eXtreme Gradient Boosting (XGBoost) Decision Tree model to the academic results of Ho Chi Minh International University's School of Computer Science and Engineering graduates in the past 5 years. Initially, the dataset is cleaned up and normalized to be usable for the prediction algorithm with the help of Python 3 programming language. It is then split into 2 subsets: one for training (80 percent) and the other for testing (20 percent). After that, the algorithm uses the training subset to build the classification model. Finally, the testing subset is loaded into the model to predict each student's career path based on the respective inputs and hyper-parameters tuning is employed to boost the model's accuracy. By utilizing this solution, the problem related to predicting students' future career paths based on their performance throughout their years studying at the university can be adequately addressed and handled.

1. INTRODUCTION

Choosing a career has always been serious and strenuous for almost anyone, as it is a decision that can significantly impact one's future. Choosing a career path may seem simple, but finding one that makes you feel comfortable and benefits both your personal and professional development is an entirely different and highly challenging issue. As the philosopher Confucius once said, "Choose the job you love, and you will never have to work a day in your life." (Poznanski, 2015) This teaching emphasizes the importance of pursuing a profession that aligns with your passions and interests.

For students, particularly those in university, selecting their future careers can be overwhelming and painstaking. Usually, students find themselves deliberately considering what they want to do with their lives while seeking happiness and satisfaction in their chosen path. However, this task is often far more challenging than any academic endeavor they have faced in their university years to date. Choosing a career is often riddled with uncertainty and ambiguity, making it seem impossible for numerous students to determine their ideal route. Some students are torn between furthering their education and entering industry to gain practical work experience. On the other hand, some deliberate whether studying abroad could provide them with better opportunities than staying in their home country for education.

This complexity is caused by students lacking a deeper understanding of their strengths and weaknesses and being unsure about what they genuinely want to pursue. While academic results are one type of indicator, they alone cannot provide the necessary insights into how these results directly relate to a student's career choice. After all, numbers are just numbers; without a proper statistical summary and analysis, they cannot unveil the precise connection between academic performance and career aspirations. Students yearn for a more comprehensive tool to bridge this gap and guide

them toward making informed decisions about their future.

Thus, this is where machine learning comes into play, offering practical, stable, and scalable solutions to address the specific problem of career prediction for university students. A simple decision tree model can be built to predict the most suitable career path based on students' academic performance. Such a model may even consider factors aside from grades, including extracurricular activities, personal interests, and achievements, to provide students with valuable insights and recommendations for their future. This research aims to develop a robust decision tree model, namely the XGBoost Decision Tree model, for career prediction, taking students' academic performance as a crucial input to assist them in making precise decisions about their future profession.

Following this introduction, the paper presents the six contents including Problem statements, Related works, The proposed model, Results, and Conclusion.

2. PROBLEM STATEMENTS

Career path prediction is undoubtedly one of the most intriguing and practical topics regarding decision-making in machine learning. Numerous approaches, algorithms, and models have been established and developed to realize this decision-making process. Despite that, predicting a student's career path depends heavily on the inputs (e.g., all the subjects' scores, a summary of the Grade Point Average (GPA), overall academic performance, personality, social activities...). To give a proper and sensible prediction on students' career paths is not a simple task, and the results vary based on the available information. Hence, further research on each student's individuality is required, as much data related to the students who participate in the prediction process should be collected to make the results as fair and unbiased as possible. Since everyone is their person, the problem does not end with figuring out which model suits the training and predicting procedures. As stated before, aside from their academic scores, other aspects that define each student's individuality should also be considered to produce sound predictions. In addition, aside from the psychological inputs, the prediction model should consider another extremely vital factor, data processing time.

3. RELATED WORKS

3.1. Student career prediction using Decision Tree Classification

The Decision Tree model is a Supervised Learning method and can be utilized for both regression and classification tasks. However, it is most preferred for tackling classification issues. A decision tree is a tree-structured classifier in which the tree's nodes represent the dataset's features, the tree's branches represent the defining rules for the decision-making process, and the leaf nodes represent the outcomes. The starting node of the tree (the node at the top level – level 0) is called the Root Node, and it is the starting point of the decision tree and the classification process. Aside from the Root Node, there are two other types of nodes in a decision tree: Decision Node and Leaf Node. The Decision Nodes are where decisions are made and have branches to go to either other Decision Nodes or Leaf Nodes. The Leaf Nodes, however, are the ending outcomes of said decisions and do not branch to any other node. Some commonly used decision trees are ID3 (Quinlan, 1986), C4.5 (Quinlan, 2014), C5.0, CART (Classification and Regression Tree) (Loh, 2011), CHAID (Chi-squared Automatic Interaction Detector). It is precisely due to their simplicity and accuracy that decision tree models have been becoming increasingly widely used in recent years to predict student performance and career paths in Educational Data Mining (EDM) (Križanić, 2020). One study (Jamal et al., 2020) used decision tree models such as C5.0 and CHAID to predict the profession choices among the graduates of a university's department in the span of 25 years (from 1995 to 2019), which yielded a prediction accuracy of approximately 92.6 percent. Moreover, along with the accuracy, they also gained more insight into the factors influencing those graduates' career choices, namely CGPA, additional expertise, and gender.

Another research (Casuat et al., 2019) done on a dataset comprising of Mock job Interview Results, Student Performance Ratings, and General Point Average proved that by applying the decision tree model, the students' employability could be predicted with an accuracy of approximately 85 percent. These researchers also performed evaluations on the performance of other machine learning models: Random Forest (RF) and Support Vector Machine (SVM), which yielded accuracies of approximately 84 percent and 91 percent, respectively. This showed that the decision tree

model could correctly predict the possibility of a student's current employability based on the information gathered about their academic performance and behavior.

The scholarly work Saa (2016) focused on analyzing students' performance during their higher education journey using EDM techniques to detect patterns and trends. The study also explored the intricate relationship between social and academic factors and their influence on students' academic achievements. Consequently, the research strived to establish a model capable of effectively classifying and predicting students' performance. A dataset consisting of 270 records was collected via Google Forms. Subsequently, the collected data was visualized using various tools such as graphs, plots, and diagrams to gain comprehensive insights. The researchers employed various data mining methodologies to generate predictions, particularly decision tree classification algorithms, including C4.5, ID3, CART, and CHAID. Among these algorithms, CART demonstrated the highest accuracy of 40 percent, followed by C4.5, with an accuracy of 35.19 percent. In addition, CHAID yielded an accuracy of 34.07 percent, while ID3 attained an accuracy of 33.33 percent. The researchers also applied the Naive-Bayes algorithm to the dataset, resulting in an accuracy of 36.40 percent. The study suggests that a student's performance cannot be solely attributed to academic factors alone, as other factors also play an equally significant role in shaping their academic journey and achievements. Therefore, an approach considering various factors is essential for comprehensively understanding and predicting students' performance in the educational context.

Despite their notable benefits and high predictive accuracy, simple decision-tree models have drawbacks. One major drawback is that they are prone to overfitting (Panhalkar, 2022), where they turn to more complex and try to memorize the training data instead of searching for ways to generalize to new and unseen data. This overfitting issue can lead to poor performance when applied to real-world scenarios. Moreover, decision trees lack robustness (Wang et al., 2021; Charbuty & Abdulazeez 2021), making them sensitive to even minor changes in the training data. Such sensitivity can cause significant variations in the resulting model, affecting its reliability and consistency. In addition, decision trees sometimes require extensive tweaking and preprocessing to effectively handle missing data, which is another common occurrence in real-world datasets. Dealing with missing data in decision trees often requires complex workarounds (Emmanuel et al., 2021), which can be time-consuming and computationally expensive.

Consequently, these limitations have led researchers and practitioners to create advanced decision tree types, such as Gradient Boosting, to overcome some of the drawbacks normal decision trees usually face. In this paper, the XGBoost Decision Tree algorithm, one of the most popular Gradient Boosting algorithms, is adopted to address these challenges and improve the performance and robustness of the decision tree model.

3.2. Student career prediction using XGBoost Decision Tree Classification

XGBoost (eXtreme Gradient Boosting) is a popular decision tree machine-learning algorithm for regression and classification tasks proposed by Chen et al. (2016). As it is called "eXtreme", the enhanced version of the gradient boosting algorithm uses decision trees in combinations to improve the model performance. In XGBoost, decision trees are created iteratively, with each new tree attempting to correct errors made by the previous tree. Additionally, XGBoost employs regularization techniques such as L1 and L2 regularization to prevent overfitting and improve general performance. It also includes many additional features designed specifically to improve performance, such as handling missing data and hyperparameters for model optimization.

In their research, Vignesh et al., (2020) conducted a study using XGBoost on a vast dataset comprising various aspects such as students' academic ability, competition involvement, programming languages, and personal interests to predict their future career paths. The researchers collected the data from multiple platforms, namely Twitter, Kaggle, Google Forms (and more), and consolidated this information into 15 distinct profession labels inhabiting the target variable, along with 36 other parameters characterizing the students for the model. To facilitate the classification process, the researchers applied the One-Hot Encoding technique, effectively transforming the categorical labels in the dataset into numerical values and enabling XGBoost to classify the data accurately. The performance of the final predictive model was assessed using the Confusion Matrix, which showed good accuracy in predicting students' career choices and highlighted the model's capability to provide

accurate predictions and valuable insights into the students' characteristics.

Besides the aforementioned study, another study (Nie et al., 2020) conducted on over 4,000 students over four years produced a successful career prediction model by applying XGBoost. This study collected data from students of the same grade from 16 different colleges, with various features falling into four categories: basic information, academic performance, behavioral characteristics, and career choice. The researchers then split the data and used 70 percent of it for training the predictive model, which was further split and categorized based on each college. The model's hyperparameters were tuned using the 5-fold cross-validation to boost its accuracy. The researchers' final model could predict with over 60 percent accuracy compared to the regular decision tree, which yielded only 53 percent accuracy. These findings suggest that using multiple data sources and applying advanced machine-learning techniques can accurately predict students' future career paths.

Roy et al. (2018) conducted a study focused on analyzing students' academic performance, specializations, preferred programming languages, analytical skills, and even various personal information; and used XGBoost to develop a predictive model for students' career areas. The dataset encompassed diverse sources, including LinkedIn profiles, employed individuals from various organizations, college alumni databases, and even randomly generated data. The researchers compiled a comprehensive dataset with 36 distinct features and approximately 20,000 historical student records. Of these records, 80 percent were employed for training the model. Multiple preprocessing techniques and One-Hot Encoding were employed to prepare the data for compatibility with the XGBoost algorithm. The researchers reported the XGBoost model achieved a decent accuracy of 88.33 percent in predicting students' career choices, thus highlighting the algorithm's efficacy and predictive prowess.

The utilization of tree-based supervised machine-learning algorithms has been substantiated by the previously mentioned studies as highly suitable and appropriate for the examination and analysis of educational datasets. XGBoost's ability to capture complex non-linear relationships and handle diverse features, coupled with its track record of achieving remarkable performance and accuracy, sets it as a valuable and highly regarded tool for acquiring insights into the factors that influence students' career decisions. Because of these reasons, this research uses it for constructing the model on students' career path prediction for Ho Chi Minh International University's Department of Computer Science and Engineering and will apply to other universities.

4. THE PROPOSED MODEL

4.1. Methodology

This research employs the eXtreme Gradient Boosting (XGBoost) Decision Tree algorithm to analyze the academic performance data of the graduates majoring in Computer Science (CS) from the School of Computer Science and Engineering (SCSE) at Ho Chi Minh International University. The primary aim is to develop a predictive model that can forecast the career decisions of these graduates and future students of the faculty. The predictive model is constructed to achieve highly accurate predictions by leveraging XGBoost's ability for self-improvement and its extensive range of optimization hyperparameters. By adopting this approach, students can acquire valuable insights into the many career paths available to them based solely on their academic performance throughout their years spent studying at the university. Applying XGBoost in this context offers an opportunity to enhance career decision-making processes among the university's students.

4.2. Research design

Figure 1 illustrates the flowchart outlining the procedures and methods employed in this study. Data comprising students' academic results and career choice labels are initially collected and organized to create a dataset. Subsequently, the dataset undergoes a cleaning process to eliminate duplicated and redundant features using Microsoft Excel (or Google Spreadsheets) and Python codes in Google Colaboratory. Once all missing values are addressed and the dataset is refined to retain only useful and relevant features, each feature is normalized and standardized to ensure uniformity for the XGBoost model.

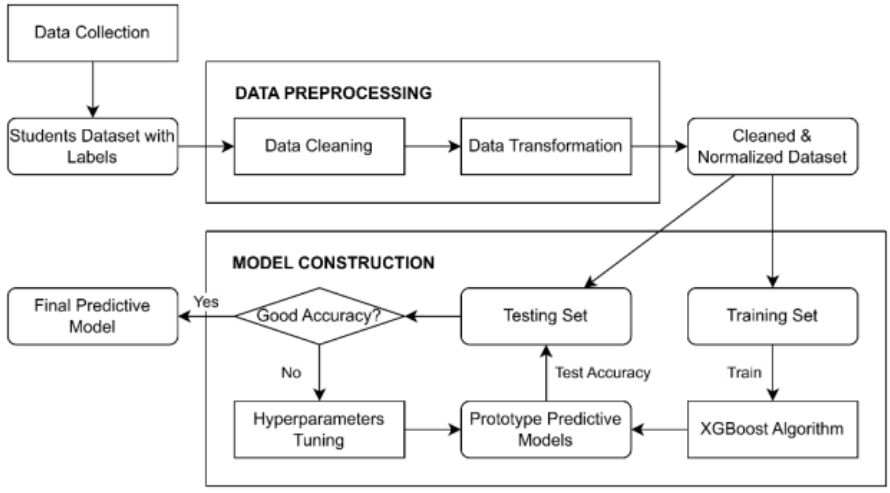

Figure 1. The research design of the proposed predictive model

The standardized dataset is divided into two subsets: one training set with 80 percent of the original dataset, and the rest of the 20 percent is used for testing. First, the training set is inputted into the XGBoost Classifier with default parameters to generate a default classification model. The model's accuracy is evaluated by testing in on the designated testing set. If the accuracy falls short of the desired level, hyperparameter tuning is employed to optimize the model. The final model is obtained upon completion of the tuning process, enabling predictions on new instances. This final step signifies the conclusion of the research.

4.3. Data collection and preprocessing

4.3.1. Data Sources

The data utilized in this study is provided by the Ho Chi Minh International University's Office of Academic Affairs. The dataset, in its original form, contains 177 graduates of the university's Faculty of Computer Science and Engineering enrolled at the university in the year from 2014 to 2018 and encompasses many essential features, including the graduates' ID, the ID, name, and credit counts associated with the courses they undertook during their academic years, as well as the specific semesters and years in which these courses were completed.

The dataset includes a comprehensive list of scores reflecting the performance of each graduate in their respective courses. These scores encompass various evaluative components, such as in-class assessments, midterm examinations, final examinations, and each course's calculated Grade Point Average (GPA). Additionally, the dataset contains information regarding the total number of credits amassed by each graduate student throughout their academic years, alongside the cumulative GPA achieved throughout their enrolment at the university.

4.3.2. Data Cleaning

Due to the relatively small dataset, the first dataset cleaning is done via Microsoft Excel (or Google Spreadsheets) and handles only some basic operations and dataset reorganizations. As the given data spreads the courses taken on each row, it cannot be inputted into the training model. This study proposes utilizing the VLOOKUP functions, combined with ARRAYFORMULA and some other operational functions, to reshape the dataset so that each row corresponds to a single graduate only. The graduates' ID is retained as the first column, the rest of the columns are spread out with all courses found in the original dataset, and the last three columns are the accumulated credits, cumulative GPA, and career choice, which is the target label of this study's predictive model. All the reshaping mentioned above is done on a new sheet and will, later on, be extracted further to obtain the most condensed dataset that can be obtained using functions on Microsoft Excel.

Upon closer inspection, the dataset reveals some inherently similar columns, meaning that the graduates likely took the courses with almost similar content but different names and belonged to different faculties. This issue will be handled manually by merging the columns directly in the Excel data file but on a different sheet than the one

mentioned before. Some graduates took courses belonging to either the Computer Engineering (CE) major or the Network Engineering (NE) major and did not take some required courses of the Computer Science (CS) major. Since this research targets CS students only, the graduates who were likely to be students of CE and NE majors shall be omitted from the dataset. The cleaning using Microsoft Excel will be discussed in greater detail in the upcoming chapter.

Naturally, some graduates took different courses than others, leading to the most common yet severe problem: missing values (Alabadla et al., (2022)). After being reorganized and shortened using Microsoft Excel, the dataset now shows many missing values; this is due to the existence of extra-curricular courses. A quick skim through the dataset hints that only two to three graduates took some extra-curricular classes and that these courses can be dropped from the dataset without causing any interference to the construction of the predictive model. In any case, handling the extra-curricular courses and the missing values is done on Google Colaboratory using Python 3 programming language. Again, further details are to be disclosed in the next section.

4.3.3. Data transformation

Once the dataset has undergone the crucial process of data cleaning, it is important to employ various data transformation techniques in order to ensure homogeneity across all the features. For instance, within the dataset, the scores of each course are measured on a scale ranging from 0 to 100. In contrast, other features, such as accumulated credits and career choices, do not share the same scale. While one feature represents the cumulative count of credits, the other is the categorical prediction label. Consequently, in order to establish comparability, it is essential to standardize all the features by utilizing the widely adopted Z-Score formula. The statistical formula for a value's z-score is calculated using the formula (1):

$$z = (x - \mu) / \sigma \quad (1)$$

where z is Z-score, x is the value being evaluated, μ is the mean and σ is the standard deviation.

Moreover, with the categorical prediction label, encoding techniques are employed to transform its representation into a suitable format that can be effectively utilized in constructing the predictive model.

4.4. XGBoost algorithm

eXtreme Gradient Boosting, commonly referred to as XGBoost, is a powerful gradient boosting and decision-tree-based ensemble machine learning algorithm introduced by Chen et al., (2016). Its underlying principle involves the aggregation of predictions from many weak decision tree models, leading to the creation of a more accurate and robust one. XGBoost builds upon the traditional gradient boosting model, which iteratively incorporates new models with the purpose of correcting the mistakes and errors made by preceding models.

In recent years, XGBoost has garnered significant attention, as shown in a diversity of studies and research, highlighted by its application in fields such as health diagnosis (Bao, 2020) (Budholiya et al., 2022), stock forecast (Gumelar et al., 2020) (Wang & Guo, 2020), and even career prediction. The overall computational complexity of the XGBoost algorithm is the combination of all the complexities mentioned above, with particular emphasis on the tree construction and the tree update steps. While these steps contribute the most to the algorithm's computational complexity, the optimized implementation of XGBoost offers efficient computation times, even when handling large-scale datasets.

In summary, XGBoost presents an efficient approach to mitigating the adverse effects of overfitting and enhancing model generalization through integrating regularization terms that control the complexity of the model. Moreover, XGBoost's optimized computational complexity significantly contributes to its superiority over numerous other machine-learning algorithms, such as the standard decision tree and gradient boosting algorithms. Therefore, this study deliberately selects XGBoost as the preferred algorithm for constructing predictive models, owing to its demonstrated merits.

4.5. The proposed model

4.5.1. Training Procedure

As mentioned above, 80 percent of the dataset is split and made into the training subset for the XGBoost algorithm. The training process begins with plugging the training sets into the standard XGBoost classifier and evaluates the generated model's accuracy using various metrics. Subsequently, the model is further refined using hyperparameter tuning. By the end of the tuning process, a more accurate model is expected to be obtained.

4.5.2. Hyperparameter tuning

Performing hyper-parameter tuning for XGBoost involves a systematic search to identify the optimal combination of hyper-parameters that maximizes the model's performance for a given problem. Various methods and techniques have been researched and proposed to address this challenge, each offering its advantages and disadvantages. Table 2 below provides a concise overview of the four noteworthy hyper-parameter tuning methods.

Table 1. Evaluation metrics used in this study

Metrics	Description
Accuracy	Quantifies the proportion of correctly classified instances relative to the total instances. A high accuracy value suggests that the model's predictions are more precise.
Precision and Recall	Precision assesses the ratio of accurately predicted positive instances to the total number of instances predicted as positive. A higher precision value indicates that positive predictions are more likely to be accurate, reflecting the model's ability to minimize false positives. Recall assesses the ratio of accurately predicted positive instances to the total number of truly positive instances. A higher recall value signifies the model's effectiveness in capturing a more significant proportion of positive instances, minimizing false negatives.
F1 Score	Represents the mean of precision and recall, ranging from 0 to 1. A higher F1 score indicates better model performance.
Area Under the ROC Curve (AUC-ROC)	Evaluates binary classification performance by calculating the area beneath the receiver operating characteristic (ROC) curve, which plots the ratio of the true positive rate to the false positive rate. The AUC-ROC value ranges from 0 to 1, and a higher value suggests better model performance.
Log Loss	Also referred to as cross-entropy loss, it gauges the disparity between the predicted possibilities and the true labels. A lower log loss value signifies a more accurate model.

4.5.3. Evaluation Metrics

When assessing the performance of machine-learning models, the utilization of various metrics is recommended, with the selection depending primarily on the particular objectives and prerequisites of the problem under research. In the context of this research, the following metrics have been specifically adapted to gauge the overall accuracy of the developed predictive model.

By utilizing these metrics, the research aims to evaluate the constructed predictive model's overall accuracy comprehensively. It is imperative to consider multiple metrics concurrently to understand the model's performance in greater lengths.

Table 2. Popular hyper-parameter tuning methods

Methods	Description
Grid Search	A grid of possible values for each hyper-parameter is defined, and all possible combinations are searched through. Despite being simple and straightforward, its computational complexity is exceptionally costly for larger hyper-parameter spaces.
Random Search	Randomly selects combinations of hyper-parameters from predefined ranges. It offers better scalability and finds suitable hyper-parameter configurations with fewer iterations than Grid Search.
Bayesian Optimization	Utilizes probabilistic models to tailor the relationship between hyper-parameters and performance. It iteratively selects hyper-parameters based on previous evaluations and focuses on promising regions of the search space.
Gradient-based Optimization	One notable gradient-based optimization is the Tree-structured Parzen Estimator (TPE), which constructs a model better than another and uses that information to guide the search process.

For this research, Bayesian Optimization is selected as the preferred method for hyper-parameter tuning in XGBoost. This choice stems from the method's ease of implementation and ability to iterate even when numerous iterations are required effectively. Furthermore, its capacity to intelligently explore the hyper-parameter space and utilize probabilistic models to guide the search process toward promising regions. By taking advantage of these benefits, Bayesian Optimization offers a reliable and efficient means of identifying the most optimal hyper-parameter configurations for XGBoost in this research.

5. IMPLEMENTATION AND RESULTS

5.1. Implementation

5.1.1. Training procedure

The dataset containing the academic results of the School of Computer Science and Engineering's graduates of the past five years, as provided by the Office of Academic Affairs, required certain modifications to make it suitable for utilization within the predictive model. The original dataset's structure presents a challenge, as the information of a single graduate is distributed across multiple rows, as depicted in Figure 2. Consequently, it is unsuitable for direct input into the XGBoost algorithm. In this study, the dataset is initially reorganized using Microsoft Excel (or Google Spreadsheets), as the operations involved are relatively straightforward and simpler compared to coding techniques.

This approach transforms individual courses into distinct columns (or features), ensuring that each graduate's data occupies a single row. Within this newly reorganized structure, the scores for each course are allocated to their respective positions. As seen in Figure 2, each course is associated with four distinct score types: in-class, midterm, final, and Grade Point Average (GPA). For this study, only the GPA scores are considered for the graduates' performance in each course. By excluding the other three score types, the resulting dataset will become less complex, reducing the number of features that might impact the accuracy of the trained model. The new data sheet is also sorted on the graduates' ID column, so that the order of these graduates will appear from oldest to newest. Following the reorganization mentioned above, the newly compiled data sheet becomes reasonably suitable for developing the machine-learning model, as depicted in Figure 3.

	A	B	C	D	E	F	G	H	I	J	K	L
1	MaSV	NHHK	MaMH	TenMH	SoTinChi	B1	K1	T1	DiemHP	DiemChuHP	SoTCTL	DTBTK
2	[redacted]01	20141	EN074IU	Reading & Writing IE2	8	65	54	57.0	59	C	143	76.9
3	[redacted]01	20141	EN075IU	Listening & Speaking IE2	8	68	35	38.0	46	D+	143	76.9
4	[redacted]01	20141	MA001IU	Calculus 1	4	100	90	98.0	97	A+	143	76.9
5	[redacted]01	20141	PT001IU	Physical Training 1	3		95	80.0	83	A	143	76.9
6	[redacted]01	20142	CH011IU	Chemistry for Engineers	3	74	39	75.0	64	B	143	76.9
7	[redacted]01	20142	CH012IU	Chemistry Laboratory	1	72		20.0	56	C	143	76.9
8	[redacted]01	20142	EN007IU	Writing AE1	2	73	60	40.0	56	C	143	76.9
9	[redacted]01	20142	EN008IU	Listening AE1	2	91	47	45.0	55	C	143	76.9
10	[redacted]01	20142	EN011IU	Writing AE2	2	71	47	63.0	61	B	143	76.9
11	[redacted]01	20142	EN012IU	Speaking AE2	2	68	76	71.0	72	B+	143	76.9
12	[redacted]01	20142	IT064IU	Introduction to Computing	3	93	57	63.0	70	B+	143	76.9
13	[redacted]01	20142	IT116IU	C/C++ Programming	4	96	73	72.0	80	A	143	76.9
14	[redacted]01	20142	MA003IU	Calculus 2	4	92	72	91.0	87	A	143	76.9
15	[redacted]01	20142	PE008IU	Critical Thinking	3	90	78	72.0	77	B+	143	76.9
16	[redacted]01	20142	PT002IU	Physical Training 2	3		50	85.0	78	B+	143	76.9
17	[redacted]01	20151	EE053IU	Digital Logic Design	3	100	58	97.0	86	A	143	76.9
18	[redacted]01	20151	EE054IU	Digital Logic Design Laboratory	1	92		65.0	84	A	143	76.9
19	[redacted]01	20151	IT069IU	Object-Oriented Programming	4	90	70	90.0	85	A	143	76.9
20	[redacted]01	20151	IT131IU	Theoretical Models in Computing	4	75	70	77.0	74	B+	143	76.9
21	[redacted]01	20151	PE013IU	Revolutionary Lines of Vietnamese Communist Party	3	50	50	55.0	53	C	143	76.9
22	[redacted]01	20151	PH014IU	Physics 2	2	75	95	100.0	91	A+	143	76.9
23	[redacted]01	20151	PH015IU	Physics 3	3	70	45	65.0	61	B	143	76.9
24	[redacted]01	20152	IT079IU	Principles of Database Management	4	100	69	67.0	78	B+	143	76.9
25	[redacted]01	20152	IT089IU	Computer Architecture	4	90	89	84.0	87	A	143	76.9
26	[redacted]01	20152	IT090IU	Object-Oriented Analysis and Design	4	60	72	74.0	69	B	143	76.9
27	[redacted]01	20152	MA020IU	Discrete Mathematics	3	70	86	71.0	75	B+	143	76.9
28	[redacted]01	20152	MA023IU	Calculus 3	4	90	77	100.0	93	A+	143	76.9
29	[redacted]01	20152	PE011IU	Principles of Marxism	5	100	80	80.0	84	A	143	76.9
30	[redacted]01	20152	PE012IU	Ho Chi Minh's Thoughts	2	75	75	75.0	75	B+	143	76.9
31	[redacted]01	20152	PH012IU	Physics 4	2	70	85	70.0	75	B+	143	76.9
32	[redacted]01	20161	IT076IU	Software Engineering	4	78	86	68.0	76	B+	143	76.9
33	[redacted]01	20161	IT091IU	Computer Networks	4	73	56	80.0	71	B+	143	76.9
34	[redacted]01	20161	IT093IU	Web Application Development	4	59	64	53.0	58	C	143	76.9

Figure 2. Original dataset's format

	A	B	C	D	E	DC	DD	DE	DF	DG
1	MaSV	Reading & Writing IE2	Listening & Speaking IE2	Calculus 1	Physical Training 1	Listening & Speaking IE2 (for twinning program)	History of Vietnamese Communist Party	SoTCTL	DTBTK	Career Choice
2	[redacted]01	59	46	97	83	#N/A	#N/A	143	76.9	Further Study
3	[redacted]04	#N/A	#N/A	84	62	#N/A	#N/A	143	76.2	Further Study
4	[redacted]06	74	66	64	87	#N/A	#N/A	143	63.8	Work
5	[redacted]20	62	56	80	65	#N/A	#N/A	143	67.8	Work
6	[redacted]21	63	55	82	87	#N/A	#N/A	143	68.2	Work
7	[redacted]24	#N/A	#N/A	61	77	#N/A	#N/A	143	66.9	Work
8	[redacted]27	75	61	61	86	#N/A	#N/A	143	72.6	Work
9	[redacted]28	76	63	90	75	#N/A	#N/A	143	76.5	Further Study
10	[redacted]31	#N/A	#N/A	80	81	#N/A	#N/A	143	87.9	Further Study
11	[redacted]33	55	37	61	73	#N/A	#N/A	143	68.5	Work
171	[redacted]25	58	67	78	96	#N/A	#N/A	143	82.1	Further Study
172	[redacted]32	#N/A	#N/A	91	94	#N/A	#N/A	143	86.6	Further Study
173	[redacted]58	60	66	79	67	#N/A	#N/A	143	79.9	Further Study
174	[redacted]86	68	64	84	90	#N/A	#N/A	143	78.1	Further Study
175	[redacted]27	54	57	52	98	#N/A	#N/A	143	69.1	Work
176	[redacted]55	52	62	76	97	#N/A	#N/A	143	71.2	Work
177	[redacted]69	#N/A	#N/A	63	84	#N/A	#N/A	143	76.0	Further Study
178	[redacted]85	56	65	77	91	#N/A	#N/A	143	82.2	Further Study

Figure 3. Newly reorganized dataset with the sheet name of 'Marks'

	Reading & Writing IE1	Listening & Speaking IE1	Reading & Writing IE2	Listening & Speaking IE2	Calculus 1	Physical Training 1	Chemistry for Engineers	Chemistry Laboratory	Writing AE1	Listening AE1	...	Digital Electronic 1	Digital Electronic 1 Laboratory	Light, Ware & Electron	Skills for Communicating Information	Functional Programming	Electricity & Magnetism	Electricity & Magnetism Laboratory	SoTCTL	DTBTK	Career Choice
0	NaN	NaN	59.0	46.0	97.0	83	64	56	56	55	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	143	76.9	Further Study
1	NaN	NaN	NaN	NaN	84.0	62	61	84	58	76	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	143	76.2	Further Study
2	NaN	NaN	74.0	66.0	64.0	87	60	54	59	74	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	143	63.8	Work
3	81.0	74.0	62.0	56.0	80.0	65	62	67	66	54	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	143	67.8	Work
4	83.0	77.0	75.0	61.0	61.0	86	56	75	73	51	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	143	72.6	Work

5 rows × 83 columns

Figure 4. Overview of the 'Shortened' dataset after loaded into Google Colaboratory

	IEs	IEs_Credits	Generals	Generals_Credits	Cores	Cores_Credits	Electives	Electives_Credits
0	Reading & Writing IE1	8.0	Calculus 1	4	Introduction to Computing	3.0	Deep Learning	4.0
1	Listening & Speaking IE1	8.0	Calculus 2	4	C/C++ Programming	4.0	Digital Image Processing	4.0
2	Reading & Writing IE2	8.0	Calculus 3	4	Theoretical Models in Computing	4.0	Software Architecture	4.0
3	Listening & Speaking IE2	8.0	Differential Equations	4	Digital Logic Design	3.0	Internet of Things	4.0
4	NaN	NaN	Physics 1	2	Digital Logic Design Laboratory	1.0	Net-Centric Programming	4.0
5	NaN	NaN	Physics 2	2	Object-Oriented Programming	4.0	Information System Management	4.0
6	NaN	NaN	Physics 3	3	Discrete Mathematics	3.0	Computer Graphics	4.0
7	NaN	NaN	Physics 3 Laboratory	1	Principles of Database Management	4.0	IT Project Management	4.0
8	NaN	NaN	Physics 4	2	Computer Architecture	4.0	Introduction to Data Mining	4.0
9	NaN	NaN	Writing AE1	2	Algorithms & Data Structures	4.0	Mobile Application Development	4.0
10	NaN	NaN	Listening AE1	2	Object-Oriented Analysis and Design	4.0	NaN	NaN
11	NaN	NaN	Writing AE2	2	Computer Networks	4.0	NaN	NaN
12	NaN	NaN	Speaking AE2	2	Operating Systems	4.0	NaN	NaN
13	NaN	NaN	Physical Training 1	3	Software Engineering	4.0	NaN	NaN
14	NaN	NaN	Physical Training 2	3	Principles of Programming Languages	4.0	NaN	NaN
15	NaN	NaN	Chemistry for Engineers	3	Web Application Development	4.0	NaN	NaN
16	NaN	NaN	Chemistry Laboratory	1	Introduction to Artificial Intelligence	4.0	NaN	NaN
17	NaN	NaN	Critical Thinking	3	Internship	3.0	NaN	NaN
18	NaN	NaN	Principles of Marxism	5	Entrepreneurship	3.0	NaN	NaN
19	NaN	NaN	Revolutionary Lines of Vietnamese Communist Party	3	Special Study of the Field	2.0	NaN	NaN
20	NaN	NaN	Ho Chi Minh's Thoughts	2	Thesis	10.0	NaN	NaN
21	NaN	NaN	Probability, Statistic & Random Process	3	NaN	NaN	NaN	NaN

Figure 5. Computer Science's curriculum (updated 2021)

Upon careful examination of both the original and the reorganized datasets, it has come to attention that two occurrences of the grade are designated with "WH" (Withheld). Such "WH" may arise due to two possible scenarios: either the respective graduates did not attend the registered classes, or their academic misconduct resulted in the withholding of their grades. In any case, assigning a zero value to these entries rather than utilizing the median of the features aligns more meaningfully with the circumstances surrounding the withheld grades. By assigning these instances with zeros, they will be classified with the type of integer instead of objects by the codes, which avoids involving an encoder to transform them into numeric values later on. It is noteworthy that there are certain instances where multiple courses overlap with each other, indicating that they may have similar studying outcomes, but with different course names and originating from different faculties. Therefore, it is appropriate to consolidate these overlapping courses into a unified course, bearing a new name. Upon closer inspection of the 'Marks' dataset, certain graduates took courses that are not registered by the majority, while simultaneously abstaining from enrolling in courses that are taken by the majority. Based on this observation, it can be inferred that these students likely major in either Network Engineering (NE) or

Computer Engineering (CE), as their course selections align with the prescribed curricula of these respective majors. Consequently, it has been decided that the data pertaining to these particular graduates shall be excluded from the dataset, meaning that only Computer Science (CS) curriculum is used for this study. This step is taken

with the aim of reducing the number of features with missing values. After removing the graduates above, the overall count of individuals in the dataset is reduced by 19, resulting in 158 graduates compared to the initial count of 177. The modified dataset, incorporating all the alterations described above, is denoted as the 'Shortened' dataset. The data preprocessing phase using Microsoft Excel (or Google Spreadsheets) is now concluded, as the dataset has undergone necessary modifications to render it suitable for subsequent preprocessing using Python codes.

5.1.2. Data Preprocessing using Google Colaboratory and Python 3

The 'Shortened' dataset is first imported to initiate the preprocessing procedures in Google Colaboratory. As the graduates' ID column does not impact the resultant predictive model in any way, it is excluded from the dataset. Figure 5 presents the first five instances of the loaded dataset, offering a glimpse into its structure and content. Checking the dataset of the number of unique rows within each feature and the number of null values in each of them provides a comprehensive overview of the features with more missing values than the others. Features with a single unique value show a predominance of missing values, potentially with no other distinct values except for the unique entry. Conversely, features without null values can be disregarded during the subsequent dataset-cleaning stage. To ensure that the dataset can preprocess properly, its courses are categorized into four distinct subsets: Intensive English (IE), general, core, and elective. The cumulative credits, overall GPA, and career choice (target label) are separated into a different set. Each of these subsets will be processed with the goal of either imputing in or eliminating all missing values.

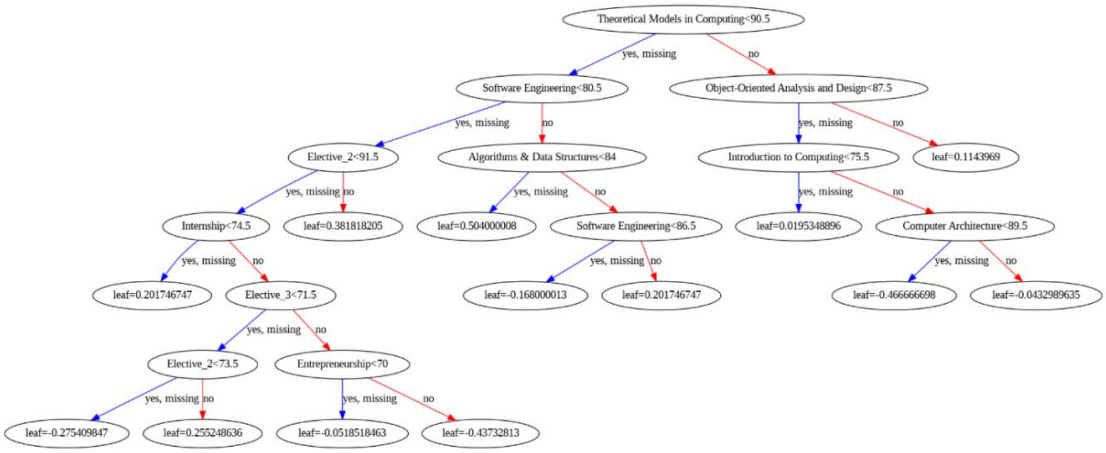

Figure 6. XGBoost Classifier tree model, with default settings and early_stopping_round = 20

5.1.3. Feature selection/extraction

The processed dataset is partitioned into two distinct subsets. The first subset encompasses the target variable, representing the graduates' career choices, while the second subset contains the grades alongside the cumulative credits and the overall GPA. As the target variable assumes a categorical data type, it necessitates encoding. In this study, the career choice only variates between 'Further Study' and 'Work'; therefore, binary encoding using 0s and 1s will represent these categories, where 0s represent 'Further Study' and 1s represent 'Work'. Subsequently, the two subsets are further used to generate a training set, and a testing set, with the ratio of 80 percent to 20 percent.

5.1.4. XGBoost Model Construction

The initial stage of developing the predictive model involves assessing the balance of the target variable. The sample weight is calculated and utilized to balance out such disparities to ensure that the predictive model can still be trained correctly in case the target variable distribution exhibits skewness or imbalance.

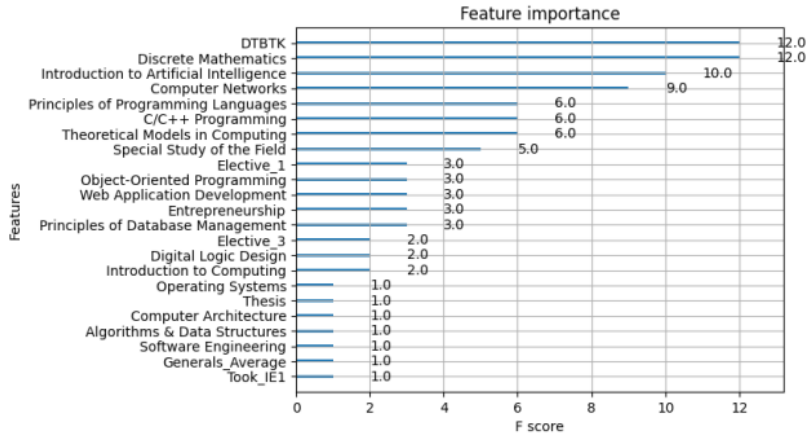

Figure 7. Feature importance plot generated by XGBoost

The calculation for determining the sample weight is employed to achieve this balance adjustment. The resulting sample weight value will serve as a parameter during the model fitting process. XGBoost supports the 'predict' and 'predict_proba' methods, which will be utilized for the evaluation phase of this study. A tree model generated from a run using the default settings is illustrated in Figure 6 and the feature importance analyzed by XGBoost is shown in Figure 7.

5.1.5. Hyperparameter tuning

To enhance the precision of a predictive model, the optimal approach involves employing optimization techniques to fine-tune its hyperparameters. The adjustment of various types of hyperparameters is entirely dependent on the algorithm employed. Specifically, XGBoost provides a wide range of hyperparameters from which to select. Nevertheless, only a few of these hyperparameters are commonly tuned due to their significant influence on both the model's performance and predictive accuracy. This study focuses on tuning seven basic hyperparameters of XGBoost, as detailed in Table 3.

Table 3. Hyper-parameters in the tuning process

Name	Alias	Description
eta	learning_rate	Ranges within the interval [0, 1], and the typical value is between 0.01 to 0.3. Smaller values mean slower convergence but potentially more accurate models. Larger values may lead to faster convergence but can give less accurate models.
max_depth		Ranges within the interval [0, ∞], and the typical value is between 3 and 10. Smaller values control the trees' complexity and prevent overfitting. Larger values allow the model to capture more complex patterns in the data.
min_child_weight		Ranges within the interval [0, ∞], and the typical value is between 1 and 20. Larger values mean more conservative tree growth.
subsample		Ranges within the interval (0, 1), and the typical value is between 0.5 and 1.0. Smaller values mean more randomness and prevent overfitting. At 1.0, the entire dataset is used.
colsample_bytree		The typical value is between 0.5 and 1.0. Determines the fraction of features to be randomly sampled for each tree. Higher values mean more diversity.
gamma	min_split_loss	Ranges within the interval [0, ∞], and the typical value is between 0 and 5. Higher values make the tree-building process more conservative.
lambda	reg_lambda	The typical value is between 0 and 1. Higher values mean more robust regularization and gain better control over the model's complexity and prevent overfitting.

5.2. Results

The results obtained from the prediction performed on the testing subset appear to vary due to the randomness associated with the data division into training and testing sets during each execution of the code. This variation is entirely expected. The accuracy of the predictive model ranged from a minimum of 78.125 percent to a maximum of 90.625 percent, with an average accuracy of around 87.5 percent.

To provide further insights into the predictive model's performance, the accuracy values obtained from five different runs of the code, both for the predictive model using default settings and the model with optimal hyper-parameters, are presented in Table 4 below. Additionally, a testing phase was conducted on the training set to assess over-fitting in the model.

Table 4. Prediction accuracy of five different runs

RUN 1						
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC-ROC**	**Log Loss**
Default	81.250%	86.364%	86.364%	0.864	0.936	0.368
Tuned	87.500%	95%	86.364%	0.905	0.916	0.413
Overfit?	90.476%	100%	84%	0.913	0.977	0.378
RUN 2						
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC-ROC**	**Log Loss**
Default	78.125%	82.609%	86.364%	0.844	0.902	0.380
Tuned	90.625%	95.238%	90.909%	0.930	0.959	0.366
Overfit?	90.476%	94.366%	89.333%	0.918	0.970	0.365
RUN 3						
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC-ROC**	**Log Loss**
Default	84.375%	90.476%	86.364%	0.884	0.955	0.310
Tuned	84.375%	86.957%	90.909%	0.889	0.920	0.441
Overfit?	89.683%	91.892%	90.667%	0.913	0.944	0.431
RUN 4						
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC-ROC**	**Log Loss**
Default	78.125%	82.609%	86.364%	0.844	0.902	0.387
Tuned	81.250%	86.364%	86.364%	0.864	0.927	0.422
Overfit?	89.683%	95.588%	86.667%	0.909	0.960	0.409
RUN 5						
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC-ROC**	**Log Loss**
Default	84.375 %	90.476%	86.364%	0.884	0.950	0.359
Tuned	87.500%	95%	86.364%	0.905	0.898	0.399
Overfit?	90.476%	100%	84%	0.913	0.961	0.367

Figure 8 displays the performance of the predictive model through 15 distinct runs, with the blue line depicting the default model's performance and the orange line depicting that of the tuned model. The lowest accuracy of the default model is around 78.125%, while in certain scenarios, the model performs with an accuracy of 84.375% at best. This constitutes the default model's average accuracy of approximately 87.5%. In addition, as seen from the illustration, the tuned model outperforms the default model in almost all the cases, which is entirely to be expected. In one of the cases, the tuned model can enhance the accuracy of its predecessor, boosting the accuracy of the default model by 12.5%, from 78.125% to 90.625%.

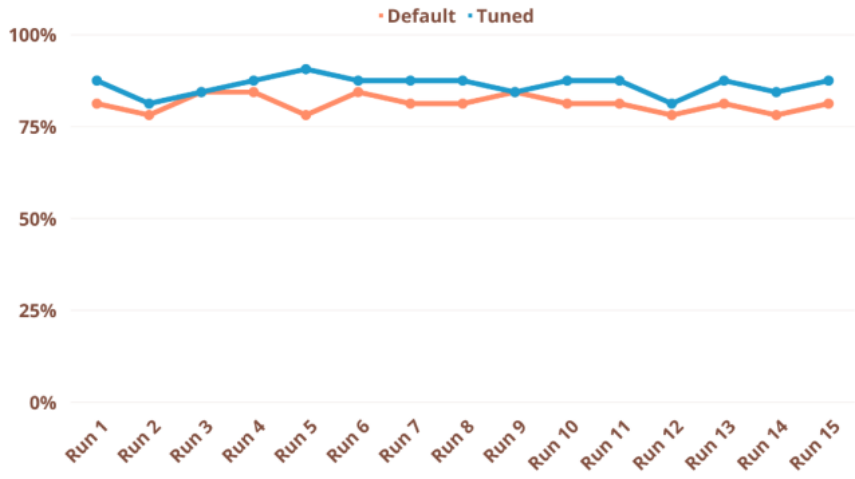

Figure 8. The predictive model's performance over 15 different runs

Table 5 details the accuracy of both the XGBoost model presented in this paper and the decision-tree based models from other studies. It is clear that the developed model yields better results compared to most of the other models. However, it is still not on par with the model suggested by Jamal et al. (2020), which utilized the C5.0 and CHAID algorithms. This is evident by the fact that this paper's model performs at an accuracy lower by over 5%.

Table 5. Comparison between current XGBoost and past studies' decision tree models

Standard Decision Tree Models	Accuracy
Saa (2016)	40%
Nie et al., (2020)	53%
Casuat et al. (2019)	85%
Current XGBoost model	*87.5%*

Table 6 shows the accuracy of the XGBoost model presented in this paper and the XGBoost models from two different studies. The proposed model outperforms the model by Nie et al. (2020) by nearly 28. This solidifies the capability of the proposed model in making predictions of similar problems and further signifies the predictive power of the XGBoost algorithm.

Table 6. Comparison between current XGBoost and past studies' XGBoost models

XGBoost Decision Tree Models	Accuracy
Nie et al., (2020)	60%
Current XGBoost model	*87.5%*

6. CONCLUSION

This paper illustrates the utilization of an academic results dataset to research and employs the XGBoost machine-learning algorithm for predicting students' future career paths. Notably, the integration of XGBoost is facilitated by its inherent support for L1 and L2 standardization, thereby rendering the data normalization and standardization unnecessary. Consequently, the most difficult phase of the research pipeline resides in correctly preprocessing the dataset. In addition, XGBoost provides a plethora of hyperparameters, enabling fine-tuning of the predictive model to better align with the specific characteristics of the given dataset. With a sufficient amount of data, the proposed model may effectively adapt to changing circumstances, thereby enhancing its predictive capability. For this study, the classification task employed by XGBoost entails binary classification, as the dataset's target label comprises solely two distinct choices.

The current accuracy of the model remains notably low in comparison to various other approaches, highlighting the criticality of acquiring a richer and more expansive dataset in the near future. Additionally, it is better to test the model using multiple other datasets as opposed to relying solely on a single dataset for evaluation. It may also be possible for the entire process, from handling the dataset to training the model, to be managed by a pipeline to avoid the need for manual execution of each step. It is noteworthy that the model's lack of interoperability with other systems poses a significant limitation in the field of machine learning. That is why it has become essential to address this constraint to enhance the model's applicability and utility.

ACKNOWLEDGMENT

We gratefully acknowledge AIoT Lab VN for their invaluable support in this research.

REFERENCES

Alabadla, M., Sidi, F., Ishak, I., Ibrahim, H., Affendey, L. S., Ani, Z. C., ... & Jaya, M. I. M. (2022). Systematic review of using machine learning in imputing missing values. *IEEE Access*, *10*, 44483-44502.

Bao, J. (2020, August). Multi-features based arrhythmia diagnosis algorithm using xgboost. In *2020 International Conference on Computing and Data Science (CDS)* (pp. 454-457). IEEE.

Budholiya, K., Shrivastava, S. K., & Sharma, V. (2022). An optimized XGBoost based diagnostic system for effective prediction of heart disease. *Journal of King Saud University-Computer and Information Sciences*, *34*(7), 4514-4523.

Casuat, C. D., & Festijo, E. D. (2019, December). Predicting students' employability using machine learning approach. In *2019 IEEE 6th international conference on engineering technologies and applied sciences* (ICETAS) (pp. 1-5). IEEE.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, *2*(1), 20-28.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 785-794).

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., & Tabona, O. (2021). A survey on missing data in machine learning. *Journal of Big Data*, *8*(1), 1-37.

Gumelar, A. B., Setyorini, H., Adi, D. P., Nilowardono, S., Widodo, A., Wibowo, A. T., ... & Christine, E. (2020, September). Boosting the Accuracy of Stock Market Prediction using XGBoost and Long Short-Term Memory. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)* (pp. 609-613). IEEE.

Jamal, K., Kurniawan, R., Husti, I., Nazri, M. Z. A., & Arifin, J. (2020, October). Predicting Career Decisions Among Graduates of Tafseer and Hadith. In *2020 2nd International Conference on Computer and Information Sciences* (ICCIS) (pp. 1-4). IEEE.

Križanić, S. (2020). Educational data mining using cluster analysis and decision tree technique: A case study. *International Journal of Engineering Business Management*, *12*, 1847979020908675.

Loh, W. Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14-23.

Nie, M., Xiong, Z., Zhong, R., Deng, W., & Yang, G. (2020). Career choice prediction based on campus big data—mining the potential behavior of college students. *Applied Sciences*, *10*(8), 2841.

Panhalkar, A. R., & Doye, D. D. (2022). Optimization of decision trees using modified African buffalo algorithm. *Journal of King Saud University-Computer and Information Sciences*, *34*(8), 4763-4772.

Poznanski, K. Z. (2015). Confucian economics: the world at work. *World Review of Political Economy*, *6*(2), 208-251.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81-106.

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Roy, K. S., Roopkanth, K., Teja, V. U., Bhavana, V., & Priyanka, J. (2018). Student career prediction using advanced machine learning techniques. International *Journal of Engineering & Technology*, *7*(2.20), 26-29.

Saa, A. A. (2016). Educational data mining & students' performance prediction. *International Journal of Advanced Computer Science and Applications*, *7*(5).

Vignesh, A., Yokesh Selvan, T., Gopala Krishnan, G. K., Sasikumar, A. N., & Ambeth Kumar, V. D. (2020). Efficient student profession prediction using XGBoost algorithm. In *Emerging Trends in Computing and Expert Technology* (pp. 140-148). Springer International Publishing.

Wang, T., Zhang, Y., & Jia, R. (2021, May). Improving robustness to model inversion attacks via mutual information regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 13, pp. 11666-11673).

Wang, Y., & Guo, Y. (2020). Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost. *China Communications*, *17*(3), 205-221.