Job Resumes Recommendation using Integration of Fuzzy Discernibility Matrix Feature Selection and Convolutional Neural Network Multi-label Text Classification

Regiolina Hayami[1,2*], Nooraini Yusoff[2], Kauthar Mohd Daud[3], Yulia Fatma[1]
[1] Department of Informatics Engineering, Faculty Computer Science, Universitas Muhammadiyah Riau, Indonesia
[2] Faculty of Data Science and Computing, Universiti Malaysia Kelantan, 16100, Malaysia
[3] Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600, Malaysia
E-mail: regiolinahayami@umri.ac.id
*Corresponding author

Keywords: automation resume content-based recommendation, fuzzy discernibility matrix, feature selection, CNN, multi-label text classification

Received: August 1, 2024

The manual analysis of job resumes poses specific challenges, including the time-intensive process and the high likelihood of human error, emphasizing the need for automation in content-based recommendations. Recent advancements in deep learning, particularly Convolutional Neural Networks (CNN) for Multi-label Text Classification (MLTC), offer a promising solution for addressing these challenges through artificial intelligence. While CNN is renowned for its robust feature extraction capabilities, it faces specific challenges such as managing high-dimensional data and poor data interpretability. To address these limitations, this study employs the Fuzzy Discernibility Matrix (FDM) feature selection technique to determine the relevance of skills for each job vacancy. FDM assigns weights to the features of each job category and ranks their relevance based on the highest scores, which are then utilized in the MLTC CNN model. The integration of FDM and MLTC CNN serves as the foundation for generating content-based recommendations derived from key features in job resumes. This study produces a content-based job recommendation system that displays the top three job categories, accompanied by explanations of the skills supporting each selected category. These recommendations also consider cosine similarity values between analyzed items and the integrated results of FDM and MLTC CNN. The application of FDM for feature weighting has been proven to enhance multi-label text classification outcomes by providing better insights during the feature selection process. With a recall of 97.26%, precision of 94.81%, and accuracy of 98.58%, the MLTC model integrating FDM feature selection and CNN demonstrates robust performance characteristics in content-based job recommendation tasks.

Povzetek: Izvirni sistem za priporočanje življenjepisov za zaposlitev združuje izbiro značilnosti z matriko fuzzy razločljivosti (FDM) in večoznačno klasifikacijo besedila s konvolucijsko nevronsko mrežo (CNN). FDM izboljša izbiro relevantnih veščin, CNN pa zagotavlja robustno klasifikacijo.

1 Introduction

Currently, up-to-date job vacancy information can be easily accessed through job listing websites. In addition to providing vacancy information, job seekers can also submit applications directly through these websites. Companies can receive a large number of resumes more quickly, but this can lead to time-consuming manual analysis. A recommendation system can automate this process by filtering data and providing recommendations for candidates who meet the company's criteria. This system offers relevant content according to user needs and can provide recommendations or predictions of user interests [1], [2]. Traditionally, recommendation systems use two approaches: Collaborative Filtering (CF) and Content-Based Filtering (CBF). CF identifies preference patterns based on an analysis of user community data, while the CBF approach is based on consideration of previous user preferences and is represented based on the content features they want to recommend. In the learning process of feature representation, CBF only focuses on the similarity of text to produce a recommendation[3]. Nevertheless, CBF has several drawbacks. CBF cannot generate suitable suggestions if the content analyzed for an item does not contain appropriate information for categorization and scalability and sparsity issues can also occur when using large amounts of data[4]. To address this limitation, there needs to be learning related to the features of the items to be recommended[5], [6] which assigns different levels of importance to different features.

Multilabel text classification (MLTC) is a field of natural language processing (NLP) that can be applied to information retrieval [7] and tag recommendation [8], [9]. MLTC works by assigning multiple labels to each sample in a dataset based on their interrelationships. In job classification, MLTC helps categorize individuals based on various skills, enhancing the efficiency of talent acquisition processes.

The advent of deep learning techniques as a new area of computer science for handling recommendation scenarios is one development in neural networks [10]. In terms of feature extraction, the application of deep learning techniques to assist in determining the outcomes of recommendations has increased [5], [11]. Originally created for image processing, Convolutional Neural Networks (CNN) have demonstrated exceptional effectiveness in adapting to text classification problems [8], [12], [13]. The evaluation matrix provides encouraging findings. CNN has outperformed conventional machine learning algorithms in text categorization thanks to its capacity to extract relevant features from input data.

Although CNNs show promising results in performing text classification tasks, there are weaknesses in their processing, particularly in single-class and multi-class classification cases. CNNs face challenges in extracting or displaying how each embedding matrix is related [14], [15]. While CNNs can achieve the best evaluation metrics in MLTC, the contribution of each feature to the value predicted by the classifier must be examined and calculated [16], [17], [18]. Feature selection reduces the training time and storage required. A long list of features, statistically sorted based on their distinctiveness for each class, is produced from feature selection methods [19]. Representative features are those that have the highest values. Therefore, feature selection that filters pertinent features is an essential step in reducing data dimensionality and improving learning performance.

There are two primary obstacles with integrating job resume data into CNN algorithms and feature selection methods. First off, because CNN is a deep learning method, it is frequently used in image classification scenarios, which means that the network architecture must be modified in order to handle text-form input. Second, feature selection presents a discrete space combinatorial optimization problem that calls for a customized coding scheme, crossover operators, and mutation. This study suggests incorporating the Fuzzy Discernibility Matrix (FDM) feature selection method into the CNN architecture for MLTC in order to address these issues. Our contributions include feature reduction by FDM feature selection, smooth integration of feature selection into CNN architecture, and building a multi-label classification model on job resume text using CNN.

As the final result of this study, we will examine how the FDM feature selection method performs well in multi-label text classification using the CNN algorithm. Subsequently, we will explain our approach and teamwork along with its connection to talent development work. Utilizing critical analysis and empirical data, the effectiveness of the methods designed to improve CNN's performance in the talent acquisition process will be revealed.

2 Related works

2.1 CNN Multi-label classification

Although CNN was initially created for image processing, text processing has since made extensive use of it. In 2014, Kim suggested 1D-CNN, a CNN adaption for text categorization that uses convolutional filters and max-pooling filters that only slide on one dimension (the y dimension) [20]. CNN has demonstrated its ability to analyse natural language in recent years and has also been utilised as a model for sentence classification [21]. Maximum accuracy has been attained in feature extraction from text data using Convolutional Neural Networks with Multi-label Classification, a crucial component in producing recommendations [8].

Multilabel Text Classification (MLTC) is an important task in the field of natural language processing (NLP), which can be applied in many real-world scenarios, such as information retrieval[7], tag recommendation[8], [9] and so on. In recent years, neural networks have achieved great success in many fields including NLP. Some neural network models have also been applied in the MLTC task and achieved important progress, especially research on the use of CNN in multi-label text classification[8], [12], [13] shows good results on the evaluation matrix.

However, it is necessary to analyze and calculate the contribution of classification results/class decisions in grading the recommended object in which this is still lacking in existing studies of deep learning-based recommendation system.

2.2 Review on current study in content-based recommendation

Various studies related to Content-Based Filtering (CBF) as an approach in recommendation systems are widely used in providing recommendations for an item with associated textual content. Traditional content-based filtering has weaknesses in terms of extraction and learning of data to produce recommendations. Further developments have been made to improve learning for content-based recommendations using machine learning algorithms to the application of deep learning algorithms with multi-label classification to improve data learning accuracy. The following table represents several years of research on content-based recommendation with text form data:

Table 1: Review on current method in content-based recommendation

Methods	Remarks/Gap
Hybrid Filtering (Content-based + Collaborative Filtering) [5], [6], [22]	Text extraction is suboptimal, lacking feature analysis by focusing only on item similarity for recommendations.
Chi-square + Softmax regression [3], K-means, Cosine Similarity [23]	Low accuracy, recall, and F-measure values persist
Hybrid Filtering + DNN [24], [25], [26]	Weak item filtering; requires algorithm optimization, parameter tuning, and improved pre-processing for better recommendations.
CNN + Bayesian approach [27], [28], [29]	Requires additional information and improved classification performance for better recommendations.
CNN + LSTM [7], [30], [31], [32]	Low system efficiency delays recommendations; optimize relevance features and deep learning integration for contextual decisions.

2.3 Feature selection using fuzzy discernibility matrix

One of the generalizations of the fuzzy rough set that has garnered a lot of interest lately is the fuzzy discriminability matrix. The Fuzzy Discernibility Matrix (FDM) is a fuzzy version of the classic decision-relative discernibility matrix [33]. A fuzzy discernibility matrix (FDM) is a representation of the fuzzy discernibility relation. When employing fuzzy rough discernibility matrices for multi-label data processing, two key issues must be resolved. One is how to use the discernibility matrix to build algorithms, and the other is how to extract correlations at the sample and label levels [34].

Using fuzzy rough set which has been proven to give good results in the case of deep learning. Research on fuzzy and deep learning for classification has been carried out in various cases, including classification of over-lapped data [35], feature selection on neural network and integration between deep CNN and fuzzy rough set for image classification [36].

Based on related works, it can be concluded that learning development has been conducted on content-based recommendations using hybrid techniques and classification through machine learning and deep learning models. However, improvements in feature analysis are necessary to generate more relevant recommendations. In addition to enhancing classification performance, feature relevance analysis also aids in improving the efficiency of the recommendation process. Therefore, this study develops learning using CNN by incorporating a feature selection technique, which is part of feature engineering, to reduce the number of features per class.

3 Method

From preprocessing to evaluation, this study was carried out in phases. The gathered data will undergo pre-processing in order to be transformed into a matrix. The CNN algorithm then uses the matrix-formatted data as input data to produce a multi-label classification. Additionally, a content-based recommendation algorithm is used to calculate data in the form of a matrix. so that the system can forecast the talents that each resume will include and deliver the final results of resume recommendations. Figure 1 below shows the completed framework:

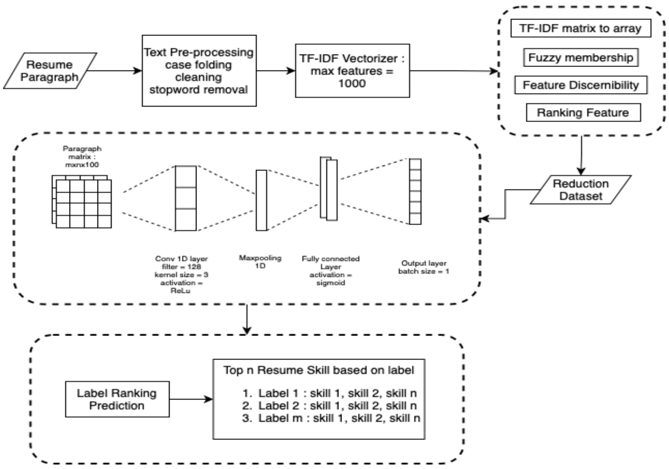

Figure 1: Flow of process integration of FDM and CNN MLTC for Job Resume Content-based Recommendation

3.1 Preprocessing

At this stage the data will enter the preprocessing stage. This stage will convert resume data and job descriptions into a matrix before being used in CNN classification and calculating content-based recommendations at a later stage. Preprocessing consists of case folding, data cleaning and tokenization. At case folding's stage, the text data is standardized by converting each word to lowercase to ensure consistency and equal treatment of all words. In data cleaning, the entire text data is cleaned by removing all attributes such as numbers and punctuation marks. After that, in step of tokenization all sentences are split into tokens using white space, and punctuation marks or any symbols other than letters are removed. Here is the example of data preprocessing:

Before:

experience python developer, using python

After preprocessing:

experience, developer, python

For example, there is a sentence described with "before." After preprocessing (case folding, cleaning, and tokenization), three words representing the sentence will be generated, as explained in the "after" section.

Before the data proceeds to the feature selection stage, word vectorization is first carried out using TF-IDF. To calculate the TF-IDF value of the words contained in the data, the steps are [37]:

- Calculate the TF value for each word in each sentence with the formula:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad (1)$$

Where, ni,j: number of occurrences of entry wi

- Calculate the IDF value for each word with a formula:

$$IDF = \log\left(\frac{|D|}{|\{j: w_i \in x_j\}|+1}\right) \quad (2)$$

Where, |D|: total number of files in the corpus

|{j : wi ∈ xj}| = number of files containing entry wi

- Compute the TF-IDF of both sentences:

$$TF - IDF_{wi} = TF_{ij} \times IDF_i \quad (3)$$

3.2 Fuzzy discernibility matrix (FDM) feature selection

The fuzzy-rough disparity matrix represents the relationship of fuzzy-rough disparity. Attribute reduction aims to identify a subset of features that can enhance the distinction between objects from different classes. Feature selection is used to generate a reduced dataset before it is fed into the learning process using CNN. FDM is constructed to describe the extent to which objects in the set U differ from each other concerning the features provided in set C. A fuzzy discernibility matrix M = (M(x, y)) is constructed on the universe U, whose size is |U| × |U|[38]. In algorithm 1, FDM feature selection is proposed and each process of the algorithm can be explained as follows:

Algorithm 1. Fuzzy Disernibility Matrix(FDM) feature selection

```
Input: tfidf_matrix T: transpose of tfidf matrix;
tfidf.get_feature_names_out(): tfidf features name
Output: indices: feature score
Begin
for i = 0 to len(tfidf.get_feature_names_out()) - 1:
   for j = i + 1 to len(tfidf.get_feature_names_out()) - 1:
   fdm[i, j] =
fuzz.interp_membership(tfidf_matrix_T[i],
tfidf_matrix_T[j], np.min(tfidf_matrix_T[j]))
fdm[j, i] = fdm[i, j]
feature scores = np.sum(fdm, axis=1)
End
```

In Algorithm 1, FDM feature selection calculates the ranking or score of features based on the fuzzy indiscernibility matrix of the transposed TF-IDF matrix. The algorithm uses a nested loop to iterate through each feature pair (i, j) where i < j. In each iteration, the value of fdm[i, j] is calculated using a fuzzy membership function to determine the level of indiscernibility between two features based on their TF-IDF values. The `fuzz. interp_membership` function is used to compute the membership of the fuzzy set, measuring how similar or dissimilar the TF-IDF values of features i and j are. `np.min(tfidf_matrix_T[j]` is used to get the minimum value from the TF-IDF value vector for feature j, which is used as a parameter to determine membership. The final result of this algorithm is `feature_scores`, which are the scores or rankings of features based on their contribution to distinguishing objects based on their TF-IDF values. The use of the Fuzzy Indiscernibility Matrix allows for a more nuanced evaluation of the differences between these features in the context of data analysis.

3.3 CNN Multi-label text classification

At this point, a multi-label classification model is created using CNN modelling. Later on, each resume's talents will be classified using this model. Resumes are categorized using CNN based on a single class. In other words, CNN classification is designed to forecast each class, so that the number of classes in the resume data is equal to the number of original classifications. The CNN architecture used for multi-label classification is shown in figure 2:

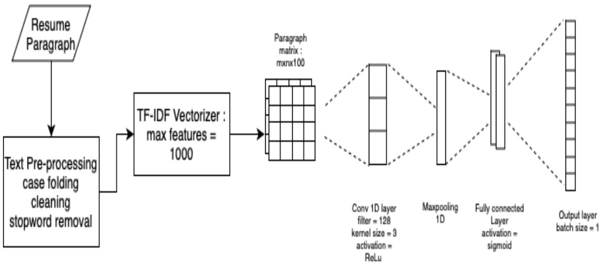

Figure. 2: CNN's Architecture for multi-label text classification

To form a training set from each basic classification result, the original multi-label data set is divided into n single-label data sets (where n is the total number of classes in the original data set). Each sub-dataset generated corresponds to a binary classification problem that focuses on each class. The process of creating the CNN model will be built using a Python library. Just like in the previous modeling process, CNN consists of three layers, namely, the convolutional layer, the pooling layer, and the output layer. In this study, the hyperparameters of the CNN architecture used, such as the number of filters, kernel size, and activation function, have been configured. This CNN architecture includes 128 Conv1D filters, a kernel size of 3, the Adam optimizer, ReLU activation function, accuracy as the metric, and 10 epochs. The selection of these hyperparameters is essential to optimize the model's ability to capture patterns and features useful for multi-label classification.

3.4 Content-based recommendation

The integration of FDM and CNN multi-label text classification for content-based recommendation is implemented through utilizing the generated model to predict top-n recommendations by analyzing the text data. In content-based recommendation, the recommendations are provided by exploring the contents of user profiles, product descriptions, or other factors related to the formation of user choices for an item. Content-based recommendation is based on calculating the similarity between items, with values approaching 1 indicating a high degree of similarity. The cosine similarity formula is used to obtain this value, as follows [39]:

$$Cosine\ Similarity = \frac{\sum_{i=1}^{n} A_i \ X\ B_i}{\sqrt{\sum_{i=1}^{n} {A_i}^2} X \sqrt{\sum_{i=1}^{n} {B_i}^2}} \quad (4)$$

Where: Ai = The feature value of the item to be recommended; Bi = The value of the user's profile / preferences.

In this study, the cosine similarity value is calculated by matching the feature values of the items resulting from the integration of FDM and CNN multi-label text classification with the TF-IDF vector values of the words that will predict the top-n class label recommendations. Then, the feature values of the words to be predicted are matched with the features representing the selected top-n classes. As the final result, this system generates a list of the top dominant features for each predicted label. These features are used as the basis for the content-based recommendation system, where labels (e.g., 1, 2, 3, etc.) are associated with a list of dominant features (feature 1, feature 2, up to feature n) that have the greatest influence on the prediction.

3.5 Evaluation

At this stage, performance measurement is carried out using several criteria such as accuracy and performance. Three measures are generally used in the literature to evaluate a multi-label classifier[40]:

The accuracy of the multi-label architecture is then calculated as the average of the accuracies found for all cases. The accuracy is determined by the formula:

$$A = \frac{1}{n} \sum_{i=1}^{n} \frac{|A_i \cap Z_i|}{|A_i \cup Z_i|} \quad (5)$$

The precision is the proportion of labels correctly predicted by the model out of all the labels anticipated for that instance. The precision is determined by the formula:

$$A = \frac{1}{n} \sum_{i=1}^{n} \frac{|A_i \cap Z_i|}{|Z_i|} \quad (6)$$

The recall for a given instance is the ratio of accurately predicted labels to all actual labels for that instance. The average of the recalls assessed across all occurrences is the overall recall. The formula is used to determine the recall:

$$A = \frac{1}{n} \sum_{i=1}^{n} \frac{|A_i \cap Z_i|}{|A_i|} \quad (7)$$

4 Result and discussion

The dataset consists of 28,707 resumes collected from the Indeed.com website and distributed into ten classes [8]. The IT courses indicated on the resume include those for project manager, database administrator, security analyst, system administrator, front-end developer, network administrator, web developer, Python developer, and Java developer. Figure 3 displays the distribution of the dataset:

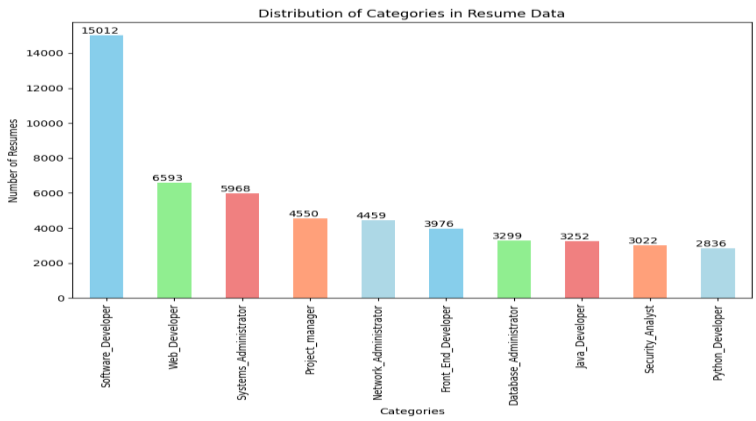

Figure 3: Data Resume Distribution based on Skill

The uniqueness of this dataset lies in its labels, which represent one or more categories for each data point, making it a multi-label dataset. The data distribution shows that 11,933 data points have 1 label, 11,190 have 2 labels, 5,089 have 3 labels, 735 have 4 labels, 72 have 5 labels, 11 have 6 labels, and 3 have 7 labels. This poses a unique challenge in building a multi-label classification model using CNNs that can handle data imbalance and label distribution effectively.

Before the training process using CNN, feature scoring is first performed using the FDM feature selection technique. In the case of multi-label classification, it is very likely that a feature can belong to one or more classes. Therefore, overlapping feature testing needs to be done to find more specific features for each label. The following are the overlapping features on the resume with FDM feature selection by comparing the top 100 features with the highest scores, which are competencies in each job class label:

Table 2: FDM-FS in discovering the overlapping of key terms describing competencies

Job Label	Web_Dev	Proj_man	Db_Adm	Net_Adm	F_E_Dev	Py_Dev	Sys_Adm	Soft_Dev	Sec_Anlst	Java_Dev
Web_Dev	**100**	45	41	45	36	15	21	22	21	19
Proj_man	45	**100**	37	49	32	29	42	33	42	39
Db_Adm	41	37	**100**	43	29	32	39	36	37	43
Net_Adm	45	45	43	**100**	33	38	41	37	45	48
F_E_Dev	36	32	29	33	**100**	44	31	41	33	35
Py_Dev	15	29	32	38	44	**100**	30	39	32	36
Sys_Adm	21	42	39	41	31	30	**100**	35	42	35
Soft_Dev	22	33	36	37	41	39	35	**100**	35	37
Sec_Anlst	21	42	37	45	33	32	42	35	**100**	40
Java_Dev	19	39	43	48	35	36	35	37	40	**100**

Based on the table 3, it can be concluded that FDM minimizes the overlap of features between labels. FDM can select more specific features for each label, thus reducing overlap. The relevance of attributes in the case of multi-label classification can be more measurable if the feature overlap between classes is minimized. Features that do not overlap between labels can also enhance the model's ability to distinguish between labels, which is important in multi-label classification.

CNN modelling is built using a layered architecture, starting with the sequential function and employing a single input and output tensor. The input tensor represents a matrix of input data obtained using the Keras library's embedding layers. CNN classifies resumes according to a single class. In other words, CNN assigns a classification to each class so that the number of initial classifications corresponds to the number of classes in the resume data. Training data and test data are the two categories into which the dataset is divided for classification. The distribution of test and train data for each job label in this investigation was 10% for test data and 90% for train data.

The results of multi-label CNN classification on job resumes are subsequently used to generate content-based recommendations. As previously explained, the concept of content-based recommendations focuses on learning the similarities between the characteristics of one data point and another, which are then used as a reference for making recommendations. Figure 4 below is an example of the resume recommendation process display based on multi-label CNN classification:

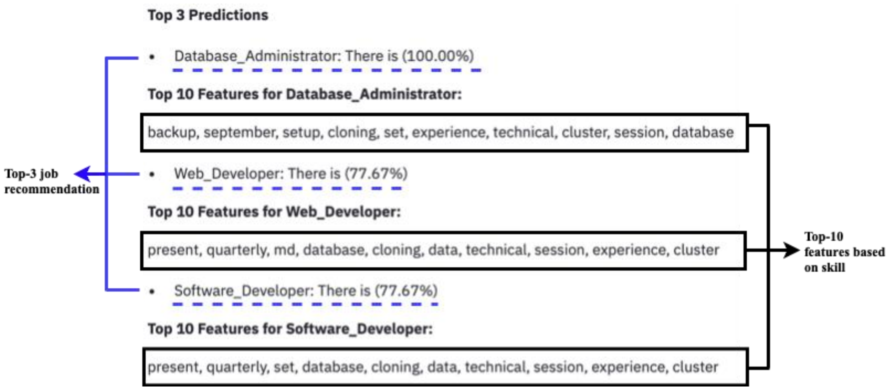

Figure 4: Example of integration of FDM FS and CNN multi-label classification for job resume recommendation

As a test example, textual data containing a job resume is inputted, and predictions are performed to determine the top-3 job recommendations represented by the job resume, along with the top-10 skills that represent the selected job descriptions. These results are obtained from cosine similarity calculations to measure the vector similarity between the searched data and the results of the integration of FDM FS with CNN multi-label text classification.

The integration of FDM FS and CNN multi-label text classification in the example above represents an effort to automate resume analysis that can be applied in companies. This automation can help expedite the resume screening process, particularly when dealing with a large number of applications.

4.1 Discussion

The performance evaluation of the integration between FDM and CNN in multi-label text classification was conducted by comparing multi-label text classification using CNN with the integration of FDM and CNN. In Figure 3, it is evident that the dataset has an imbalanced data distribution and variations in the number of labels within the dataset. To evaluate the capability of FDM and CNN in handling this imbalance, observations were made on the loss function during the training process. Below is the graph of training loss and validation loss from this study, which integrates FDM Feature Selection and CNN Multi-label Text Classification.

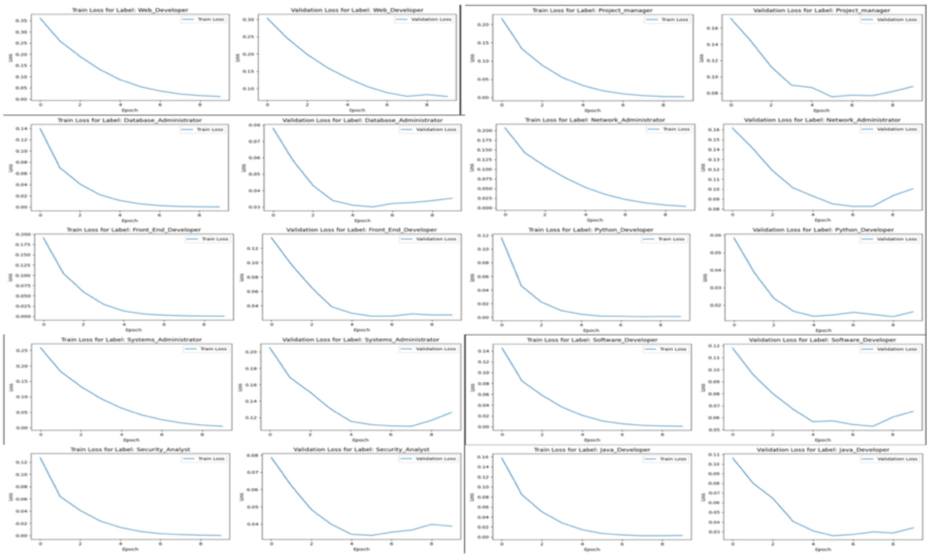

Figure 5: Train and validation loss on job resume

In Figure 5, a consistent decrease in training loss can be observed as the number of epochs increases. This indicates that the model is effectively learning patterns in the training data. Additionally, the validation loss tends to decrease and stabilize, demonstrating that the model has good generalization capabilities. From this analysis, it can be concluded that the integration of FDM FS and CNN shows strong learning capabilities when handling imbalanced datasets with varying numbers of labels for each data. The evaluation metrics used include accuracy, precision, and recall for each job class as well as the overall average performance of the model. Table 4 below presents the results of the performance evaluation comparison between CNN and integration of FDM FS and CNN:

Table 3: Performance of CNN multilabel resume classification

Label	CNN			CNN + FDM		
	Accuray	Precision	Recall	Accuray	Precision	Recall
Web Developer	86.96	75.69	90.84	98.17	95.27	99.55
Project Manager	92.24	79.14	86.41	98.03	91.61	96.42
Database Administrator	98.07	89.41	91.13	98.84	95.25	92.87
Network Administrator	93.39	81.79	75.41	97.75	91.44	94.79
Front End Developer	96.00	90.22	90.71	99.62	98.83	99.36
Python Developer	97.84	89.21	95.30	99.66	97.50	100
System Administrator	92.45	85.19	76.20	96.83	90	95
Software Developer	96.83	98.17	96.91	98.64	98.33	99.59
Security Analyst	97.64	86.80	89.01	98.88	93.11	95.49
Java Developer	96.41	86.89	91.56	99.39	96.81	99.53
AVERGAE	**94.98**	**86.25**	**88.35**	**98.58**	**94.81**	**97.26**

Based on Table 4, the performance comparison between using CNN and CNN+FDM FS in solving the multi-label text classification task can be observed. The application of FDM FS has been proven to improve the performance of the CNN model in terms of accuracy, precision, and recall across all class labels compared to CNN without FDM FS. The integration of FDM FS and CNN in multi-label job resume classification achieved an average accuracy of 98.58%, precision of 94.81%, and recall of 97.26%. Further evaluation was conducted by comparing the results of the experiments with those of previous research [8], as shown in Table 5 below:

Table 4: Comparison of our method with existing study

Method	Accuracy (%)	Precision (%)	Recall (%)
doc2vec+LR	78.63	81.45	94.68
one-hot+LR	78.07	80.11	92.28
Word2vec+CNN	90.22	91.34	98.79
FDM FS+CNN	**98.58**	**94.81**	97.26

Based on Table 5, it can be observed that the integration of FDM FS and CNN outperforms previous studies in terms of accuracy and precision. Efforts to enhance the model's recall performance can be made to meet or even surpass the benchmarks set by earlier studies. This can be achieved through hyperparameter tuning or the application of advanced optimization techniques in the model architecture. FDM provides a richer representation by considering the frequency of words within documents. This capability enables FDM to better capture the importance and relevance of specific words compared to one-hot encoding, doc2vec, and word2vec. Furthermore, CNN is more adept at capturing patterns than LR, allowing it to identify complex relationships within the text.

5 Conclusion

Based on the results of this study, it can be concluded that the multi-label classification model using CNN can be utilized to generate content-based recommendations capable of analyzing the compatibility of job resumes with the classifications of job positions offered. The integration of CNN multi-label text classification with Fuzzy Discernibility Matrix (FDM FS) feature selection in content analysis makes recommendations more adaptive to individual preferences and enhances the relevance of the recommended items. By employing FDM feature selection (FDM FS), this study successfully identifies the most relevant feature subset for each label in a multi-label dataset. This is achieved through ranking discernibility values between items within labels, reducing feature overlap across labels, and improving overall classification accuracy. This contribution enriches text classification methods for multi-label datasets, facilitating broader applications in complex classification tasks such as text content analysis, automatic tagging, and content-based recommendations.

In the future, the integration of FDM and CNN can be tested on more diverse multi-label datasets and applied to other types of data, such as images or audio. The objective is to evaluate the model's performance across various domains and scenarios. The recommendation system can be enhanced with real-time personalization, dynamically adjusting recommendations based on changing user preferences. This approach would make the system more responsive and adaptive to the evolving needs of users over time.

Fundings

The research is supported by: Funding and Implementation of Muhammadiyah National Research Grant Batch VII Year 2024 (No. 0258.285/I.3/D/2024).

References

[1] M. N. Freire and L. N. de Castro, “e-Recruitment recommender systems: a systematic review,” *Knowl Inf Syst*, vol. 63, no. 1, pp. 1–20, 2021, doi: 10.1007/s10115-020-01522-8.

[2] Y. Afoudi, M. Lazaar, and M. Al Achhab, “recommendation system,” *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pp. 1–6, 2019.

[3] D. Wang, Y. Liang, D. Xu, X. Feng, and R. Guan, “Knowledge-Based Systems A content-based recommender system for computer science publications,” *Knowl Based Syst*, vol. 157, no. May, pp. 1–9, 2018, doi: 10.1016/j.knosys.2018.05.001.

[4] G. van Capelleveen, C. Amrit, D. M. Yazan, and H. Zijm, “The recommender canvas: A model for developing and documenting recommender system design,” *Expert Syst Appl*, vol. 129, pp. 97–117, 2019, doi: 10.1016/j.eswa.2019.04.001.

[5] Z. Batmaz, A. Yurekli, A. Bilge, and C. Kaleli, “A review on deep learning for recommender systems : challenges and remedies,” *Artif Intell Rev*, 2018, doi: 10.1007/s10462-018-9654-y.

[6] Y. Afoudi, M. Lazaar, and M. Al Achhab, “recommendation system,” *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, pp. 1–6, 2019.

[7] C. Gan, Q. Feng, and Z. Zhang, “Scalable multi-channel dilated CNN–BiLSTM model with attention mechanism for Chinese textual sentiment analysis,” *Future Generation Computer Systems*, vol. 118, pp. 297–309, 2021, doi: 10.1016/j.future.2021.01.024.

[8] K. Florentin and F. Jiechieu, “Skills prediction based on multi-label resume classification using CNN with model predictions explanation,” *Neural Comput Appl*, vol. 0123456789, 2020, doi: 10.1007/s00521-020-05302-x.

[9] P. Zhang, W. Gao, J. Hu, and Y. Li, “Multi-label feature selection based on the division of label topics,” *Inf Sci (N Y)*, vol. 553, pp. 129–153, 2021, doi: 10.1016/j.ins.2020.12.036.

[10] M. Naumov *et al.*, "Deep learning recommendation model for personalization and recommendation systems," *arXiv preprint arXiv:1906.00091*, 2019, doi: https://doi.org/10.48550/arXiv.1906.00091.

[11] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.

[12] W. Liao, Y. Wang, Y. Yin, X. Zhang, and P. Ma, "Improved sequence generation model for multi-label classification via CNN and initialized fully connection," *Neurocomputing*, vol. 382, pp. 188–195, 2020, doi: 10.1016/j.neucom.2019.11.074.

[13] B. Guo, C. Zhang, J. Liu, and X. Ma, "Improving text classification with weighted word embeddings via a multi-channel TextCNN model," *Neurocomputing*, vol. 363, no. xxxx, pp. 366–374, 2019, doi: 10.1016/j.neucom.2019.07.052.

[14] Z. Shen and S. Zhang, "A Novel Deep-Learning-Based Model for Medical Text Classification," in *Proceedings of the 2020 9th International Conference on Computing and Pattern Recognition*, 2020, pp. 267–273. doi: https://doi.org/10.1145/3436369.3436469.

[15] M. Pota, M. Esposito, G. De Pietro, and H. Fujita, "Best practices of convolutional neural networks for question classification," *Applied Sciences (Switzerland)*, vol. 10, no. 14, 2020, doi: 10.3390/app10144710.

[16] B. Liu, Y. Zhou, and W. Sun, "Character-level text classification via convolutional neural network and gated recurrent unit," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 8, pp. 1939–1949, 2020, doi: 10.1007/s13042-020-01084-9.

[17] B. Guo, C. Zhang, J. Liu, and X. Ma, "Improving text classification with weighted word embeddings via a multi-channel TextCNN model," *Neurocomputing*, vol. 363, pp. 366–374, Oct. 2019, doi: 10.1016/j.neucom.2019.07.052.

[18] J. Xu *et al.*, "Incorporating context-relevant concepts into convolutional neural networks for short text classification," *Neurocomputing*, vol. 386, pp. 42–53, Apr. 2020, doi: 10.1016/j.neucom.2019.08.080.

[19] A. Omar, T. M. Mahmoud, T. Abd-El-Hafeez, and A. Mahfouz, "Multi-label Arabic text classification in Online Social Networks," *Inf Syst*, vol. 100, Sep. 2021, doi: 10.1016/j.is.2021.101785.

[20] Y. Song, Q. V. Hu, and L. He, "P-CNN: Enhancing text matching with positional convolutional neural network," *Knowl Based Syst*, vol. 169, pp. 67–79, 2019, doi: 10.1016/j.knosys.2019.01.028.

[21] A. Jacovi, O. S. Shalom, and Y. Goldberg, "Understanding convolutional neural networks for text classification," *arXiv preprint arXiv:1809.08037*, 2018.

[22] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation Preface to the special issue on Recommender systems based on rich item descriptions," *User Model User-adapt Interact*, vol. 29, no. 2, pp. 239–249, 2019, doi: 10.1007/s11257-019-09231-w.

[23] H. Setiadi, R. Saptono, R. Anggrainingsih, and R. Andriani, "Recommendation Feature of Scientific Articles on Open Journal System Using Content-based Filtering," *2019 6th IEEE International Conference on Engineering Technologies and Applied Science*, no. 516, 2019.

[24] M. Çakır, Ş. G. Öğüdücü, and R. Tugay, "A deep hybrid model for recommendation systems," in *International Conference of the Italian Association for Artificial Intelligence*, Springer, 2019, pp. 321–335.

[25] Z. A. Pardos, Z. Fan, and W. Jiang, "Connectionist recommendation in the wild: on the utility and scrutability of neural networks for personalized course guidance," *User Model User-adapt Interact*, vol. 29, no. 2, pp. 487–525, 2019.

[26] H. Lee, W. Lee, and J. Lee, "A Cascade-hybrid Recommendation Algorithm based on Collaborative Deep Learning Technique for Accuracy Improvement and Low Latency," *Journal of Korea Multimedia Society*, vol. 23, no. 1, pp. 31–42, 2020.

[27] J. Ni, Z. Huang, J. Cheng, and S. Gao, "An effective recommendation model based on deep representation learning," *Inf Sci (N Y)*, vol. 542, pp. 324–342, 2021, doi: 10.1016/j.ins.2020.07.038.

[28] G. Li *et al.*, "Multi-modal visual adversarial Bayesian personalized ranking model for recommendation," *Inf Sci (N Y)*, vol. 572, pp. 378–403, 2021, doi: 10.1016/j.ins.2021.05.022.

[29] C. Wang *et al.*, "Personalized and Explainable Employee Training Course Recommendations: A Bayesian Variational Approach," *ACM Trans Inf Syst*, vol. 40, no. 4, pp. 1–32, 2022, doi: 10.1145/3490476.

[30] T. Pradhan, A. Gupta, and S. Pal, "HASVRec: A modularized Hierarchical Attention-based Scholarly Venue Recommender system," *Knowl Based Syst*, vol. 204, p. 106181, 2020, doi: 10.1016/j.knosys.2020.106181.

[31] H. An and N. Moon, "Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM," *J Ambient Intell Humaniz Comput*, pp. 1–11, 2019.

[32] D. Nguyen, S. Huynh, P. Huynh, C. V Dinh, and B. T. Nguyen, "S2CFT: A New Approach for Paper Submission Recommendation BT - SOFSEM 2021: Theory and Practice of Computer Science," T. Bureš, R. Dondi, J. Gamper, G. Guerrini, T. Jurdziński, C. Pahl, F. Sikora, and P.

W. H. Wong, Eds., Cham: Springer International Publishing, 2021, pp. 563–573.

[33] R. Chatterjee, D. K. Sanyal, and A. Chatterjee, "Fuzzy-Discernibility Matrix-based an Efficient Feature Selection Technique for Improved Motor-Imagery EEG Signal Classification," *bioRxiv*, 2021.

[34] A. Tan, J. Liang, W. Z. Wu, J. Zhang, L. Sun, and C. Chen, "Fuzzy rough discrimination and label weighting for multi-label feature selection," *Neurocomputing*, vol. 465, pp. 128–140, 2021, doi: 10.1016/j.neucom.2021.09.007.

[35] T. J. Sejnowski, "Neural Information Processing Systems," in *The Deep Learning Revolution*, The MIT Press, 2018, pp. 161–168. doi: 10.7551/mitpress/11474.003.0014.

[36] X. Chen, D. Li, P. Wang, and X. Yang, "A Deep Convolutional Neural Network with Fuzzy Rough Sets for FER," *IEEE Access*, vol. 8, pp. 2772–2779, 2020, doi: 10.1109/ACCESS.2019.2960769.

[37] Z. Qi, "The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, IEEE, Jun. 2020, pp. 1241–1246. doi: 10.1109/ICAICA50127.2020.9182555.

[38] W. Qian, C. Xiong, and Y. Wang, "A ranking-based feature selection for multi-label classification with fuzzy relative discernibility," *Applied Soft Computing Journal*, vol. 102, p. 106995, 2021, doi: 10.1016/j.asoc.2020.106995.

[39] P. Khadka and P. Lamichhane, "Content-based Recommendation Engine for Video Streaming Platform," Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.08406

[40] J. M. Moyano, E. L. Gibaja, K. J. Cios, and S. Ventura, "Review of ensembles of multi-label classifiers: Models, experimental study and prospects," *Information Fusion*, vol. 44, pp. 33–45, 2018, doi: 10.1016/j.inffus.2017.12.001.