Journal of Current Science and Technology, October-December 2025 Vol. 15 No. 4, Article 137

ISSN 2630-0656 (Online)

Cite this article: Chaiaroon, P., Promrit, N., Sitdhisanguan, K. Waijanya, S., & Kanraweekultana, N. (2025). Digital workforce matching: A machine learning approach for skill-based job classification and recommendation. *Journal of Current Science and Technology*, *15*(4), Article 137. https://doi.org/10.59796/jcst.V15N4.2025.137

Journal of Current Science and Technology

Journal homepage: https://jcst.rsu.ac.th

Digital Workforce Matching: A Machine Learning Approach for Skill-Based Job Classification and Recommendation

Paweena Chaiaroon[1], Nuttachot Promrit [1], Karanya Sitdhisanguan [1], Sajjaporn Waijanya[1,*], and Natratanon Kanraweekultana[2]

[1]Center of Excellence in AI and NLP, Department of Computing, Faculty of Science, Silpakorn University, Sanam Chandra Palace Campus, Nakhon Pathom 73000, Thailand
[2]Information Technology and Digital Business Program, Faculty of Business Administration, Rajamangala University of Technology Krungthep, Bangkok, 10120, Thailand.

*Corresponding author; E-mail: waijanya_s@silpakorn.edu

Received 9 April 2025; Revised 18 June 2025; Accepted 25 June 2025; Published online 20 September 2025

Abstract

This research presents an integrated machine learning approach for optimizing digital workforce matching in Thailand's evolving digital economy. The study develops a novel job recommendation system combining Natural Language Processing (NLP) with Random Forest classification to analyze job market data from Thailand's leading recruitment platforms. Using FastText for initial job classification and a Random Forest model for skill-based matching, the system achieves 75% accuracy in job recommendations across 20 digital job categories. The methodology incorporates automated skill extraction, cross-validated model comparison, and a user-friendly web interface for practical applications. Our findings reveal distinct skill clusters and job-skill relationships in Thailand's digital sector, with the Random Forest model outperforming traditional Decision Tree approaches by 4% in accuracy metrics. The system demonstrates robust performance in real-world testing, achieving 86.67% accuracy in matching previously unseen job postings. This research contributes to both theoretical understanding of skill-based job matching and practical workforce development, offering insights for curriculum development and career planning for workforce development stakeholders in Thailand's digital sector.

Keywords: *digital workforce matching; skill-based job; classification; recommendation; machine learning model*

1. Introduction

Digital technology has become an essential driver across both public and private sectors, influencing operations, service delivery, and socio-economic transformation. The inherently dynamic and rapidly evolving nature of digital technology continues to reshape economic structures and societal needs. Consequently, digital literacy and expertise, particularly among digital technology professionals, are now critical components of national development. In Thailand, however, the supply of digital professionals remains critically low. As of 2019, only 1.49% of the national workforce is employed in the digital technology sector, reflecting an insufficient supply of digital talent to meet market expectations (Office of the National Digital Economy and Society Commission, 2019). From an educational perspective, the pipeline of future professionals is also limited: in the academic year 2023, just 4.50% of undergraduate students were enrolled in Information and Communication Technologies (ICTs) programs (Ministry of Higher Education, Science, Research and Innovation, 2024). This low enrollment indicates a significant shortfall in the higher education system's ability to supply the workforce needed for a rapidly digitizing economy.

This supply-side inadequacy not only limits workforce availability but also raises concerns about the alignment between graduate skillsets and market demands. Rapid technological advancements continuously reshape the digital labor landscape, requiring universities to adapt curricula that equip students with relevant, up-to-date competencies. Without a clear understanding of in-demand digital skills, educational institutions risk producing graduates with mismatched or outdated qualifications, exacerbating labor shortages and contributing to rising unemployment in the tech sector. On the demand side, employers are faced with an equally pressing challenge: identifying and recruiting candidates with the precise skillsets required in an increasingly complex and specialized job market. Yet the current landscape of workforce-skills data is fragmented, inconsistent, and largely unstructured. There is no centralized or regularly updated statistical record that details occupational-level digital skill requirements (Office of the National Economic and Social Development Council, 2017). This makes it difficult to pinpoint the exact competencies demanded for specific roles. Moreover, job announcements often vary in terminology, titles, and listed qualifications depending on the employer, creating further barriers to accurate and timely market analysis (Office of the National Digital Economy and Society Commission, 2019).

To effectively bridge the gap between supply and demand, it is crucial to conduct a comprehensive and continuous analysis of labor market data, especially by examining job postings and extracting skill requirements. Such analysis must be scalable and adaptive to keep pace with the fluid nature of technological change. In this context, advanced data analytics and digital tools can play a transformative role in informing curriculum design, workforce planning, and policy formulation aimed at developing a digitally competent labor force.

The challenge of producing and developing an adequate digital technology workforce, both in terms of quantity and quality, necessitates a deep understanding of market-demanded skills. This task is particularly complex due to the varied, intricate, and unstructured nature of workforce-skills requirements data, with no comprehensive statistics regularly compiled at the occupational level. Additionally, the dynamic nature of digital technology means that skill requirements are constantly evolving, requiring continuous analysis and updating of market demand data. This analysis typically involves examining detailed job announcements for each position to identify required skills, a process complicated by the large volume of data and inconsistencies in job titles and qualification requirements across different employers. Research in this field has employed various sophisticated approaches to address these challenges. Many studies utilize Natural Language Processing (NLP) as their primary technique for analyzing and grouping job requirements data from websites (Pundir et al., 2024; Anthony, 2024; Pias et al., 2024). Some researchers have implemented decision trees to classify required skills for specific positions (Pillai & Amin, 2020), while others have employed rapid automatic keyword extraction (RAKE) techniques to identify key market expectations and needs (Phaphuangwittayakul et al., 2018). The analysis of relationships between job positions and required skills has been facilitated through the use of graph databases, which are particularly well-suited for storing and accessing interconnected data (Giabelli et al., 2021). The methodology landscape has further expanded with the implementation of ontology-based approaches to structuring relationships between skills and job requirements (Fuzul & Horvat, 2019; Ibadov et al., 2020). Classification methods have evolved to include sophisticated algorithms such as neural networks (Qin et al., 2020), Support Vector Machine (SVM), Naïve Bayes, and k-NN (Alghamlas & Alabduljabbar, 2019). The field has seen significant advancement with the introduction of computational job market analysis using deep learning techniques, particularly in skill extraction and job classification. However, the rapid growth and interdisciplinary nature of these methodologies have led to challenges in dataset creation and characterization (Senger et al., 2024). Recent studies have focused on aligning big data professional education with labor market demands (Hassan et al., 2023), exploring the multifaceted skillsets required by employers and emphasizing the importance of continuous learning in response to rapid technological advancements. Advanced text analysis techniques, including Word2Vec, LSTM, and BERT algorithms, have been employed not only for analyzing current job positions but also for predicting future market demands (Melo et al., 2023; Weichselbraun et al., 2024). These predictive models have shown promising results, with machine learning techniques such as SVM and neural networks achieving accuracy rates of up to 82% and 88% respectively in predicting employer skill requirements (Weichselbraun et al., 2024). This complex landscape of digital workforce development presents both challenges and opportunities

for Thailand's economic and social advancement, which is a research gap that has not yet been addressed by digital workforce matching and applied to skill-based job classification and recommendation using Machine Learning. The research aims to address the fundamental question of "which skill sets are necessary for digital technology jobs in Thailand?" through comprehensive data analysis. This includes studying the principles and techniques of data collection from websites, implementing text data analysis processes, and creating data recommendation models through surveys and analysis (Pundir et al., 2024; Ingole et al., 2024). The ultimate goal is to present a detailed analysis of the overall demand for digital job positions, identify the digital skills most sought after by the market, and map the relationships between job positions and their required skills in Thailand's digital technology sector.

The importance of this research extends beyond immediate workforce development, as it contributes to the broader objective of transforming Thailand into an innovation-driven economy. By providing detailed insights into market demands and skill requirements, this research supports both individual career development planning and institutional workforce development strategies, ultimately contributing to the nation's technological and economic advancement.

2. Objectives

The objective of this research is to develop a machine learning-based model for job classification and to identify the relationship between individual skills and job positions. The model aims to accurately recommend the top 3 most suitable job positions for a user based on their input skills, with a target accuracy of at least 80% as measured by validation datasets. The model will also generate a list of skill gaps for each suggested job position to guide personalized skill development. The ultimate goal is to implement this model in a user-friendly format, such as a prototype web or mobile application, to support real-world decision-making for job seekers and career development platforms.

3. Materials and Methods

Since this research presents a model for recommending digital job positions based on technical skills, the technical process begins with collecting data from the top 5 job posting websites in Thailand. The job titles from the postings are then grouped because it has been observed that the same job positions are often referred to by various names. To classify these job positions, this research develops and trains a job position classification model using FastText. The job descriptions and job responsibilities were processed to extract only the relevant technical skills. These extracted skills were then labeled with job position groups in a multi-class format. A Job Recommendation Model was subsequently developed and trained using the Random Forest algorithm. The well-trained model was deployed on a server, enabling it to recommend job positions based on the skills provided. The framework of the proposed work is given in Figure 1.

3.1 Data Collection

Data collection of digital job posting details began with selecting the top 5 most popular job posting websites in Thailand, based on data from Google Trends in 2021. The top 5 most popular websites are: 1) Jobthai, 2) Jobsdb, 3) JOBBKK, 4) JOBTOPGUN, and 5) LinkedIn. The data collection process from job posting websites followed a structured procedure. As shown in Figure 2, data from all five websites were collected using the web scraping technique (Pillai & Amin, 2020). The process started by specifying the URL and sending a request to the web page from which the researcher intended to collect data. The webpage was then saved as an HTML file, allowing the researcher to test data extraction from the HTML structure without needing to send repeated requests to the server. For websites that did not allow direct requests, Selenium's web driver was used to simulate access by opening a web browser instead. Once the files were saved successfully, they were opened using Beautiful Soup, a Python library, to extract data from the HTML structure (HTML parser), as illustrated in Figure 3. To extract necessary information from content with multiple text elements, regular expressions were applied, as shown in Figure 4. Only the essential information required for data analysis was selected, including job title, company, location, salary, posted date, job description, job responsibilities, qualifications, jobsite, and link. A total of 11,365 job positions were collected during the specified data extraction period from online websites. Finally, the selected data were structured into a DataFrame format.

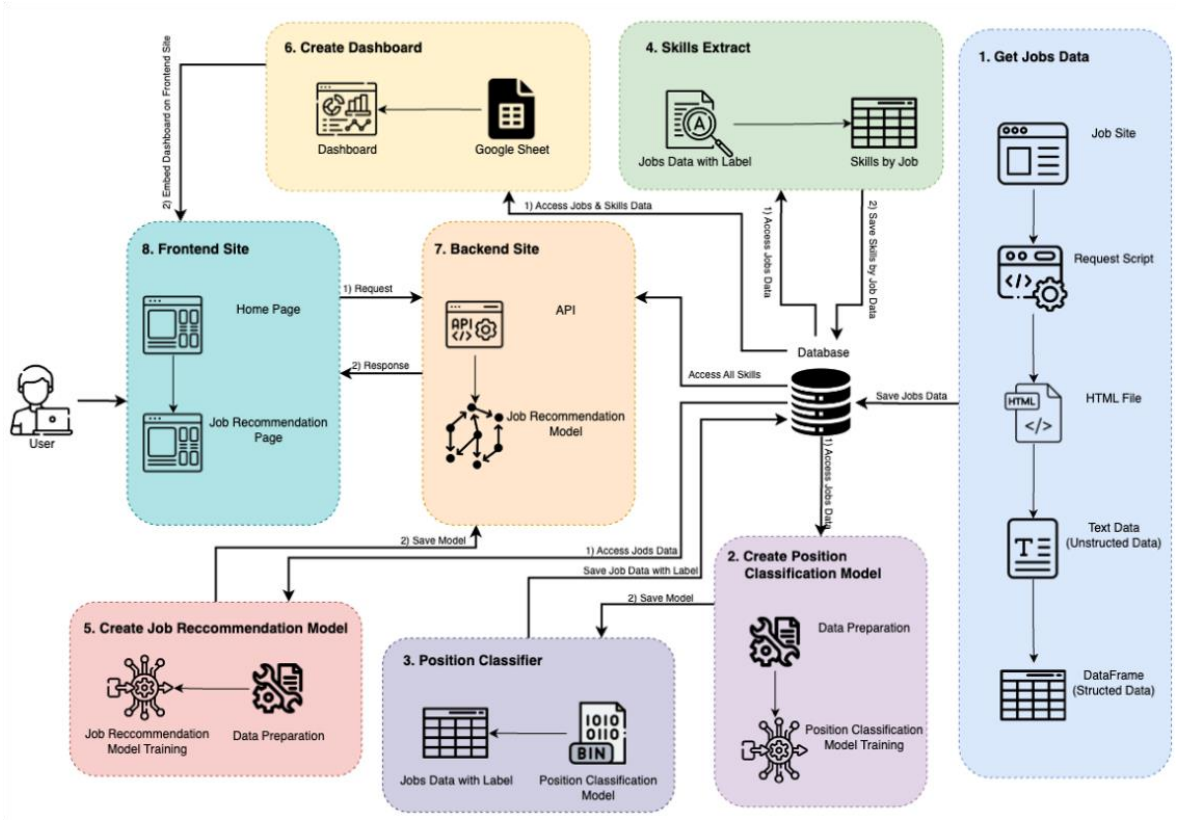

Figure 1 Methodology Framework

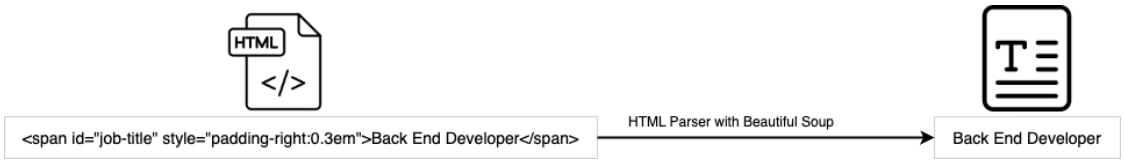

Figure 2 Example of extracting data from HTML structure

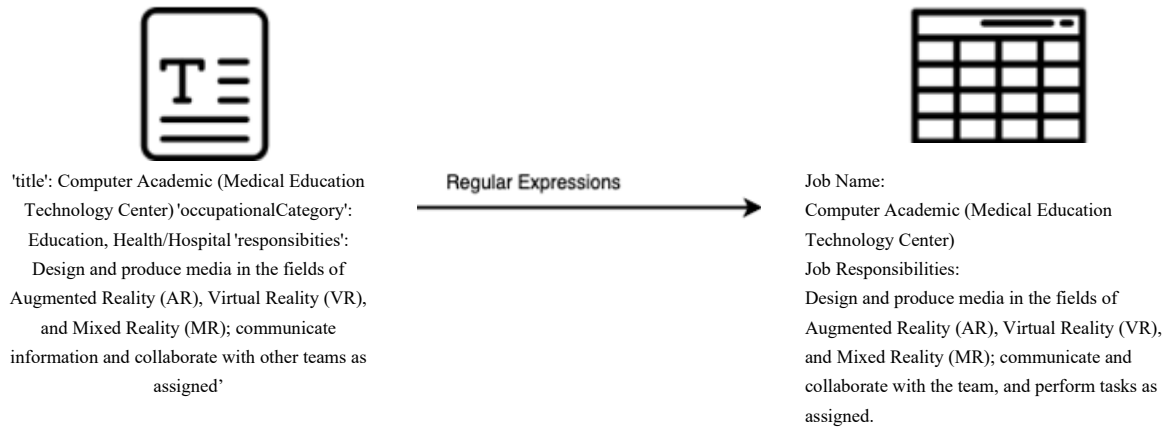

Figure 3 Example of extracting data from content consisting of multiple texts

3.2 Job Position Classification Model

3.2.1 Data Preparation for Job Position Classification Model

Due to the wide range of job titles appearing on job posting platforms, many different titles can have the same meaning. To address this diversity, the first step involves compiling all collected job titles and organizing them into job groups a process referred to as "labeling" by the researcher. Using these labeled job titles and job categories, a job classification model can then be developed to systematically group similar job positions (Rahhal, 2023), as shown in Table 1.

3.2.2 Job Position Classification Model Training

The job position classification model was built using fastText, a text clustering library (Dehghani & Manthouri, 2021). To build a model, two pieces of input data are required: job group and job title. This requires "label" to be prefixed to the job group name, and the imported data must be formatted as specified by fastText. As shown in the sample input data in Table 2. The structure of the model is illustrated in Figure 4.

Table 1 Example of Job Position Classification Based on Job Titles

Jobname	Job Position Group
IT Support Staff	It-support
IT Support Officer (Junior Level)	
Senior IT Support onsite (BTS Ploenchit)	
Full Stack Developer (Digital Transformation)	full-stack-developer/programmer
Senior Fullstack Developer (Java / Javascript)	
Senior Fullstack Developer	
Senior Software Engineer (Java, AWS)	software-engineer
Software Engineer/พนักงานฝ่ายค้นคว้าคว้าและพัฒนาวิจัย - Software	
Software Engineer (Junior/Senior)	

Table 2 Example of input data for building job classification model

__label__	jobname
__label__it-support	it support staff
__label__it-support	it support officer (junior level)
__label__it-support	senior it support onsite (bts ploenchit)
__label__full-stack-developer/programmer	full stack developer (digital transformation)
__label__full-stack-developer/programmer	senior fullstack developer (java / javascript)
__label__full-stack-developer/programmer	senior fullstack developer
__label__software-engineer	senior software engineer (java, aws)
__label__software-engineer	software engineer/ Research and Development Staff - software

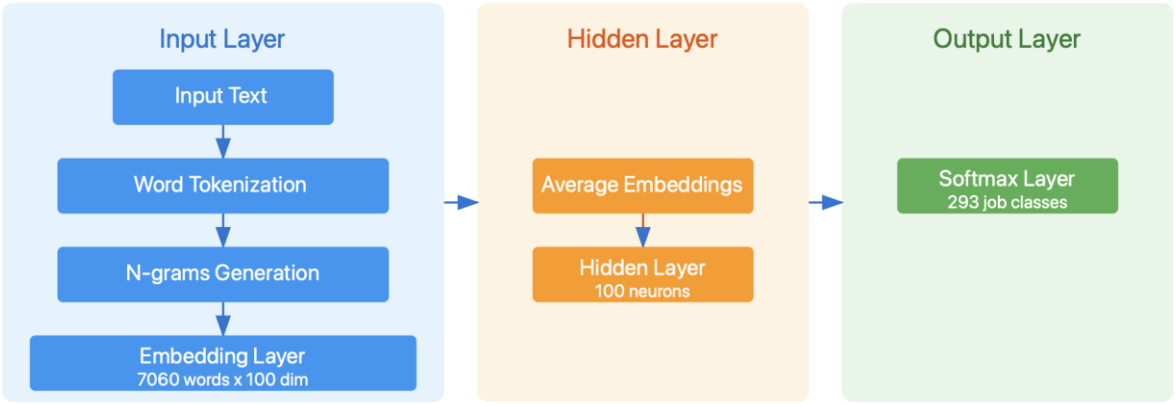

Figure 4 Job Position Classification FastText Model Structure

As shown in Figure 4, the FastText model architecture comprises three main components:

1) Input Layer

The input to the model consists of job title text (jobname), which is preprocessed through PyThaiNLP tokenization to handle Thai language text effectively. The tokenized text is further processed to generate word n-grams (with n=2), capturing local word order and contextual patterns. The n-grams are transformed into dense vector representations using an embedding layer, mapping a vocabulary of 7,060 words into 100-dimensional embedding vectors (Yahya et al., 2024).

2) Hidden Layer

The embedding vectors are aggregated through mean pooling, producing a unified representation of the input text. This representation is passed into a hidden layer comprising 100 neurons, where higher-level features of the text data are learned (Dalal et al., 2024).

3) Output Layer

The final layer uses a softmax activation function to generate a probability distribution across the target classes. This layer maps the learned representations to one of 293 distinct job classes, with the softmax function ensuring that the output probabilities sum to one (Htet & San, 2024). This allows the model to make probabilistic predictions for job classification tasks.

4) Hyperparameter Configuration

The hyperparameter configuration was optimized to enhance model performance. We employed a learning rate of 0.1, striking a balance between convergence speed and training stability. Training proceeded for 500 epochs (Bhushan et al., 2024), ensuring adequate model convergence while preventing underfitting. Word-level bigrams were utilized to preserve contextual information in both Thai and English job titles, capturing meaningful word combinations critical for job classification. The multi-class classification task was optimized using the softmax loss function (Htet & San, 2024).

3.3 Job Recommendation by Skill Model

3.3.1 Data Preparation for Job Recommendation Model

The data preparation process consists of two main phases: skill extraction and data transformation. In the skill extraction phase, 10,424 digital skills-such as "self-learning," "security," and "cloud"-are tokenized to create a custom skill dictionary. This study uses PyThaiNLP, a Python NLP library, for word tokenization (Hardeniya et al., 2016). To extract skill information from text, the researcher first builds a skill dictionary and then identifies specified skill terms within the text data. The texts used for this purpose are compiled from the job_description, job_responsibilities, and qualifications fields. Once extracted, the skill data is stored in a database, as shown in Table 3.

In the data transformation phase, job postings and their associated skills are merged into a single table to analyze the frequency of required skills per job. The table's first column contains the job posting ID, the second column shows the job category, and the remaining columns represent the extracted skills. Each skill is marked with a binary value: '1' if the skill is required for the job posting and '0' if not. This structure enables the mapping of each the job posting to its relevant skills using binary indicators, with job posting ID serving as the primary key, as demonstrated in Table 4.

In the Thai job market, Full-stack Developers/Programmers are typically responsible for both front-end and back-end development, aiming to quickly deliver features that meet business needs. This role is well-suited for startups or agile projects and often emphasizes Java skills. In contrast, Software Engineers focus on software architecture, system design, and long-term, scalable solutions, making them more suitable for large organizations that require robust systems, typically emphasizing C# skills.

To improve data quality and reduce noise from workplace-specific requirements, a frequency-based filtering method was applied. Skills appearing in more than 20% of job postings within a given category were assigned a value of '1', while those below this threshold received a '0'. Finally, these binary values were multiplied by the original frequency counts to reflect the true intensity of skill demand across job categories, as shown in Table 5.

Table 3 Example of a data structure of required skills according to job postings

id	job_id	label	skill	posted_date
155124	12717	it-support	information security	2023-07-31
155125	12717	it-support	self learning	2023-07-31
155126	12717	it-support	security	2023-07-31
155127	12717	it-support	cloud	2023-07-31

Table 4 Skill Frequency Required for Each Job Posting

job_id	label	skill_.net	skill_cloud	skill_java	skill_C#	…	skill_zyxel
15722	full-stack-developer/programmer	1	0	1	0	…	0
15801	software-engineer	1	0	0	1		0
15879	software-engineer	0	0	0	1		0
15885	it-support	1	1	0	0		1
15999	full-stack-developer/programmer	0	0	1	0		0
16052	it-support	0	1				0
16128	full-stack-developer/programmer	1	0	1	0		0
16554	software-engineer	1	0	0	1		0
18753	it-support	1	1	0	0		0

Table 5 filtering the essential skills required for each job category.

label	skill_.net	skill_cloud	skill_java	skill_C#	…	skill_zyxel
.net-developer/programmer	1	0	1	0	…	0
it-support	0	1	0	0		0
developer/programmer	1	0	1	0		0
full-stack-developer/programmer	1	0	1	0		0
it-support	1	1	0	0		0
full-stack-developer/programmer	1	0	1	0		0

3.3.2 Job Recommendation Model Training

The study focused on digital technology positions with substantial representation in the dataset. We established a minimum threshold of 50 samples per job category to ensure statistical significance. The final dataset encompassed 20 distinct job positions as follows: 1) .Net-developer /programmer 2) Back-end-developer / programmer 3) Business-analyst 4) Data-analyst 5) Data-engineer 6) Database-administrator 7) Developer / programmer 8) DevOps 9) Front-end-developer / programmer 10) Full-stack-developer / programmer 11) IT-support 12) Java-developer / programmer 13) Mobile-developer / programmer 14) Network-engineer 15) Project-manager 16) Software-engineer 17) System-engineer 18) Tester 19) UX/UI-designer and 20) Web-developer/programmer. Job positions with overlapping skill requirements and similar role descriptions were consolidated to minimize data redundancy and improve model robustness. Overlapping job skills were collapsed by selecting only the group of positions with a larger sample size and grouping positions with similar names and required skills. Positions with overlapping skills were grouped by counting the skill frequencies of each position and selecting positions with higher frequencies. For example, most IT Support positions require cloud skills as their primary focus. .NET skills came up, but they were a small percentage, so cloud skills remained the primary focus.

1) Cross-Validation Strategy

To rigorously evaluate model performance, particularly given the constraints of limited data availability, we implemented a 5-fold cross-validation methodology (Mahlich et al., 2024). The dataset X was partitioned into five mutually exclusive subsets.

Where X_i represents a distinct subset of the data. For each iteration i of the cross-validation process, the training set was constructed by combining all subsets except the i^{th} fold, as defined in equation (1);

$$X_{Train}^{(i)} = \bigcup_{j \neq i} X_i \quad (1)$$
$$X_{Test}^{(i)} = X_i$$

Performance evaluation for each fold was conducted using the metric defined in equation (2);

$$M_i = \text{Evaluate Model } (X_{Train}^{(i)}, X_{Test}^{(i)}) \quad (2)$$

The overall model performance was then calculated by averaging across all folds, as shown in equation (3);

$$M_{Mean} = \frac{1}{5} \sum_{i=1}^{5} M^i \quad (3)$$

2) Model Implementation

We implemented and compared two machine learning approaches: Decision Tree and Random Forest classifiers.

The Decision Tree model utilized the Gini impurity criterion for node splitting as shown in equation (4);

$$Gini(D) = 1 - \sum_{K-1}^{K} P_k^2 \quad (4)$$

where P_k represents the proportion of class k in dataset D. The model performance for each fold was evaluated using equation (5);

$$M_i = Evaluate(X_{test}^{(i)}, \hat{y}^{(i)}) \quad (5)$$

with the overall performance calculated using equation (6);

$$M_{Mean} = \frac{1}{5} \sum_{i=1}^{5} M_i \quad (6)$$

The complete Decision Tree evaluation process is expressed in equation (7);

$$M_{Mean} = \frac{1}{5} \sum_{i=1}^{5} \text{Evaluate } (X_{test}^{(i)}, \text{Decision Tree } (X_{train}^{(i)})) \quad (7)$$

For the Random Forest implementation, we employed a more complex methodology beginning with bootstrap sampling as defined in equation (8);

$$X_b \sim Bootstrap(X) \quad (8)$$

followed by feature subset selection shown in equation (9);

$$X_b^{(i)} \subseteq \{1,2,\ldots..,p\}, \left|F_b^{(i)}\right| = m \quad (9)$$

where m features are selected from p total features. Individual decision trees t_b^(i) are created using these subsets, with final classification determined through majority voting as expressed in equation (10);

$$\hat{y}^{(i)} = \text{mode } \{t_1^{(i)}(x), t_2^{(i)}(x), \ldots\ldots..t_b^{(i)}(x), \} \quad (10)$$

The complete Random Forest model evaluation is defined in equation (11);

$$M_{Mean} = \frac{1}{5} \sum_{i=1}^{5} \text{Evaluate } (X_{test}^{(i)}, \text{RamdomForest}(X_{train}^{(i)})) \quad (11)$$

This systematic approach, encompassing equations (5)-(11) provides a comprehensive framework for model development and evaluation, with the Random Forest implementation building upon and enhancing the base Decision Tree methodology.

The best performing model is the Random Forest Model, able to be specifying the best parameters (Best Parameter) to create a model with the highest efficiency, then creating a model for actual use by using all the data to create a model to recommend job groups from skills and then saving the resulting model to prepare for further use on web applications. Using GridSearchCV (Chaudhary et al., 2016; Hakim et al., 2024; Kanraweekultana et al., 2024) to find the best parameter as follows: 1) n_estimators = 400, 2) max_depth = 60, 3) min_samples_leaf = 1, 4) min_samples_split = 2. In which the obtained model has 400 decision trees, and the max depth of each decision tree is 60 layers demonstrating that the decision tree model shows the first two examples, with the Root node separated by skill_java < = 0.5, where if the value of skill_java < = 0.5, the data is passed to the left node. But if the value is greater than 0.5, the data will be sent to the right node, the next node on the left side is separated by skill_android < = 0.5, the right side is separated by skill_css <= 0.5. as illustrated in Figure 5.

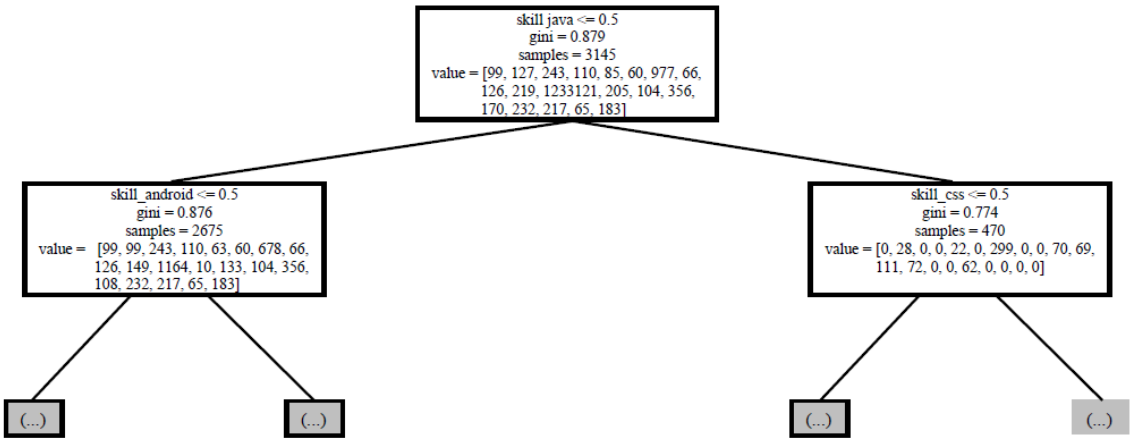

Figure 5 A decision tree from a Random Forest model built for practical use

Using Random Forest is a suitable choice for model development in this task due to the algorithm's advantages in handling complex data and reducing the risk of overfitting, which is often a major drawback when using a single Decision Tree. Random Forest combines the results from multiple trees to make decisions, thereby reducing the volatility that may arise from learning through only a single tree, which can affect model accuracy. Additionally, Random Forest automatically selects important features for decision-making, enhancing prediction efficiency. Even though the model achieved only 75% accuracy despite using GridSearchCV to find optimal parameters, further adjustments to max_depth and n_estimators could be explored. Testing with values higher or lower than 400 and 60, respectively, may yield better results. Moreover, due to imbalanced job position requirements, it is essential to perform data balancing to improve the model's performance.

Afterwards, the trained model was deployed via an application programming interface (API) that connected the functionality between the API, database, and job recommendation page. The feature helped recommend job categories based on skills. Initially, no data was displayed in the results and dashboard until the user selected at least 4 skills, based on the analysis of the overall demand for the digital labor market in Thailand. When testing the skills system with at least four skills, job groups were clearly separated. The list of skills for users to choose from was derived from all digital skills data, allowing job posting information and the relationship between job groups and skills to be displayed. Once the skills are selected, the system sends the chosen skills to the API, which then calls the model with the skill data. When the model receives the skill information, it returns the appropriate job categories to the API (Surendar et al., 2024), which in turn sends the results to the dashboard on the webpage. In analyzing the overall demand for digital workforce skills in Thailand, testing with at least 4 selected skills can clearly separate job categories. The skill options provided to users are drawn from the complete set of digital skills stored in the database. Once the analysis is complete, the job categories that best match the selected skills are shown. The dashboard will then display relevant data, including job postings and the relationship between job categories and skills (Wu, 2024).

4. Results

4.1 Model Performance

The job recommendation model matches users with 1 of the 20 job categories based on their selected skills. Performance testing using K-Fold Cross Validation compared Decision Tree and Random Forest models. The Decision Tree model showed lower accuracy compared to Random Forest. The Random Forest model achieved superior performance with 75.00% accuracy, outperforming the Decision Tree approach, as shown in Table 8.

The performance evaluation using k-fold cross validation revealed that the Decision Tree model achieved an accuracy of 71.00%, precision of 80.00%, recall of 66.00%, and an F1-Score of 71.00%, as summarized in Table 9. Additionally, the confusion matrix and the ROC curve are presented in Figure 6.

Table 8 Algorithm Efficiency Comparison Results

Algorithms	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision tree	71.00	80.00	66.00	71.00
Random forest	75.00	87.00	72.00	77.00
Support Vector Machine (SVM)	4.39	3.00	4.00	4.00
K-Nearest Neighbors (k-NN)	38.42	40.00	40.00	41.00

Table 9 Classification Report on Decision tree

Classification Report	Precision	Recall	F1-score	Support
.net-developer/programmer	0.84	0.64	0.73	25
back-end-developer	0.74	0.45	0.56	31
business-analyst	0.55	0.37	0.44	49
data-analyst	0.92	0.71	0.80	17
data-engineer	0.93	0.76	0.84	17
database-administrator	0.71	0.60	0.65	20
developer/programmer	0.79	0.76	0.78	204
devops	0.77	0.83	0.80	12
front-end-developer/programmer	0.65	0.68	0.67	22
full-stack-developer/programmer	0.79	0.70	0.74	37
it-support	0.58	0.91	0.71	247
java-developer/programmer	0.71	0.76	0.73	29
mobile-developer/programmer	1.00	0.94	0.97	33
network-engineer	0.95	0.81	0.88	26
project-manager	0.76	0.43	0.55	67
software-engineer	0.71	0.36	0.48	33
system-engineer	0.71	0.36	0.48	33
tester	1.00	0.50	0.67	44
ux/ui-designer	0.82	0.45	0.58	20
web-developer/programmer	0.81	0.85	0.83	34
accuracy			0.71	1000
macro avg	0.80	0.66	0.71	1000
weighted avg	0.75	0.71	0.71	1000

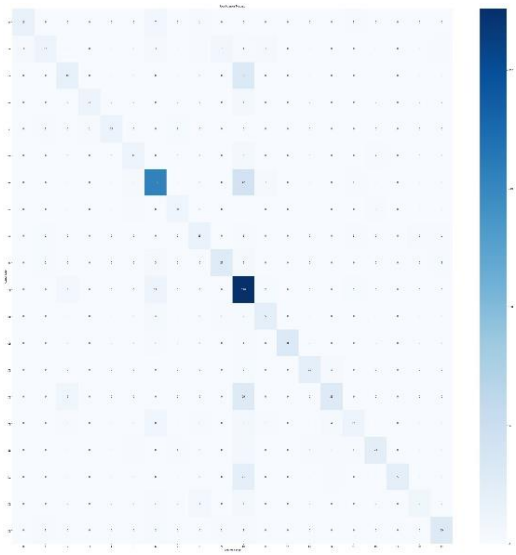

Figure 6 Confusion Matrix of Decision tree

The ROC curve of the Decision Tree model demonstrates a moderate classification capability. The area under the curve (AUC) is relatively low, indicating limitations in distinguishing between complex or imbalanced classes. Although the model shows decent precision, the recall remains low, which affects the overall accuracy. This is illustrated in Figure 7.

The performance test results using k-fold cross validation show that the Random Forest model outperforms the Decision Tree model by 4.00%. The Random Forest model achieved an accuracy of 75.00%, precision of 87.00%, recall of 72.00%, and F1-Score of 77.00 in classifying job categories, as shown in Table 10 and Figure 8.

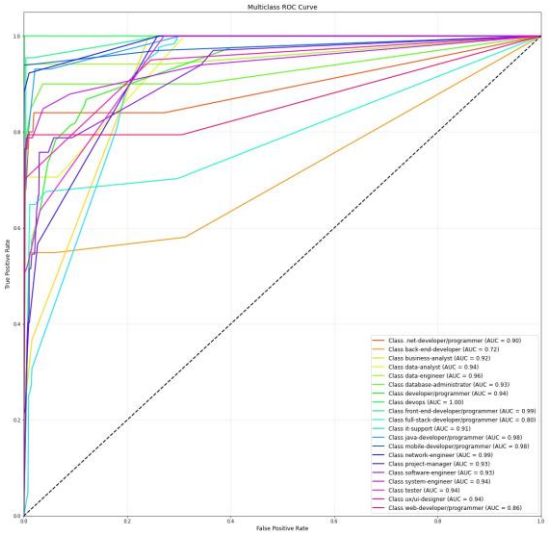

Figure 7 ROC Curve of Decision tree

Table 10 Classification Report of Random Forest

Classification Report	Precision	Recall	F1-score	Support
.net-developer/programmer	0.92	0.48	0.63	25
back-end-developer	0.89	0.52	0.65	31
business-analyst	0.55	0.37	0.44	49
data-analyst	0.93	0.82	0.87	17
data-engineer	1.00	0.94	0.97	17
database-administrator	1.00	0.70	0.82	20
developer/programmer	0.79	0.81	0.80	204
devops	0.92	1.00	0.96	12
front-end-developer/programmer	0.71	0.91	0.80	22
full-stack-developer/programmer	0.78	0.76	0.77	37
it-support	0.60	0.91	0.72	247
java-developer/programmer	0.77	0.79	0.78	29
mobile-developer/programmer	1.00	0.94	0.78	29
network-engineer	0.92	0.88	0.90	26
project-manager	0.83	0.43	0.57	67
software-engineer	0.83	0.45	0.59	33
system-engineer	1.00	0.76	0.59	33
tester	1.00	0.50	0.67	44
ux/ui-designer	1.00	0.60	0.75	20
web-developer/programmer	0.80	0.82	0.85	34
accuracy			0.75	1000
macro avg	0.87	0.72	0.77	1000
weighted avg	0.78	0.75	0.74	1000

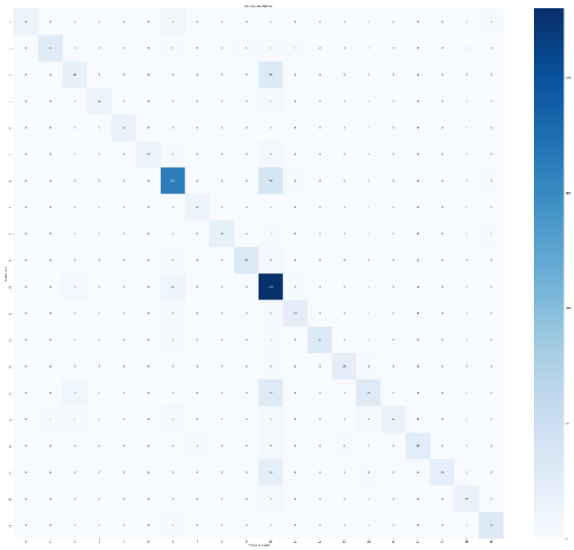

Figure 8 Confusion Matrix of Random Forest

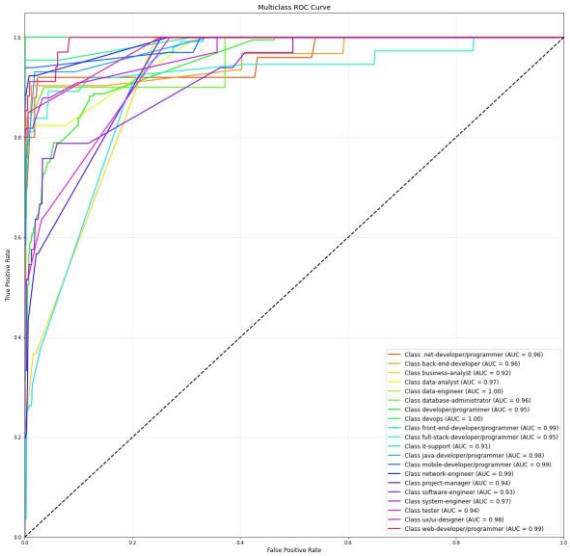

Figure 9 ROC Curve of Random Forest

The ROC curve of the Random Forest model clearly shows superior classification performance compared to the Decision Tree. It has a higher area under the curve (AUC), reflecting the model's enhanced ability to distinguish between class groups. The ensemble learning approach, which combines predictions from multiple trees, reduces overfitting and improves the model's stability when predicting new data. This is illustrated in Figure 9.

4.2 Real-world Application

After training the job recommendation models, we selected the Random Forest model, which demonstrated superior performance, for deployment. The model was deployed on a web server with an API interface connecting the front end to the model (). Users can specify a minimum of 4 skills, which the web application sends as a request to the job recommendation model. The model then returns recommended job positions that correlate with the user's input skills. This process follows the algorithm outlined in Algorithm 1, with recommendation results displayed as shown in Figure 10.

Algorithm 1 Recommendation System Algorithm

```
Start
    Input:
      List<skills>, DigitalSkillsDatabase

    do SkillMatchingAlgorithm()

    Function: SkillMatchingAlgorithm()
      Initialize job_groups as an empty list
      If length of skills < 4
        Display "Please select at least 4 skills"
        Return {}
      End If

      For each skill in skills:
        Search the DigitalSkillsDatabase for matching job roles
        Add matching job roles to job_groups
      End For

      AnalyzeRelationship(skills, job_groups)
      Return {
        List<job_groups>
      }
    EndFunction

    Output: do DisplayResultsOnDashboard(List<job_groups>)
End
```

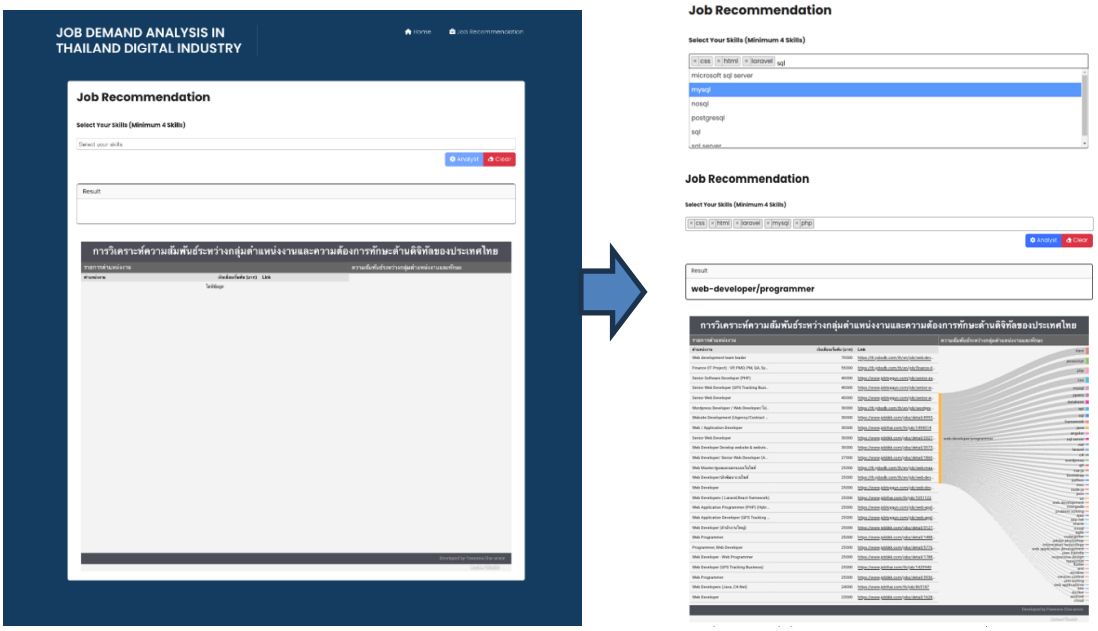

Figure 10 Job Recommendation Page

4.3 System Testing

The usability test results for the dashboard and web application pages, aimed at analyzing the overall demand for digital labor skills in Thailand, including the ease of use and understanding of the workflow for the job recommendation function based on suitable skills, involved a total of 7 testers. It was found that the participants were most satisfied with the website navigation menu and the dashboard on the home page, with a satisfaction rate of 97.00%. Meanwhile, satisfaction with the job recommendations provided by the system, which matched the selected skills, was the lowest at 88.60%. Overall satisfaction with the website user experience was 91.40%. Participants also commented on the display of links to job announcement pages, suggesting that only links to job postings that were still open should be shown. Additionally, the skill criterion could be specified as either all selected skills or any one of the selected skills.

5. Conclusion

This research presents the development and application of a machine learning-based system for digital job matching in Thailand, focusing on skill-based alignment between job seekers and labor market demands. The study successfully integrates data analytics, machine learning techniques, and an interactive dashboard to provide comprehensive insights into the current state of the digital labor market. These insights include job distribution by region, skill requirements by occupation, and salary trends supporting both macro-level workforce planning and micro-level career decision-making.

The system demonstrates that machine learning can effectively analyze large-scale job posting data and extract meaningful patterns, particularly in identifying the relationship between job categories and required skills. The dashboard not only facilitates real-time exploration of labor market trends but also supports targeted filtering by job group, skill set, and location, thereby enhancing its utility for diverse user groups-policymakers, educators, employers, and job seekers alike.

The job recommendation model, built upon users' self-identified skill profiles, shows practical value in guiding individuals toward suitable job categories. Although the model exhibits limitations-such as imbalanced data, inconsistent job classifications, and coverage constraints-it nonetheless proves the feasibility of automated skill-based job matching in dynamic and complex labor market environments.

From an academic standpoint, the study contributes to the growing body of knowledge on labor market analytics and intelligent recommendation systems. The approach demonstrates how machine learning can be applied not only for classification and prediction but also as a strategic tool to support real-world decision-making in human resource development.

In terms of policy impact, the insights generated from the system can inform curriculum development, training programs, and regional workforce planning. By identifying gaps between supply and demand of digital skills across provinces and job categories, this research provides evidence-based support for formulating targeted and responsive labor policies. The adaptability of the system architecture further allows for localization and expansion into other sectors or regions through retraining with context-specific datasets.

Looking ahead, several recommendations are proposed for future development and research; 1) Enhance data collection and integration. Future systems should incorporate mechanisms to automatically harvest real-time job posting data, ensuring timeliness and completeness of labor market intelligence. Expanding the dataset to include job positions across all economic sectors will allow for a more holistic understanding of national workforce demands. 2) Improve model performance and robustness. Applying advanced machine learning techniques such as gradient boosting, deep learning, and ensemble methods will help address limitations caused by imbalanced and incomplete datasets. Incorporating NLP models like BERT can also improve understanding of unstructured job descriptions and skill tags. 3) Address data imbalance and labeling challenges, techniques such as SMOTE, class weighting, and semi-supervised learning can help overcome the issues of class imbalance. Automating the job classification process using clustering or weak supervision methods may reduce human bias and improve scalability. 4) Align with occupational standards. Future development should align the job classification framework with national and international standards (e.g., NSO, ISCO-08) to ensure consistency, comparability, and policy interoperability and 5) Expand personalization and recommendation capabilities. Enhancing the job recommendation module to provide ranked and diversified job group suggestions based on skill similarity and market

demand will increase the value of the system for end users.

In conclusion, this study demonstrates the value of applying machine learning to support digital workforce development. The system offers a scalable and adaptable solution that bridges the gap between educational outcomes and labor market needs, contributing not only to academic advancement but also to practical applications in workforce planning, career guidance, and policy formulation in the digital economy.

6. CRediT Statement

Paweena Chaiaroon: Conceptualization, methodology, software, formal analysis, investigation, data curation. **Nuttachot Promrit**: Methodology, software, formal analysis, resources, writing – original draft, writing – review & editing, supervision, funding acquisition. **Karanya Sitdhisanguan**: Software, validation, visualization.
Sajjaporn Waijanya: Conceptualization, methodology, validation, resources, writing – original draft, writing – review & editing, supervision, project administration. **Natratanon Kanraweekultana**: Validation, investigation, data curation, writing – original draft, writing – review & editing, visualization.

7. Abbreviations

AI	Artificial Intelligence
NLP	Natural Language Processing
ICTs	Information and Communication Technologies
RAKE	Rapid Automatic Keyword Extraction
SVM	Support Vector Machine
k-NN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
API	Application Programming Interface
AR	Augmented Reality
VR	Virtual Reality
MR	Mixed Reality
UI	User Interface
UX	User Experience

8. References

Alghamlas, M., & Alabduljabbar, R. (2019). Predicting the suitability of IT students' skills for the recruitment in Saudi labor market [Conference presentation]. *The 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*, Riyadh, Saudi Arabia. https://doi.org/10.1109/CAIS.2019.8769577

Anthony, G. S. (2024). *Developing a framework to identify professional skills required for banking sector employee in UK using Natural Language Processing (NLP) techniques* [Doctoral dissertation], The University of Salford, Manchester, United Kingdom.

Bhushan, N., Mekhilef, S., Tey, K. S., Shaaban, M., Seyedmahmoudian, M., & Stojcevski, A. (2024). Dynamic K-decay learning rate optimization for deep convolutional neural network to estimate the state of charge for electric vehicle batteries. *Energies*, *17*(16), Article 3884. https://doi.org/10.3390/en17163884

Chaudhary, A., Kolhe, S., & Kamal, R. (2016). An improved random forest classifier for multi-class classification. *Information Processing in Agriculture*, *3*(4), 215-222. https://doi.org/10.1016/j.inpa.2016.08.002

Dalal, A., Rayan, R., Barua, A., Vasserman, E. Y., Sarker, M. K., & Hitzler, P. (2024). On the value of labeled data and symbolic methods for hidden neuron activation analysis [Conference presentation]. *International Conference on Neural-Symbolic Learning and Reasoning*. Cham, Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-71170-1_12

Dehghani, M., & Manthouri, M. (2021). Semi-automatic detection of Persian stopwords using FastText library [Conference presentation]. *The 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, Mashhad, Iran. https://doi.org/10.1109/ICCKE54056.2021.9721519

Fuzul, E., & Horvat, M. (2019). Formal model of student competencies in higher education and required skills in the job market [Conference presentation]. *The 2019 Central European Conference on Information and Intelligent Systems (CECIIS)*, Varaždin, Croatia.

Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & Seveso, A. (2021). Skills2Job: A recommender system that encodes job offer embeddings on graph databases. *Applied Soft Computing*, *101*, Article 107049. https://doi.org/10.1016/j.asoc.2020.107049

Hakim, L., Sari, Z., Aristyo, A. R., & Pangestu, S. (2024). Optimizing android program malware classification using GridSearchCV optimized random forest. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *9*(2), 173-180. https://doi.org/10.22219/kinetik.v9i2.1944

Hardeniya, N., Perkins, J., Chopra, D., Joshi, N., & Mathur, I. (2016). *Natural language processing: Python and NLTK*. Birmingham, UK: Packt Publishing Ltd.

Hassan, M. U., Alaliyat, S., Sarwar, R., Nawaz, R., & Hameed, I. A. (2023). Leveraging deep learning and big data to enhance computing curriculum for industry-relevant skills: A Norwegian case study. *Heliyon*, *9*(4). Article e15407. https://doi.org/10.1016/j.heliyon.2023.e15407

Htet, T. S., & San, K. K. (2024). Transforming tech hiring: BERT-powered neural networks for automated job title classification [Conference presentation]. *The 2024 5th International Conference on Advanced Information Technologies (ICAIT)*, Yangon, Myanmar.

Ibadov, I., Aksenov, A., Iumanova, I., & Sozykin, A. (2020). The concept of a dynamic model of competencies for the labor market analysis [Conference presentation]. *The 2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, Ekaterinburg, Russia. https://doi.org/10.1109/USBEREIT48449.2020.9117691

Ingole, A., Wadurkar, A., Chaudhari, S., & Chavhan, B. (2024). Skill-based job role suggestion using machine learning. *International Journal of Ingenious Research, Invention and Development*, *3*(2), 86-94. https://doi.org/10.5281/zenodo.11034674

Kanraweekultana, N., Waijanya, S., Promrit, N., Nopnapaporn, U., Korsanan, A., & Poolphol, S. (2024). Comparison of capability of data classification models to predict consistent results for depression analysis based on user-behaviour tracking and facial expression recognition during PHQ-9 assessment. *Engineering and Applied Science Research*, *51*(1), 11-21. https://doi.org/10.14456/easr.2024.2

Mahlich, C., Vente, T., & Beel, J. (2024). From theory to practice: Implementing and evaluating e-fold cross-validation. *arXiv preprint arXiv:2410.09463*. https://arxiv.org/abs/2410.09463

Melo, G., Chaves, M., Kolter, M., & Schleifenbaum, J. H. (2023). Skills requirements of additive manufacturing-a textual analysis of job postings using natural language processing [Conference presentation]. *The 2023 International Conference on Additive Manufacturing in Products and Applications*, Cham, Switzerland.

Ministry of Higher Education, Science, Research and Innovation. (2024). *Higher education statistics*. Retrieved from https://info.mhesi.go.th (In Thai)

Office of the National Digital Economy and Society Commission. (2019). *National policy and plan on digital development for the economy and society.* Retrieved from https://datacatalog.bde.go.th/dataset/eb2e0e52-adc5-4f0c-aca1-96358b79ce32 (In Thai)

Office of the National Economic and Social Development Council. (2017). *Thai social conditions in the first quarter of 2017.* Retrieved from https://www.m-society.go.th/ewtadmin/ewt/mso_web/article_attach/21484/21278.pdf (In Thai)

Phaphuangwittayakul, A., Saranwong, S., Panyakaew, S. N., Inkeaw, P., & Chaijaruwanich, J. (2018). Analysis of skill demand in Thai labor market from online jobs recruitments websites [Conference presentation]. *The 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Nakhon Ratchasima, Thailand. https://doi.org/10.1109/JCSSE.2018.8457393

Pias, S. A., Hossain, M., Rahman, H., & Hossain, M. M. (2024). Enhancing job matching through natural language processing: A BERT-based approach [Conference presentation]. *The 2024 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, Dhaka, Bangladesh. https://doi.org/10.1109/ICISET62123.2024.10939860

Pillai, P., & Amin, D. (2020). Understanding the requirements of the Indian IT industry using web scrapping. *Procedia Computer Science*, *172*, 308-313. https://doi.org/10.1016/j.procs.2020.05.050

Pundir, R. S., Dhasmana, A., Karakoti, U., Sikder, A., Sharma, S., & Manchanda, M. (2024). Enhancing resume recommendation system through skill-based similarity using deep learning models [Conference presentation]. *The 2024 International Conference on Inventive Computation Technologies (ICICT)*, Lalitpur, Nepal. https://doi.org/10.1109/ICICT60155.2024.10544875

Qin, C., Zhu, H., Xu, T., Zhu, C., Ma, C., Chen, E., & Xiong, H. (2020). An enhanced neural network approach to person-job fit in talent recruitment. *ACM Transactions on Information Systems (TOIS)*, *38*(2), 1-33. https://doi.org/10.1145/3376927

Rahhal, I., Carley, K. M., Kassou, I., & Ghogho, M. (2023). Two stage job title identification system for online job advertisements. *IEEE Access*, *11*, 19073-19092. https://doi.org/10.1109/ACCESS.2023.3247866

Senger, E., Zhang, M., Van Der Goot, R., & Plank, B. (2024). Deep learning-based computational job market analysis: A survey on skill extraction and classification from job postings. *arXiv preprint arXiv:2402.05617*. https://doi.org/10.48550/arXiv.2402.05617

Surendar, V., Sriram, E., & Subhash, S. (2024). Web-based deep learning model for zero day vulnerability detection using FastAPI [Conference presentation]. *The 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, Chennai, India. https://doi.org/10.1109/ADICS58448.2024.10533540

Weichselbraun, A., Süsstrunk, N., Waldvogel, R., Glatzl, A., Braşoveanu, A. M., & Scharl, A. (2024). Anticipating job market demands-A deep learning approach to determining the future readiness of professional skills. *Future Internet*, *16*(5), Article 144. https://doi.org/10.3390/fi16050144

Wu, X. (2024). *FastAPI as a backend framework* [Bachelor Thesis]. Tampere, Tampere University, Finland.

Yahya, A. E., Yafooz, W. M., & Gharbi, A. (2024). Mapping graduate skills to market demands: a holistic examination of curriculum development and employment trends. *Engineering, Technology & Applied Science Research*, *14*(4), 14793-14800. https://doi.org/10.48084/etasr.7454