JOIV : Int. J. Inform. Visualization, 9(1) - January 2025 231-240

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Optimizing iCadet Assignment through User Profiling

Peak-Fei Yap [a,*], Choo-Yee Ting [a], Hairul A. Abdul-Rashid [b]

[a] *Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, Cyberjaya, Cyberjaya, Selangor, Malaysia*
[b] *Faculty of Engineering, Multimedia University, Persiaran Multimedia, Cyberjaya, Cyberjaya, Selangor, Malaysia*
*Corresponding author: *yappeakfei@gmail.com*

Abstract*—Industry Cadetship program is a program that assigns penultimate year students to companies matching their profiles, bridging academic learning and industry skills. Manual data analysis for assignments is time-intensive, prompting this study's objectives: (i) propose an algorithm to optimize student-company assignment by using the student and company profiles, (ii) propose a method for the assignment of lecturers to company, and (iii) use similarity measure techniques to recommend companies with similar characteristics. Data was collected from a university student, company, and lecturer datasets. To assign students to companies, the Haversine, OpenStreetMap, and NetworkX were used to calculate the shortest geographical distance between the students and the companies; evaluated based on mean, variance, standard deviation, and utilization rate. For the lecturer assignment, cosine similarity was applied to measure the similarity between domain descriptions and company or lecturer information after performing Voyage AI embeddings. Lecturers are assigned to companies based on the highest domain similarity scores. The performance was evaluated using accuracy, precision, recall, and F1- score. Findings showed that embedding techniques significantly enhanced the matching process, with accuracy improved from 0.464 to 0.6071, precision increased from 0.417 to 0.5058, recall saw an equal rise from 0.464 to 0.6071, and the F1-score advanced from 0.417 to 0.5264. Longer descriptive inputs further improved performance, with accuracy rising from 0.6154 to 0.7692, precision from 0.5744 to 0.7751, recall remaining steady at 0.7692, and F1-score increasing from 0.5807 to 0.7484. This work can be extended to explore job portal dataset by aligning profiles with geography and specialization.

***Keywords*— iCadet; user profile; company profile; similarity measure; matching algorithm.**

Manuscript received 14 Jan. 2024; revised 19 Aug. 2024; accepted 27 Nov. 2024. Date of publication 31 Jan. 2025.

I. INTRODUCTION

The Industry Cadetship (iCadet) program is a new initiative with the aim to help undergraduate students of all faculties [1]. This program helps narrow the gap between what students learn in school and what is required in the workplace. Those involved in the iCadet program will engage in a variety of activities, including industry visits, corporate social responsibility (CSR) initiatives, onboarding events, and company gatherings, to immerse themselves in the corporate culture [1].

Historically, matching students to companies was a lengthy process that involved evaluating many factors like student skills and job requirements. It often relied on human input, which could be subjective and inefficient [2]. Similar to the job matching problem, iCadet placement was formerly a difficult and time-consuming process that required taking into account several variables, including the qualifications of the student, the demands of jobs, the skills, and many more. To make the process faster and more effective, we used Artificial Intelligence (AI).

Workplace location has been identified as a significant yet unexplored factor in job matching, according to recent findings [3]. Moreover, geographical factors are crucial in the students' ability to find jobs, as some regions have a higher demand for specific professions or industries, as discussed by [4]. Firstly, there was a critical need for an efficient algorithmic framework to facilitate the assignment of students and supervisors to companies. Workplace location and geographical factors are significant yet underexplored aspects influencing students' ability to find jobs, especially in regions with higher demand for specific professions [3], [4]. Addressing this, the primary goal of this study was to design and implement a system leveraging student profiles, company profiles, and lecturers' profiles to ensure optimal matches between students, lecturers, and companies.

Another critical aspect is to assign appropriate lecturers to supervise iCadets placements, matching their expertise with the needs of the companies' hosting students. This approach

ensured that students received guidance and mentorship, and enhancing their educational experience [5].

Furthermore, the current placement system frequently falls short of offering customized advice based on each student's distinct profile and goals. Students found it difficult to locate and establish connections with businesses that closely matched their interests and career goals in the absence of this customized approach. To solve this problem, this study focused on using similarity measure techniques.

The objectives of this project are as follows:

a. To propose an algorithm for student-company assignments through student profiles and company profiles.
b. To propose a method for the assignment of the supervisor to a company.
c. To use similarity measure techniques to recommend companies with similar characteristics.

Task assignment problems involve allocating a set of tasks to a set of agents in a way that optimizes one or more objectives, such as minimizing total cost, maximizing efficiency, or achieving a fair distribution of work. Assigning workers their interested tasks is critical to ensuring continuous worker performance. If workers are assigned uninterested tasks, they may complete them with poor quality or even sabotage them, negatively impacting businesses. Consideration of worker preferences is thus a significant challenge [6].

According to [7], existing works shows that many studies on gender look at where workers are and what they to do separately. Not understanding how these two things connect can lead to bad job assignments. To fix this, a new method combines location and preferences to assign tasks. It aims to pick workers who are nearby and willing to do the job. A new approach called MAJA helps to get the most tasks done while following certain rules.

The researchers then mentioned that ways to avoid task starvation with low gain should be considered in future work. LSTM-based model for extracting workers' latent feelings from historical data was proposed by [8]. The researchers then developed an efficient greedy algorithm and a Kuhn-Munkras (KM)-based algorithm to achieve optimal task assignment, taking into consideration the workers' feelings.

This research highlighted that graduate employability extends beyond academic credentials to include unique market contributions. Moreover, the study revealed an early gender wage gap, where female interns earned more than non-interns but less than their male counterparts. Notably, the significant salary disparities between genders were attributed solely to the age of the respondents, with older individuals favoring male candidates [9].

Odlin et al. [10] found that internships located far from the home institution for in inherently riskier settings, such as factories or politically unstable regions, posed increased risks. Suck locations, crucial for specific fields like engineering or hospitality, often involve higher costs and potential isolation for students, reducing oversight and elevating risk.

The findings of [11] showed internship experiences help students understand the work world better. Before these internships, many didn't really know what to expect. According to [12], employers identified graduates' poor performance in the workplace as a result of their inability to apply classroom theories in practice.

The Students Industrial Work Experience Scheme (SIWES) helps students link what they learn in school with real jobs. It gives them a chance to work with real machines and tools. This hands-on experience helps students learn in ways that they can't in the classroom [13].

A. User Features in Student-Company Matching

According to the table below, the majority of the researchers used gender as a factor in internship placement. Gender distribution in internship placements can reveal patterns that can help guide diversity strategies in traditionally male or female-dominated fields.

Next, age is another factor that researchers often consider. It can indicate how much experience someone might have. Some internship programs are set up for students in certain years, like those in their second-to-last year. Analyzing age distribution helps ensure that internship opportunities are appropriate for the target audience.

TABLE I
USER FEATURES IN STUDENT-COMPANY MATCHING

Author	Gender	Age	Major / Academic Program	Race	Work Experience	Marital Status	Education Level	SSE	Soft Skills	Year of graduation
[9]	√	√			√		√			
[12]					√					
[15]	√	√	√	√						
[16]	√	√						√		
[17]	√	√	√	√						
[18]					√					
[19]	√	√	√						√	
[20]	√	√				√	√			
[21]	√	√		√			√			
[22]	√	√	√		√					√
[23]					√					
[24]	√	√	√							
[25]			√						√	
Total count	9	9	8	3	5	1	3	1	2	1

Furthermore, majors reflect a student's likely skill set and area of expertise. Internship placements that are relevant to a student's major allow them to apply and improve the skills learned during their academic coursework, resulting in a more meaningful and productive internship experience. As a result, a few researchers use major/academic programs to assess the alignment between a student's academic background and internship requirements.

B. User Profiling in Job Matching

Previously, finding the right job mostly relied on people making decisions, with little help from technology. Digital platforms introduced a basic system that used keywords for

job matching. But these systems often missed the finer details of job descriptions and candidate backgrounds. They also struggled with the fast-changing job market where new skills and roles pop up all the time. That's where machine learning and AI come in. They can update job profiles in real-time, making sure matches are accurate and relevant.

Previous research suggested that user profiling is vital for helping candidates find appropriate jobs. By understanding user backgrounds, job systems can give more tailored recommendations [25]. The main goal is to investigate what job candidates prefer based on their past job interviews and applications. Also, there is a growing interest in using natural language processing (NLP) to improve accuracy. Document ranking and comparing document similarities have been identified as major tasks in natural language processing (NLP) [26]. Additionally, NLP is used to extract user profiles, such as skills, education, and experience from unstructured resumes, which results in a summary of each application[27].

User profiling is critical to addressing the challenges that candidates face when navigating the complex landscape of job recruitment. By thoroughly analyzing individual user profiles, job recommendation systems can help candidates identify and secure positions that are closely related to their field of interest and expertise. This tailored approach helps reduce the frustration and uncertainty often associated with the job search process. Candidates receive recommendations that are personalized to their skills, experience, and career aspirations [28], [29], [30], [31], [32], [33].

For job matching platforms, accessing user profiles enables systems to incorporate a wide range of data, such as age, country, past learning activities, and educational background. This information helps identify users with similar learning of professional preferences [34]. These user profiles might be the important features for the model to recommend the job.

For expert recommendation systems, BERTERS, a multimodal classification approach for expert recommendation systems have been applied to identify patterns in candidates' expertise and preference. [35]. Additionally, the skills2job recommendation system, which begins with a set of user preferences for skills and identifies the most suitable jobs as users emerge from a large dataset of Online Job Vacancies (OJVs) [35]. The researchers utilize European Skills, Competences, Qualifications, and Occupations Taxonomy (ESCO) to assess the similarity between occupations and users' skills.

Cui and colleagues discovered a gap in the existing recommendation system, which always ignored the inherent relationship between the user's preference and time. In reality, the user's interest changes over time [36], [37]. To address the gap, the researchers proposed a novel recommendation model based on the time correlation coefficient and an improved K-means with cuckoo search (CSK-means). Systematic experimental results show that their model is effective [38].

Current recommendation systems rely on past interaction history to estimate user preferences, which limits their ability to capture fine-grained and dynamic user preferences [39]. Thus, the researchers proposed Conversational Path Reasoning (CPR). It walks through the attribute vertices based on user feedback, explicitly using the user-preferred attributes. CPR reduces irrelevant candidate attributes, increasing the likelihood of identifying user-preferred attributes. The researchers discovered user preferences such as rating information, tag information, the number of users, and the number of products on the books.

Furthermore, a Conversational Recommender System (CRS) called Estimation-Action-Reflection (EAR) [37]. This system estimates user preferences for both items and item attribute and then uses learning dialogue policies to decide whether to inquire about attributes or recommend items based on ongoing conversation and user preferences. The underlying model consists of factorization machines that have been trained on user profiles and item attributes.

It was discovered that in many existing CRS, many current systems have a fixed set of user intents. This means they rely heavily on background knowledge that is built by hand [38]. In Natural Language Processing (NLP), figuring out what the user wants and picking the best response is really important.

A new way to recommend jobs uses different machine learning models and language processing techniques [33], [39]. Researchers looked at user skills and job requirements to make better suggestions. They combined features to fix the problems with older recommendation systems. The researchers found that the Random Forest classifier algorithm worked best for their main model. For their language processing needs, the Spacy Phrase Matcher did a great job.

Additionally, a Collaborative Filtering method using the K-NN algorithm. This one helped find job fits for Informatics Engineering students by checking how close their skills were to tech jobs [40].Finally, researchers also used cosine-similarity along with K-NN to match CVs with job descriptions [28].

II. Materials and Method

Fig 1 indicates the flow of methods that are applied in this project. In this project, data preprocessing and missing data handling are performed to transform the raw data in as useful for assignment algorithms. Assignment algorithms are constructed after the missing data handling have been done.

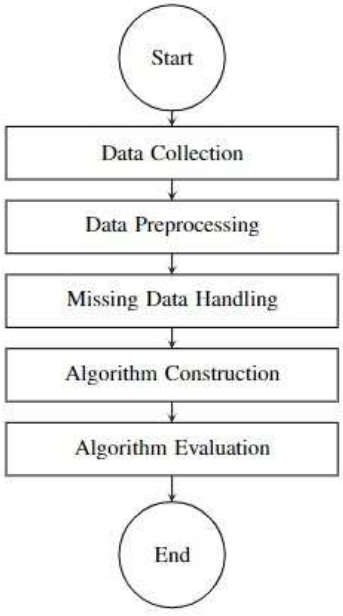

Fig. 1 Flowchart of methods

Fig 2 shows a framework overview of student-com from data collection to data preprocessing including feature engineering, and missing value handling. Next, construct algorithms by setting up preferences using the latitude, longitude, and domains. The margin of error of latitude and longitude is set to 2km. The threshold of the distance for the assignment is set to 10km, 20km, 30km, and 40km.

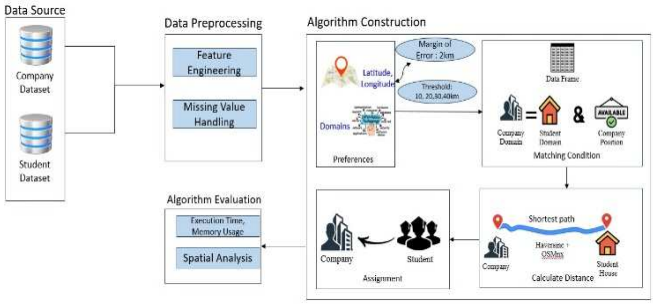

Fig. 2 Framework overview of student-company assignments.

The algorithm first checks whether the company domain is identical to the student domain, and if it is identical, then continues checking the company's available position. All the matching data will be saved into a data frame called a *Matching Companies* csv file. The algorithm continues with calculating the shortest path between the company and the student using Haversine. Then, OSMnx is applied to calculate the drivable route on the map. Finally, the student is assigned to the nearest company.

Fig 3 shows a framework overview of lecturer-company allocation where the entire process passed through preprocessing pipelines for both lecturers and companies, whereby the domain descriptions, related subjects, and company descriptions were cleaned and tokenized. Several preparatory steps have been taken such removal of stop words, stemming, and lemmatization.

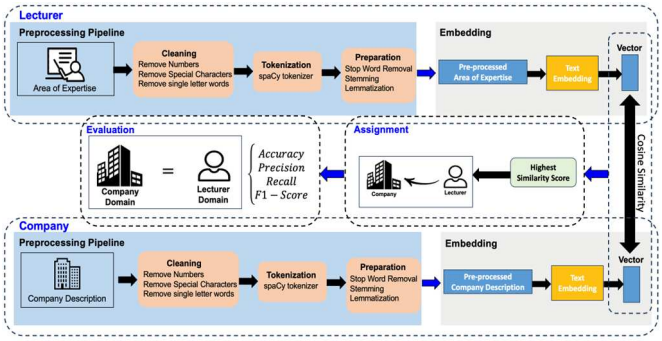

Fig. 3 Framework overview of lecturer-company assignments.

The cleaned and normalized texts were sent to the embedding stage using Voyage AI for conversion into numerical vectors. Similarity was computed between texts using cosine similarity. Hence, the most relevant domains have been assigned to both companies and lecturers. The assignment of lecturers to a company uses the domains of both companies and lecturers to perform a matching.

A. Data Source

In this work, four datasets are involved. Let *ITP* be the dataset that contains the company info that will be used for the matching for the student, *Student* will be the dataset that contains student info such as demographic data, and *Lecturer* will be the dataset of the lecturer with name, email address, expertise, and other's information.

TABLE II
FEATURES IN THE DATA COLLECTED

Dataset	Features
ITP Historical	Faculty, Company Name, Company Address, Academic Program, Major Code
Student	Student ID, Latitude, Longitude, Program Name
Lecturer	Name, Email Address, Expertise, Related Subjects, Best Domain

B. Data Extraction & Data Preprocessing

Before using assignment algorithms, the data set is preprocessed to remove duplicate rows, missing values, noisy data, and outliers. Real-world data is rarely clean or complete. Thus, data preprocessing is an important step in delivering processed data to improve assignment accuracy. *ITP* dataset consists of basic information about companies. Preprocessing steps like double backslashes, and extra spaces have been removed. Company name and address standardization also have been performed. Then, the company's major, such as Software Engineering, is converted into a specific major code, making data processing and analysis more programmatic. Furthermore, the dataset *Lecturer* includes detailed information about each supervisor. All text has been converted to lowercase and stop words and extra spaces have been removed to ensure data consistency. Additionally, special characters such as "Â Â" have been meticulously removed.

C. Missing Data Handling

For the *ITP* dataset, the missing company address was filled in by navigating Waze with Selenium and saving the company's location address. The missing longitude and latitude were filled in with a combination of Nominatim and ArcGIS techniques. This can remove the complexities of dealing with long, descriptive names that may contain special characters or spaces, which can be inconvenient in coding, database management, and reporting.

This project is mainly focused on Malaysia country Therefore, in the *Student* dataset we removed 78 rows of students who live outside of Malaysia. Next, the missing value in the *Lecturer* dataset indicated that the lecturer did not have that information. Initially, the missing value dropped. However, to retain valuable information and improve the robustness of the analysis, we replaced the missing value with "None". This ensured that no data was discarded, maintaining the dataset's completeness.

D. Algorithm Design & Construction

A preferences-matching algorithm constructed in this project aims to assign students to companies based on major alignment and proximity, considering company capacity. First, it filters the company data to find companies that match the student's major code and check for the available space of that company. Then, it calculates the distance from each company to the student by longitude and latitude using the Haversine formula, it sorts companies by distance, and the unit of distance is set as Kilometers (km). The Haversine formula is shown below:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \quad (1)$$

After sorting companies, now the algorithms are constructed to calculate the driving distance using OSMnx. The alternative way of calculating the driving distance will be Taxicab if the OSMnx cannot find the driving path. Then, the algorithm attempts to assign the student to the nearest company that matches their preferences within the distance threshold. The threshold of the distance is set to 10km, 20km, 30km, and 40km. It updates the company's available space, removes the student from the pool, and saves assignment information.

After students with matching major codes have been assigned, another function is set to assign the remaining students to any company with available space, regardless of the major code. It updates the company's space and records each assignment, just like the first function. This is to ensure that all students are assigned to a company. The calculation of distance between locations is the most important factor in determining how quickly someone can find or arrive at their destination. However, latitude and longitude coordinates facilitate the calculation of the distance between two locations on Earth.

The best way to measure how far apart two places are on Earth is by using the great-circle distance. This gives you the shortest path across the globe. A popular formula for figuring this out is the Haversine formula. It's commonly used in navigation systems. The Haversine formula helps you find the straight-line distance between two points using their longitude and latitude. The Haversine Formula is an essential equation for calculating the straight-line distance between two coordinates on Earth using longitude and latitude parameters.

OSMnx is a Python package built by geopandas, network and matplotlib to retrieve, model, analyze, and visualize street networks from OpenStreetMap. OSMnx was used for map building and visualization in their works by [41], while [42] employed OSMnx to determine the geographic node distances from real road courses in the network area.

In the iCadet assignment, the location profile was a critical consideration, necessitating the shortest possible distance between points. To address this, a novel method combining the Haversine formula and OSMnx was developed. Initially, the Haversine formula was used to filter candidates based on geographic proximity, providing a quick estimation of distances using latitude and longitude coordinates. This step aimed to reduce the time consumed in the initial screening.

Subsequently, for more accurate road-based distance calculations necessary for final assignments, OSMnx was employed. A Python function titled "Calculate driving distance" was created to determine the driving distances between companies and students. This function received two tuples containing the geographic coordinates of both entities. The algorithm first calculated a bounding box around these points by determining the minimum and maximum longitude and latitude values, incorporating a margin of error equivalent to about two kilometers. This margin ensured the inclusion of both locations and a surrounding buffer zone in the generated graph, facilitating accurate path calculation. Here is the Margin of error formula set for 2km, written in equations as follows:

$$lat_{margin} = \frac{2}{111} \quad (2)$$

$$lon_{margin} = \frac{2}{111 - \cos\left(radians(student_{lat})\right)} \quad (3)$$

Then, graph G is created within the bounding box around the two locations and is used to extract a driving network graph from OpenStreetMap data using the OSMnx library. This graph represents the network of drivable roads within the bounding box. Once the graph is obtained, the function locates the nearest nodes on this graph to the student's and company's coordinates. These nodes represent the closest points on the driving network to the specified locations. Using the NetworkX library, the function attempts to find the shortest path between the student's node and the company's node on the graph, weighted by the physical length of the roads. If a path exists, the function returns the length of this path in kilometers (as the length is initially calculated in meters).

However, if no path can be found between the two nodes—an exception raised by NetworkX as NetworkXNoPath—the function falls back to an alternate method of calculating the shortest path. Taxicabs were used as an alternative way to find the shortest path. In summary, this function is an integral part of the geographic information system (GIS) analysis providing a practical tool for measuring the accessibility of iCadet locations for students based on real-world road networks.

Algorithm 1 CalculateDrivngDistance

Input: $S_{loc}(lat_s, lon_s), C_{loc}\ (lat_c, lon_c)$
Output: D_i

Begin
  Δlat ← |Lat_s – Lat_c|
  Δlon ← |Lon_s – Lon_c|

  Lat_margin ← $\frac{2.11}{111}$
  Lon_margin ← $\frac{2.11}{111} * \cos(\frac{lat_s + lat_c}{2})$

  $x_{min} \leftarrow \min(lon_s, lon_c) - margin_{lon}$
  $x_{max} \leftarrow \max(lon_s, lon_c) + margin_{lon}$
  $y_{min} \leftarrow \min(lat, lat_C) - margin_{lat}$
  $y_{max} \leftarrow \max(lat, lat_C) + margin_{lat}$

  $node_s \leftarrow NeartestNode\ (G, lat_s, lon_s)$
  $node_c \leftarrow NeartestNode\ (G, lat_c, lon_c)$
  Try
    $path \leftarrow ShortestPath(G, node_s, node_c, weight =' length')$
  Catch
    Return none
  EndTry
  Return D_i
End

The proposed algorithm in the study assigns students to companies based on their profiles by considering the students' majors, geographical proximity to the companies, and the available spaces at the companies. The algorithm *StudentCompanyAssignment* automates this process, taking in three arguments: S_i, containing information about students; C_i, containing information about companies; and a t , a distance threshold value within which students are considered for placement.

The function begins by initializing an empty array called *assignments* to store details of the assigned students. It then

iterates over each student in the S_i, retrieving their unique ID, major code, and geographic coordinates *(s.lat, s.lon)* Using the student's major code, the function filters the C_i to find companies that match the student's major and have available capacity (c.space > 0). If no matching companies are found, the function continues to the next student.

For students with matching companies, the function calculates the distance between the student's location and each company using the Haversine formula. The resulting distances are added to the *matching_companies, M* DataFrame as a new column. The companies are then sorted based on their distance from the student in ascending order. The function iterates through the sorted list of companies and identifies those within the specified distance threshold. For each company within the threshold, the function attempts to calculate the driving distance using the previously defined *CalculateDrivingDistance* algorithm. If successful, it normalizes this distance to kilometers; if not, it sets the driving distance as None.

Once a suitable company is found, the function updates the space column in the C_i, DataFrame to account for the filled internship position and removes the student from the S_i DataFrame to prevent them from being assigned again. An assignment record is created, capturing the student ID, the company's major code, company name, driving distance, count of internships, and remaining space. This record is added to the *assignments* array. Finally, the function creates a new DataFrame from the *assignments* array with appropriate column names and returns it along with the updated *student_data*, reflecting the students who have yet to be assigned.

In conclusion, this algorithm demonstrates an efficient and structured approach to resolving the complex task of internship placements, ensuring students are matched with appropriate companies based on academic alignment and geographic accessibility. This method balances the needs of both students and companies by optimizing the placement process and ensuring that students are placed in relevant and accessible internships.

Algorithm 2 StudentCompanyAssignment

Input: $S_i, C_i\ t$
Output: $assignments_{df}$
Begin
Initialize A← ∅

For each $S \in S_i$ **do**
$student_{id} \leftarrow s.ID$
$major_s \leftarrow s.major$
$loc_s \leftarrow (s.lat, s.lon)$

M ← $\{ c \in C_i \mid c.major = major_s \;\&\&\; c.space > 0 \}$

If |M| = 0:
Continue to next student

For each c ∈ M:
$loc_c \leftarrow (c.lat, c.lon)$
$d(c,s) \leftarrow Haversine(loc_s, loc_c)$
$c.distance \leftarrow d(c,s)$

M ← $sort\ M\ by\ c.distance\ in\ ascending\ order$

For each $c \in M$:
If c.distance ≤ t:
Try:
$D_{drive}(c,s) \leftarrow CalculateDrivingDistance(loc_s, loc_c)$
Catch:
$D_{drive}(c,s) \leftarrow None$

$c.space \leftarrow c.space - 1$
$s \leftarrow S_i\ \{s\}$
A ← $A \cup \{a\}$
Break

Return $assignments_{df}$
End

The Haversine formula helps us find the distance between two points on the Earth using their latitude and longitude.

The Haversine formula in navigation calculates the distance of a circle between latitude and longitude points, assuming the earth's radius R is 6367.45 km. The Haversine formula's assumption ignores the earth's surface structure (valley depth and hill height), which is quite accurate in most calculations because the ellipsoidal effect is eliminated.

In this project, the Haversine method is used to calculate the straight-line distance between the student's location and each company, and the result is stored in a new column called *distance*. After the calculation is performed, companies are sorted in ascending order based on the distance. This prioritizes companies closer to the student, making them more likely candidates for assignment. Here is the Haversine formula [43] written in equation (1) as follows:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \quad (4)$$

To find the driving distance between two places, we use Open Street Maps (OSM) and Python tool called OSMNX. OSM is a great source for detailed map info, including roads, buildings, rivers, and mountains. Many people help keep OSM updated. These include hobbyists, mappers, disaster risk experts, and GIS professionals. Since OSM is open to everyone, anyone can use its data. OSMNX takes this data and uses it to create network for different uses.

For lecturer-company assignments, the framework is divided into several major components: preprocessing pipelines for both lecturers and companies, embedding processes, and a domain matching mechanism.

Algorithm 3 PreprocessText

Input: T_i
Output: T_{clean}

Begin
$T_i \leftarrow RemoveStopwords(T_i)$
$T_i \leftarrow Tokenize(T_i)$
$T_i \leftarrow Stemming(T_i)$
$T_i \leftarrow Lemmatization(T_i)$
End

The area of expertise of lecturers and company description have been undergo preprocessing step where all numeric characters are removed from text to focus on textual data, non-alphanumeric characters are stripped out to standardize the text. The text is tokenized using spaCy, a powerful NLP tool, breaking it down into individual words or tokens. Common words that are irrelevant such as “the”, “is” and “at” are removes.

Algorithm 4 EmbedText

Input: $T_{clean,}M_{model}, I_{type}$
Output: V_t

Begin
$V_t \leftarrow embed([T_{clean}], model = M_{model}, input = I_{type})$
Return V_t
End

After preprocessing, the text data is converted into numerical form. The cleaned and tokenized text is transformed into vectors using Voyage AI embedding techniques using "voyage-large-2". The embedding process captures semantic meaning and contextual relationships between words in the text.

Algorithm 5 CalculateSimilarity

Input: V_t, V_d
Output: $S_{similar}$

Begin
For each $(\boldsymbol{D}, \boldsymbol{V_d})$ ***in*** $\boldsymbol{V_d}$ ***do***
$S_{similar} \leftarrow 1 - cosine\ (V_t, V_d)$
EndFor
Return $\boldsymbol{S_{similar}}$
End

There is the core component where the processed data is used to match lecturers to companies. The similarity between the lecturer's and company's domain is calculated using cosine similarity. This measure helps in identifying how close or related two sets of text data are, based on their vector representations. Based on similarity scores, the best matching domain for each lecturer and company are identified. This involves selecting the company domain that has the highest similarity score with the lecturer's domain description.

$$cosine\ similarity = \frac{A \times B}{||A||||B||} \quad (5)$$

Algorithm 6 LecturerCompanyMatching

Input: L_i, C_i
Output: $Assignments$

Begin
For each $L \in L_i$ ***do***
For each $C \in C_i$ ***do***
If $L_d = C_d$ ***then***
$Assignment \leftarrow L_{id}, C_{id}$)
Break
EndIf
EndFor
EndFor
Return Assignments
End

Lecturers are assigned to companies where their expertise best matches the company's needs. The final matches are compiled, typically in a structured format such as a data frame, showing which lecturer is assigned to which company along with the similarity scores and other relevant details.

E. Algorithm Evaluation

As the complexity of algorithms increases across various domains, evaluating their performance becomes critical to ensure efficiency and effectiveness. This section introduced the comprehensive evaluation of algorithms of student-company assignments through encompassing variance and standard deviation analysis, utilization rate measurement, and spatial analysis. Besides, accuracy, precision, recall and F1-score are used to evaluate the performance of the lecturer-company assignment.

F. Variance

Variance is used as an evaluation metric in this study to evaluate the goodness of an algorithm in the context of driving distance. Variance measures the spread or dispersion of driving distance. A lower variance indicated that the driving distances were more consistent or stable. The equation of variance is written as where n is the number of data points, x_i is each data point, and $\bar{x}$ is the mean of the data.

$$Variance, s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2 \quad (6)$$

G. Standard Deviation

Standard deviation is the square root of the variance and provides another measure of the amount of variation or dispersion in a set of values, which is driving distances. A lower standard deviation could contribute to more reliable and stable outcomes as a low standard deviation implies that the values are tightly clustered around the mean.

$$Standard\ Deviation, s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2} \quad (7)$$

H. Utilization Rate Measure

The utilization rate of the company's resources was scrutinized. This rate was calculated by subtracting the remaining capacity from the initial capacity for each item in the company data, summing these values, and then dividing by the total capacity to find the percentage. The resulting utilization rate stood at *utilization rate (*%), which reflects the extent to which the company's resources were effectively employed during the period under review.

$$Utilization\ Rate = \left(\frac{\sum(capacity_{initial} - capacity_{remaining})}{\sum capacity_{total}}\right) \times 100\ \% \quad (8)$$

I. Statistics

Variance depicted the spread or dispersion of values within each faculty's dataset, providing insights into how significantly the utilization rates deviated from the mean.

J. Accuracy

Accuracy measures the proportion of true results (both true positives and true negatives) among the total number of examine in cases.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

K. Precision

Precision measures the accuracy of positive predictions. It illustrates the ratio of true positive to all positive predicted by the classifier.

$$Precision = \frac{TP}{TP+FP} \quad (10)$$

L. Recall

Recall measures the algorithm's ability to identify all relevant instances. It is crucial for cases where missing a

positive is significantly worse than falsely identifying a negative.

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

M. F1-Score

The F1-Score is the harmonic means of Precision and Recall, and it provides a balance between the two metrics. It is particularly useful when the classes are imbalanced.

$$\text{F1-Score} = \frac{2TP}{2TP+FP+FN} \quad (12)$$

III. Results and Discussion

In assessing driving distances and utilization patterns across faculties, significant variations were observed. Most faculties exhibited high variance and standard deviation in driving distances, reflecting the diverse locations of students and companies. Faculty 9 recorded the highest average driving distance at 0.3605km, with a variance of 4.5699 and standard deviation of 2.1377, indicating a broad range of travel distances.

Conversely, Faculty 6 experienced a notably low utilization rate, primarily due to mismatch between the majority of students majoring in accounting and the absence of companies specializing in this area within the dataset. This spatial mismatch, especially where companies were clustered in specific geographical areas, led to underutilization in some faculties.

TABLE III
FACULTY STATISTICS

Faculty	Mean	Variance	Standard Deviation	Utilization Rate (%)
1	0.1782	0.7919	0.8899	49.71
2	0.0482	0.3589	0.5991	63.61
3	0.0831	0.4426	0.6653	15.03
4	0.0776	0.7727	0.8791	11.52
5	0.0734	0.4433	0.6658	24.77
6	0.0034	9.6310	0.0031	18.35
7	0.0337	0.0422	0.2054	18.13
8	0.0288	0.1521	0.3898	15.98
9	0.3605	4.5699	2.1377	100.00

From Fig 5, markers were sparsely scattered across different states in Malaysia. Blue markers represented iCadets, whereas red markers with a briefcase icon likely signified company. The presence of markers in both East and West Malaysia suggested that the assignment encompassed a national scope.

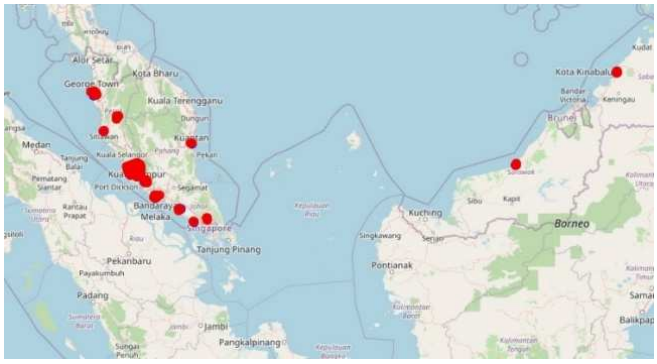

Fig. 5 Geographical distribution of assigned student and company

Fig 6 displayed a dense concentration of red markers in the state of Selangor, indicating a high density of companies in a central location. This clustering of markers represented a significant number of opportunities for iCadets (students) in that area. Given the economic vibrancy and the abundance of corporate entities, students assigned to this region were likely to have access to a diverse range of professional experiences.

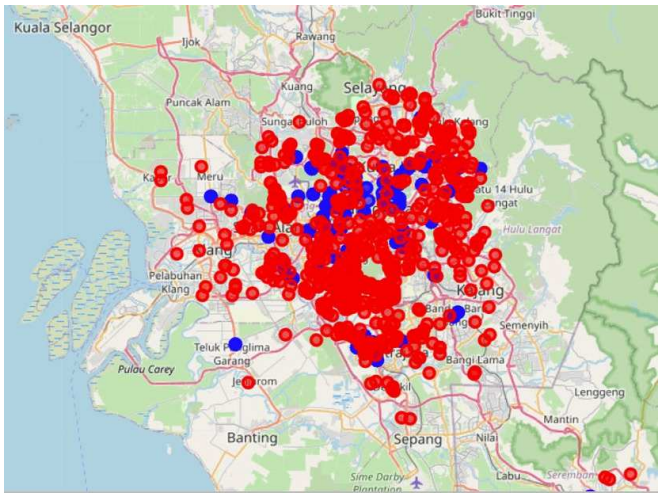

Fig. 6 Dense concentration of assigned student and company in Selangor

After the implementation of the algorithms, students and companies were matched by prioritizing their preferences. Five CSV files were generated, assigned based on distance thresholds of 10 km, 20 km, 30 km, 40 km, and a final assignment. The final assignments CSV file included all assignments for students who were not matched according to their preferences. This file contained their student ID, student and company major codes, company name, driving distance in kilometers, company capacity (count), and company current capacity (space). All students and companies had been successfully assigned at this stage.

For the lecturer-company assignment, the tables above provide a clear comparison of performance metrics both with and without the use of embedding techniques in the domain assignment of lecturers. The results have demonstrated that incorporating embeddings significantly enhanced the effectiveness of the matching process between domain descriptions and related subjects. Specifically, the accuracy improved from 0.464 to 0.6071, precision increased from 0.417 to 0.5058, recall saw an equal rise from 0.464 to 0.6071, and the F1-score advanced from 0.417 to 0.5264.

TABLE IV
LECTURER DOMAIN ASSIGNMENT WITHOUT EMBEDDINGS

Without Embedding (spaCy Method)			
Accuracy	**Precision**	**Recall**	**F1-score**
0.4638	0.4170	0.4638	0.4170

TABLE V
LECTURER DOMAIN ASSIGNMENT WITH EMBEDDINGS

With Embedding (Voyage AI)			
Accuracy	**Precision**	**Recall**	**F1-score**
0.6071	0.5058	0.6071	0.5264

Table VI presents that utilizing embeddings from 100 words description perform better in domain assignments compared to 50 words descriptions. The superiority of the

longer descriptions can contribute to their ability to provide a more comprehensive and detailed representation of a company's profile.

There was a significant improvement in accuracy from 0.6154 to 0.7692 when the description length was increased from 50 words to 100 words. This suggests that longer descriptions provide more detailed information, leading to better matches between lecturers and companies. Precision increased from 0.5744 to 0.7751 with longer descriptions. Higher precision indicates that a greater proportion of the matches identified were correct. Recall improved from 0.6154 to 0.7692. Higher recall means that a greater proportion of relevant matches were successfully identified. The F1-score, which is the harmonic mean of precision and recall, increased from 0.5807 to 0.7484. This comprehensive metric shows a balanced improvement in both precision and recall, demonstrating that longer descriptions significantly enhance the overall quality of the matching process.

TABLE VI
COMPARISON OF DOMAIN ASSIGNMENT PERFORMANCE BY DESCRIPTION LENGTH

Description	Voyage AI Embedding			
	Accuracy	Precision	Recall	F1-score
50 words	0.6154	0.5744	0.6154	0.5807
100 words	0.7692	0.7751	0.7692	0.7484

Improvements in the matching process when using longer descriptions in the embeddings can be observed in increased contextual understanding and enhanced semantic similarity. With more context, embedding captures more relevant information, leading to more accurate matches. Besides, embedding helps in distinguishing between closely related but different concepts, reducing false positives and improving the granularity of matches, leading to more precise results. This enhanced capability makes it easier for models to understand the intended meaning behind words or phrase with multiple meanings. Models trained on richer data tend to generalize better, performing better on unseen data. The table below provided an overview of the successful outcomes of rule-based algorithms used to assign lectures to companies based on their domains.

TABLE VII
RESULT OF LECTURER-COMPANY ASSIGNMENT

Lecturer ID	Company ID	Domain
1	1001	Data Science
2	1002	Game Development
3	1003	Information System
4	1004	Software Engineering
5	1005	Cybersecurity

IV. CONCLUSION

This study has significantly optimized the implementation of the iCadet program, enhancing the process of assigning penultimate-year students and lecturers to companies. This study first addressed the pivotal factors of geographical location and domain-specific alignment in the student-company placement process by implementing Haversine and OSMnx method. This study effectively minimized the logistical challenges of distance, ensuring that students are placed with companies within shortest travel confines.

Next, this study underscores the significant benefits of employing embedding techniques in lecturer-company assignments. Through the application of machine learning methods, including the use Voyage AI embeddings and Cosine Similarity for domain matching, the use of embeddings to has markedly increased the accuracy of matching lecturers to companies. This study documented improvements in all performance metrics including accuracy, precision, recall and F1-score when embeddings were employed. However, the algorithms developed and utilizes in this study as those for matching and distance calculations, might be tailored for specific conditions or datasets, which could limit their applicability in different contexts or with different data structures.

For further improvement, a job portal dataset will be used to solve the remaining unassigned student issue, which currently is solved by assigning all those students to the company who have available space regardless of the student profile which is their geographical location and specialization.

REFERENCES

[1] ICADET, "ICADET." [Online]. Available: https://www.mmu.edu.my/icadet/

[2] C. Qin *et al.*, "An enhanced neural network approach to person-job fit in talent recruitment," *ACM Transactions on Information Systems (TOIS)*, vol. 38, no. 2, pp. 1–33, 2020, doi:10.1145/3376927.

[3] S. Y. Ong, C. Y. Ting, H. N. Goh, A. Quek, and C. L. Cham, "Workplace Preference Analytics Among Graduates," *Journal of Informatics and Web Engineering*, vol. 2, no. 2, pp. 233–248, 2023, doi:10.33093/jiwe.2023.2.2.17.

[4] M. Z. Abd Majid, M. Hussin, M. H. Norman, and S. Kasavan, "The employability skills among students of Public Higher Education Institution in Malaysia," *Geografia*, vol. 16, no. 1, 2020, doi:10.17576/geo-2020-1601-04.

[5] M. M. Hussain, S. Akbar, S. A. Hassan, M. W. Aziz, and F. Urooj, "Prediction of Student's Academic Performance through Data Mining Approach," *Journal of Informatics and Web Engineering*, vol. 3, no. 1, pp. 241–251, 2024, doi:10.33093/jiwe.2024.3.1.16.

[6] J. Liu, L. Deng, H. Miao, Y. Zhao, and K. Zheng, "Task assignment with federated preference learning in spatial crowdsourcing," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1279–1288, doi:10.1145/3511808.3557465.

[7] X. Wei, B. Sun, J. Cui, and M. Qiu, "Location-and-Preference Joint Prediction for Task Assignment in Spatial Crowdsourcing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 928–941, 2022, doi:10.1109/TCAD.2022.3188960.

[8] Z. Wang, Y. Zhao, X. Chen, and K. Zheng, "Task assignment with worker churn prediction in spatial crowdsourcing," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2070–2079, doi:10.1145/3459637.348230.

[9] L. H. Pinto and P. C. Pereira, "'I wish to do an internship (abroad)': investigating the perceived employability of domestic and international business internships," *High Educ (Dordr)*, vol. 78, pp. 443–461, 2019, doi: 10.1007/s10734-018-0351-1.

[10] D. Odlin, M. Benson-Rea, and B. Sullivan-Taylor, "Student internships and work placements: approaches to risk management in higher education," *High Educ (Dordr)*, vol. 83, no. 6, pp. 1409–1429, 2022.

[11] S. M. Zehr and R. Korte, "Student internship experiences: learning about the workplace," *Education+ Training*, vol. 62, no. 3, pp. 311–324, 2020, doi: 10.1108/ET-11-2018-0236.

[12] O. T. Adeosun, A. I. Shittu, and T. J. Owolabi, "University internship systems and preparation of young people for world of work in the 4th industrial revolution," *Rajagiri Management Journal*, vol. 16, no. 2, pp. 164–179, 2022, doi: 10.1108/RAMJ-01-2021-0005.

[13] M. Molino, C. G. Cortese, and C. Ghislieri, "The promotion of technology acceptance and work engagement in industry 4.0: From personal resources to information and training," *Int J Environ Res Public Health*, vol. 17, no. 7, p. 2438, 2020, doi:10.3390/ijerph17072438.

[14] I. Kapareliotis, K. Voutsina, and A. Patsiotis, "Internship and employability prospects: assessing student's work readiness," *Higher Education, Skills and Work-Based Learning*, vol. 9, no. 4, pp. 538–549, 2019, doi: 10.1108/HESWBL-08-2018-0086.

[15] A. C. G. Ocampo *et al.*, "The role of internship participation and conscientiousness in developing career adaptability: A five-wave growth mixture model analysis," *J Vocat Behav*, vol. 120, p. 103426, 2020, doi: 10.1016/j.jvb.2020.103426.

[16] M. T. Hora, E. Parrott, and P. Her, "How do students conceptualise the college internship experience? Towards a student-centred approach to designing and implementing internships," *Journal of Education and Work*, vol. 33, no. 1, pp. 48–66, 2020, doi:10.1080/13639080.2019.1708869.

[17] A. Fauzan, M. B. Triyono, R. A. P. Hardiyanta, R. W. Daryono, and S. Arifah, "The Effect of Internship and Work Motivation on Students' Work Readiness in Vocational Education: PLS-SEM Approach," *Journal of Innovation in Educational and Cultural Research*, vol. 4, no. 1, pp. 26–34, 2023, doi: 10.46843/jiecr.v4i1.413.

[18] B. Xing, D. Xie, S. Li, and Q. Wang, "Why you leave and what can we do? The roles of job burnout and vocational skill in hotel internships," *J Hosp Leis Sport Tour Educ*, vol. 32, p. 100424, 2023, doi: 10.1016/j.jhlste.2023.100424.

[19] Q. Xu *et al.*, "The relationship between personality traits and clinical decision-making, anxiety and stress among intern nursing students during COVID-19: A cross-sectional study," *Psychol Res Behav Manag*, pp. 57–69, 2023.

[20] Q. Yang, L. Yang, C. Yang, X. Wu, Y. Chen, and P. Yao, "Workplace violence against nursing interns and patient safety: The multiple mediation effect of professional identity and professional burnout," *Nurs Open*, vol. 10, no. 5, pp. 3104–3112, 2023, doi:10.1002/nop2.1560.

[21] Y. Kim *et al.*, "Respiratory sound classification for crackles, wheezes, and rhonchi in the clinical field using deep learning," *Sci Rep*, vol. 11, no. 1, p. 17186, 2021.

[22] F. Bittmann and V. S. Zorn, "When choice excels obligation: about the effects of mandatory and voluntary internships on labour market outcomes for university graduates," *High Educ (Dordr)*, vol. 80, no. 1, pp. 75–93, 2020, doi: 10.1007/s10734-019-00466-5.

[23] S. Chaurasia, "Student Internship Placement Management System using Python," *International Journal of Research in Science & Engineering (IJRISE) ISSN: 2394-8299*, vol. 3, no. 03, pp. 30–49, 2023.

[24] H. Mydyti, "Internship management system (IMS)," M.S. thesis, South East European University, Tetovo, North Macedonia, 2023. [Online]. Available: http://repository.seeu.edu.mk/.

[25] R. Yan, R. Le, Y. Song, T. Zhang, X. Zhang, and D. Zhao, "Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 914–922, doi: 10.1145/3292500.3330963.

[26] A. Tamang and S. Adhikari, "Scoring of Resume and Job Description using word2vec & matching them using Gale Shapley Algorithm", doi:10.1007/978-981-16-2126-0_55.

[27] C. Daryani, G. S. Chhabra, H. Patel, I. K. Chhabra, and R. Patel, "An automated resume screening system using natural language processing and similarity," *Ethics and Information Technology [Internet]. Volkson Press*, pp. 99–103, 2020, doi:10.26480/etit.02.2020.99.103.

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

[29] K. Tejaswini, V. Umadevi, S. M. Kadiwal, and S. Revanna, "Design and development of machine learning based resume ranking system," *Global Transitions Proceedings*, vol. 3, no. 2, pp. 371–375, 2022, doi:10.1016/j.gltp.2021.10.002.

[30] D. Lamba, S. Goyal, V. Chitresh, and N. Gupta, "An integrated system for occupational category classification based on resume and job matching," in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[31] S. Gadegaonkar, D. Lakhwani, S. Marwaha, and Prof. A. Salunke, "Job Recommendation System using Machine Learning," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 2023, pp. 596–603. doi:10.1109/ICAIS56108.2023.10073757.

[32] A. Mulay, S. Sutar, J. Patel, A. Chhabria, and S. Mumbaikar, "Job recommendation system using hybrid filtering," in *ITM Web of conferences*, 2022, p. 2002, doi: 10.1051/itmconf/20224402002.

[33] S. A. Alsaif, M. Sassi Hidri, I. Ferjani, H. A. Eleraky, and A. Hidri, "NLP-based bi-directional recommendation system: Towards recommending jobs to job seekers and resumes to recruiters," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 147, 2022, doi:10.3390/bdcc6040147.

[34] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, "A systematic review: machine learning based recommendation systems for e-learning," *Educ Inf Technol (Dordr)*, vol. 25, pp. 2635–2664, 2020, doi: 10.1007/s10639-019-10063-9.

[35] A. Giabelli, L. Malandri, F. Mercorio, M. Mezzanzanica, and A. Seveso, "Skills2Job: A recommender system that encodes job offer embeddings on graph databases," *Appl Soft Comput*, vol. 101, p. 107049, 2021, doi: 10.1016/j.asoc.2020.107049.

[36] Z. Cui *et al.*, "Personalized recommendation system based on collaborative filtering for IoT scenarios," *IEEE Trans Serv Comput*, vol. 13, no. 4, pp. 685–695, 2020, doi: 10.1109/TSC.2020.2964552.

[37] W. Lei *et al.*, "Estimation-action-reflection: Towards deep interaction between conversational and recommender systems," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 304–312, doi: 10.1145/3336191.3371769.

[38] D. Jannach, A. Manzoor, W. Cai, and L. Chen, "A survey on conversational recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021, doi: 10.1145/3453154.

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

[40] 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, p. 63, 2020.

[41] D. Wang and B. Zhang, "A Path Simulator Focusing on Time Consumption-Based on the Transport Network and the Data of Public Traffic Vehicles in Shanghai", doi: 10.23977/jeis.2023.080305.

[42] K. Cibis, J. Wruk, and M. Zdrallek, "Application of Routing Algorithms in Automated Distribution Network Planning," in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, 2020, pp. 1–6, doi: 10.1109/SEST48500.2020.9203552.

[43] A. Upadhyay, "Haversine formula – Calculate geographic distance on earth," 2023. [Online]. Available: https://www.igismap.com/haversine-formula-calculate-geographic-distance-earth/.