INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

Performance of Deep Feed-Forward Neural Network Algorithm Based on Content-Based Filtering Approach

Received:
27 June 2024
Accepted:
9 August 2024
Published:
31 August 2024

[1*]Fikri Maulana, [2]Erwin Budi Setiawan
[1,2]Informatics, School of Computing, Telkom University
E-mail: [1]maulanafikri@student.telkomuniversity.ac.id,
[2]erwinbudisetiawan@telkomuniversity.ac.id
*Corresponding Author

Abstract— Background: Selecting a restaurant in a diverse city like Bandung can be challenging. This study leverages Twitter data and local restaurant information to develop an advanced recommendation system to improve decision-making. **Objective:** The system integrates content-based filtering (CBF) with deep feedforward neural network (DFF) classification to enhance the accuracy and relevance of restaurant recommendations. **Methods:** Data was sourced from Twitter and PergiKuliner, with restaurant-related tweets converted into rating values. The CBF combined Bag of Words (BoW) and cosine similarity, followed by DFF classification. SMOTE was applied during training to address data imbalance. **Results:** The initial evaluation of CBF showed a Mean Absolute Error (MAE) of 0.0614 and a Root Mean Square Error (RMSE) of 0.0934. The optimal DFF configuration in the first phase used two layers with 32/16 nodes, a dropout rate of 0.3, and a 20% test size. This setup achieved an accuracy of 81.08%, precision of 82.89%, recall of 76.93%, and f1-scores of 79.23%. In the second phase, the RMSprop optimizer improved accuracy to 81.30%, and tuning the learning rate to 0.0596 further increased accuracy to 89%, marking a 9.77% improvement. **Conclusion:** The research successfully developed a robust recommendation system, significantly improving restaurant recommendation accuracy in Bandung. The 9.77% accuracy increase highlights the importance of hyperparameter tuning. SMOTE also proved crucial in balancing the dataset, contributing to a well-rounded learning model. Future studies could explore additional contextual factors and experiment with recurrent or convolutional neural networks to enhance performance further.

Keywords— Recommender Systems; Twitter; Bag of Words; Content-Based Filtering; Deep Feed Forward

Corresponding Author:
Fikri Maulana,
Informatics,
Telkom University,
maulanafikri@student.telkomuniversity.ac.id
Orchid ID: https://orcid.org/0009-0008-3523-8445

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

I. INTRODUCTION

In the current phase of technological development, efficient recommendation systems are crucial for easing decisions. Culinary experiences have become not only a necessity but also a part of lifestyle and entertainment for society[1]. Bandung, renowned for its diverse culinary offerings, including traditional Sundanese cuisine, has emerged as a popular destination for tourists eager to explore its rich food culture [2], [3].

In response to the booming culinary industry in Bandung, recommendation systems have become increasingly essential. These systems are designed to help users find culinary options that match their preferences, offering a more personalized culinary experience amidst the city's diverse food scene. Innovation arises from the integration of Twitter data (X), enriching the information sources with users' tweets about their culinary experiences. Twitter, founded in 2006 by Jack Dorsey, has evolved into a social media platform that allows users to share ideas, thoughts, and feelings within a 280-character limit [4]. This approach not only creates a dynamic and relevant recommendation experience but also provides real-time insights into the culinary tastes of Bandung's residents, resulting in more precise and impactful recommendations.

Based on previous research, the effectiveness of various methods in recommendation systems has been well-documented, particularly with Content-Based Filtering (CBF). Study by [1] used CBF with Bidirectional Gated Recurrent Unit (Bi-GRU) classification, achieving a Mean Absolute Error (MAE) of 0.254 and a Root Mean Square Error (RMSE) of 0.425. By implementing optimization techniques such as Nadam, and SMOTE, accuracy increased from 74.7% to 86.8%. Study by [5] used CBF with Convolutional Neural Network (CNN) classification, achieving a Mean Absolute Error (MAE) of 0.28 and a Root Mean Square Error (RMSE) of 0.67. By implementing optimization techniques such as Stochastic Gradient Descent (SGD), SMOTE, and embedding techniques, accuracy increased from 78.83% to 86.41%. Similarly, study by [6] employed a switching hybrid filtering (SHF) method that combined CBF with RoBERTa and item-based collaborative filtering, achieving a MAE of 0.0617 and an RMSE of 0.1178, with an RNN classifier resulting in an accuracy of 86.11%. Study by [7] implemented Cascade Hybrid Filtering with CNN and RMSProp optimization, achieving an MAE of 0.8643, RMSE of 0.6325, and an improved accuracy of 88.40%, marking a 6.00% increase from the baseline. Study by [8] used Collaborative Filtering (CF) and Feed Forward (FF) classification for a movie recommendation system, achieving an accuracy of 67.235% with a network architecture of 5-7-5-3-14.

Despite these advancements, the combination of CBF and DFF has not been thoroughly explored. Research [1], [5], [6], [7] focused on CBF but did not integrate DFF for classification.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

Additionally, research [8] employed Feed Forward networks with Collaborative Filtering rather than CBF. This study addresses the gap by integrating CBF with DFF to enhance recommendation accuracy and provide a more personalized recommendation system.

To implement Content-Based Filtering (CBF), this research sources data from two platforms: Twitter and PergiKuliner. Data crawling from Twitter collects relevant details such as usernames, tweets, and other posting information, while PergiKuliner provides data on restaurant names, types, and descriptions. This data undergoes preprocessing to clean and prepare it for the CBF process. To address class imbalance that may arise from CBF, the Synthetic Minority Over-sampling Technique (SMOTE) is applied before moving on to Deep Feed Forward (DFF) classification. The DFF classification is then performed in three stages: determining the optimal number of layers, selecting the most effective optimization techniques (e.g., Stochastic Gradient Descent), and fine-tuning the learning rate. This research aims to assess how effectively DFF, when combined with CBF, can enhance recommendation accuracy.

II. RESEARCH METHOD

This study develops a recommendation system for culinary spots in Bandung using two different approaches. The first approach employs Content-Based Filtering (CBF), while the second utilizes a Deep Feed Forward (DFF) for classification. The system architecture is depicted in Fig. 1.

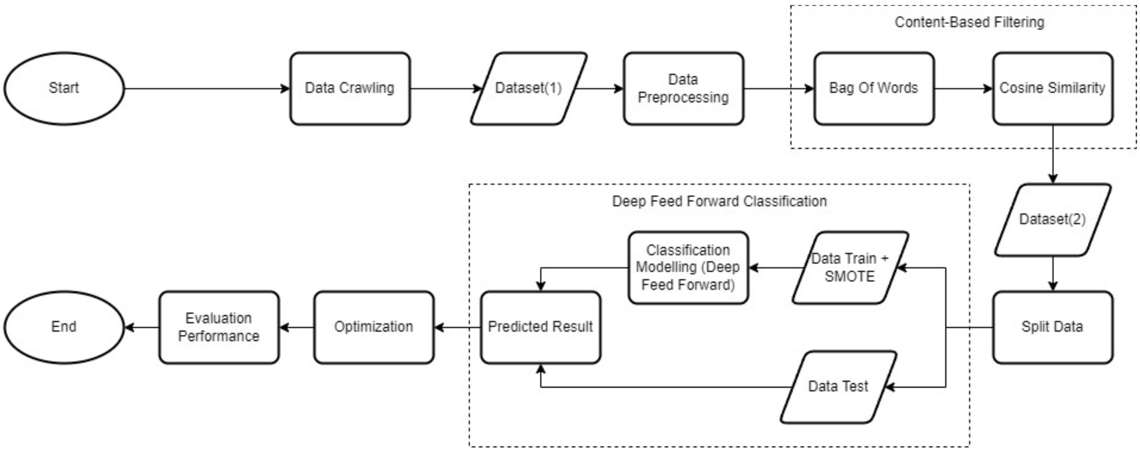

Fig 1. CBF and DFF Design System

A. Recommender Systems

Recommendation systems are algorithms that allow users to receive items according to their preferences [9]. Generally, there are four types of recommendation systems: Collaborative Filtering (CF), Content-Based Filtering (CBF), Knowledge-Based, and Hybrid algorithms [9],

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

[10], [11]. Among these, CF and CBF are the primary choices. Their success lies in their different approaches: CF leverages user preferences and similarities with other users, while CBF uses item characteristics and user preferences. Combining these methods, as in hybrid approaches, can overcome the weaknesses of each, providing more accurate and diverse recommendations for users. Therefore, CF and CBF are often the preferred choices for implementing recommendation systems [12].

B. Data Crawling

Data crawling is the automated process of extracting information from websites. This technique is essential for gathering large volumes of data quickly and efficiently, allowing researchers to compile comprehensive datasets for analysis. This research utilizes two types of data from different sources. The first dataset, "Restaurants Reviews Dataset" consists of tweets from Twitter collected through crawling based on keywords matching those in the second dataset, as shown in Table 1. The tweets were gathered using the Tweet-Harvest library. The second dataset, "Restaurants Dataset" contains restaurant information obtained through web crawling on the PergiKuliner website, featuring details such as restaurant name, type of restaurant, and address. Since the PergiKuliner dataset lacks descriptions, these were manually added by the researcher as shown in Table 2. Data crawling for restaurant information was performed using the BeautifulSoup library. Both datasets were preprocessed before being integrated into the recommendation system stage.

Table 1. Example of Restaurants Reviews Dataset

Account Name	Restaurant	Tweet Text
aarrddyyee_95	de.u Coffee	@moejaaaa Km cari coffee shop yg ky gmn nih? Yg rame sepi atau yg kopi nya enak ….
.....	.....	.....
wijayasasnaa	Bakmie Tjo Kin	top tier perbakmie-an di bdg udh paling mantep bakmie 96 (2x), bakmie sedjuk(1x), bakmie tjoi (2x). tjo kin udh 2x ttp meh heheu ….

Table 2. Example of Restaurants Dataset

Restaurant Name	Type of restaurant	.....	Description
150 Coffee and Garden	[Kafe]	.....	150 Coffee and Garden adalah kafe yang terletak di area Taman Hutan Raya Ir. H. Djuanda. Kafe ini memiliki suasana yang tenang dan asri, dengan pemandangan hutan yang indah ….
.....	.....	.....	.....
Yoshinoya	[Jepang]	.....	Yoshinoya adalah pilihan yang tepat bagi Anda yang ingin menikmati cita rasa Jepang yang autentik dengan harga terjangkau. Restoran ini menawarkan berbagai menu gyudon yang lezat dan porsi besar.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

C. Data Preprocessing

In this research, the conversion of reviews into ratings was carried out during the data preprocessing phase, which involved removing irrelevant information such as emoticons, tags, mentions, and links. This process effectively cleaned up the reviews, eliminated unnecessary elements, and converted the text to lowercase. The polarity score was calculated using the TextBlob library, resulting in a score ranging from 0 to 5 that represents the user's rating of the restaurants. After preprocessing, the restauran review data was transformed into a rating dataset as depicted in Table 3.

Table 3. Rating Dataset

Account Name	Restaurant	Polarity Score
BaseBDG	150 Coffee and Garden	3.366071
BaseBDG	Ayam Geprek Pangeran	3.000000
…..	…..	…..
velabllza	Capdangu	3.000000
velabllza	Cupola	2.645833

Next, the rating dataset was combined and transformed into a 200 x 45 matrix, representing the number of restaurants and users. In this matrix, the columns represented users from Twitter, while the rows represented Restaurants name. The matrix values reflected the polarity and rating scores from these websites. This dataset, named the "Final Dataset" was the product of the data preparation process as shown in Table 4.

Table 4. Final Dataset

	BaseBDG	DraftAnakUnpad	…..	undipmenfess	velabllza
150 Coffee and Garden	3.37	0.00	…..	0.00	0.00
Ambrogio Patisserie	0.00	0.00	…..	0.00	0.00
…..	…..	…..	…..	…..	…..
Xing Fu Tang	0.00	0.00	…..	0.00	0.00
Yoshinoya	2.98	2.5	…..	0.00	0.00

D. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is an over-sampling technique that increases the number of instances in the minority class by generating synthetic data from existing minority class instances. The over-sampling process in SMOTE involves selecting instances from the minority class and finding their k-nearest

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

neighbors, then creating synthetic instances based on these neighbors [13]. This approach helps mitigate excessive overfitting by avoiding the mere replication of existing minority class instances.

In the SMOTE algorithm, the process begins by calculating the difference between the feature vectors of a minority class instance and its nearest neighbor from the same class. This difference is then multiplied by a random number between 0 and 1. The result of this calculation is added to the original feature vector to generate a new synthetic feature vector[14].

E. Content-Based Filtering

Content-Based Filtering (CBF) is a method in recommendation systems that utilizes item content information, such as features, descriptions, or genres, to recommend similar items based on feature similarities with user preferences [12], [15], [16], [17]. Its main advantage lies in its ability to provide quick and relevant recommendations based on the feature similarities between items and user preferences [18]. The CBF method starts by constructing a Bag of Words (BoW) vectorizer to calculate similarity using cosine similarity and generate a rating prediction value. This prediction value is then used to identify the Top N Recommendations from both approaches.

1. Bag Of Words (BoW)

Bag Of Words (BoW) is one of the commonly used feature extraction techniques. The BoW model is an approach in natural language processing that treats each document as a "bag" of words [19]. In BoW, the order and structure of words are disregarded, and only the presence of specific words is considered. Each document is represented as a vector where the value of each element reflects the presence or frequency of a specific word in that document.

2. Cosine Similarity

Cosine similarity is a frequently used calculation for measuring similarity between items [18], [20], [21]. Essentially, a similarity function is a function that takes two real values (0 and 1) as input and produces a similarity value between these two values in the form of a real number. This approach is used to calculate the cosine angle value between two vectors, particularly in the context of measuring similarity between two documents [21]. The cosine similarity function between item A and item B can be described in Equation (1).

$$sim(A,B) = \frac{n(A \cap B)}{\sqrt{n(A)n(B)}} \qquad (1)$$

If the similarity value between two objects is 1, then they are identical; conversely, if the similarity value is 0, they are not identical. The higher the similarity value, the more similar the two objects, and vice versa.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

F. Deep Feed Forward Classification

Feed Forward (FF) (fig 2) is a type of neural network used for prediction. FF utilizes the backpropagation algorithm to adjust its weights [22]. This backpropagation algorithm was initially developed by David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams in 1986. The algorithm involves three main stages: FF from input patterns, calculation and backpropagation of errors, and weight adjustment [22].

The structural difference between Feed Forward (FF) and Deep Feed Forward (DFF) (Fig 3) lies in the number of hidden layers. A DFF network has a structure very similar to FF, with the main difference being the depth of the hidden layers. Neural networks with only one hidden layer are often referred to as "shallow" networks or simply FF networks. On the other hand, Deep networks, which involve multiple hidden layers (2 or more), are characterized by greater depth [23].

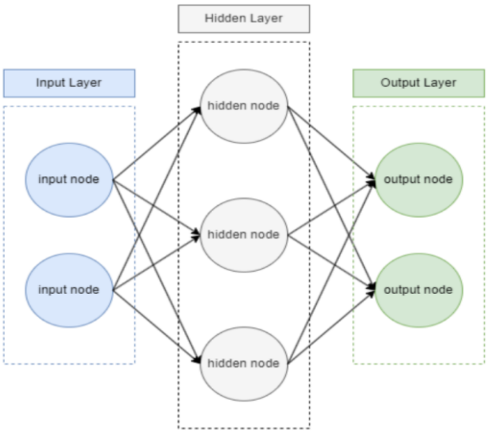

Fig 2. Feed Forward

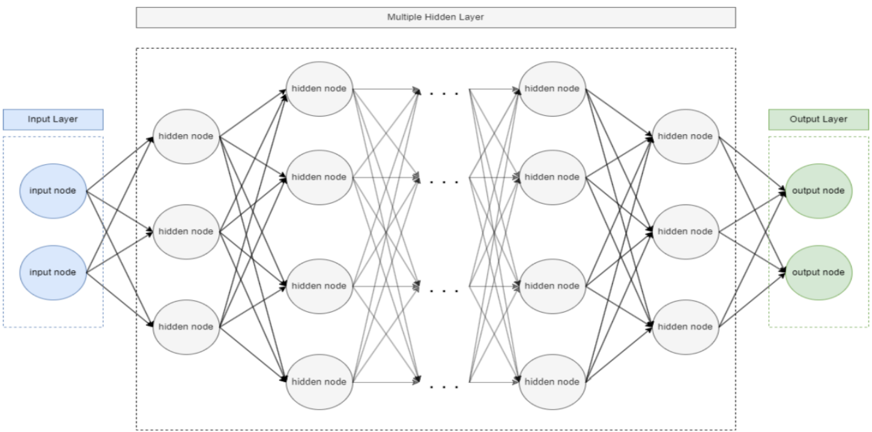

Fig 3. Deep Feed Forward

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

Illustrating the structural difference between FF and DFF can be presented in the form of a diagram highlighting the difference in the number of hidden layers in both networks, as shown in Fig. 2, and 3. Adding additional layers aims primarily to enhance the network's performance in handling more complex tasks.

G. Optimization

With the progression of deep learning, diverse model architectures have emerged, highlighting the significance of optimizers in the training process. These optimizers play a crucial role in enhancing the performance and efficiency of neural networks[24]. Optimization in neural networks involves refining the model's performance through various adjustments to boost accuracy. Numerous optimization algorithms are available for neural network models, including Adam, Nadam, Adagrad, Adadelta, SGD, RMSprop, and others. These algorithms adjust the model's weights and learning rate during training to minimize the loss function and increase accuracy [6].

The learning rate is a crucial hyperparameter in the optimizer, determining the extent to which model weights are adjusted with each update during training. If the learning rate is set too low, the training process can be excessively slow or may have difficulty converging. On the other hand, a higher learning rate can prevent poor results by maintaining a broader search range, though it can make convergence more challenging [6], [25]. Hence, selecting the right learning rate is essential for efficient training and optimal model convergence.

H. Evaluation Performance

Evaluation performance in recommendation systems involves several aspects, one of which can be measured using classification accuracy metrics that rely on the Confusion Matrix [21]. The Confusion Matrix is a method used to evaluate system performance, containing information that compares the system's classification results with the expected classification outcomes [26], [27].

Table 5. Confusion Matrix

Confusion Matrix		**Predict Values**	
		Positive	***Negative***
Actual Values	Positive	TP	FN
	Negative	FP	TN

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$Acuraccy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

$$F1\ Scores = 2\ X \frac{Recall\ x\ Precision}{Recall + Precision} \quad (5)$$

From Table 5, when recommendations match user preferences, they are True Positives (TP). If recommendations do not match, they are False Positives (FP). If the system does not provide a recommendation and it does not match preferences, it is a True Negative (TN) [28]. Precision, recall, accuracy, and F1 Scores are formulated with Equations (2), (3), (4), and (5), respectively. Precision measures the accuracy of the match between requested and provided information. Recall indicates the model's success rate in retrieving relevant information. Accuracy shows how well predicted values match actual values [25]. The F1 Scores, the harmonic mean of precision and recall, balances these metrics, particularly useful for imbalanced datasets.

Next, the evaluation process will use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). These two metrics will calculate the average error in predictions by comparing the actual rating values and the predicted ratings of products by users [29]. MAE and RMSE can be calculated using Equations (6) and (7).

$$MAE = \frac{\sum_{i=1}^{n} |p_i - r_i|}{n} \quad (6)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - r_i)^2}{n}} \quad (7)$$

Let n be the number of ratings in the test set, p_i refers to the predicted rating, and r_i refers to the true rating in the test data. In the calculation of MAE and RMSE, a lower result indicates a higher accuracy of the recommendation system's predictions.

III. RESULT AND DISCUSSION

This study was primarily divided into two phases: recommendation system, and classification. In the recommendation system phase, the focus was on predicting ratings using Content-Based Filtering (CBF). The resulting rating predictions were evaluated using MAE and RMSE metrics. The classification phase aimed to label each rating, utilizing a Deep Feed Forward (DFF) for this purpose. The effectiveness of this phase was measured through accuracy, precision, recall, and f1-score. Each phase is described in more detail below:

A. Content-Based Filtering Result

Content-Based Filtering (CBF) was applied to Dataset 1 (Final Dataset) by combining the "Description," "Restaurant," "Type of restaurant," and "Price Range" columns from the "Restaurant Dataset" into a "content" column. This feature was cleaned up, such as removing 0

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

value entries, and used to build item profiles. The Bag of Words (BoW) approach was then used to convert the "content" column into numerical vectors, enabling the calculation of the cosine similarity matrix between restaurants. As shown in Tables 6 and 7, the predicted rating values ranged between 0 and 1. A rating close to 0 (less than or equal to 0.5) indicated that the restaurant was not recommended, while a rating close to 1 (more than 0.5) indicated that the restaurant was recommended. CBF using BoW produced an MAE of 0.0614 and an RMSE of 0.0934.

Table 6. Cosine Similarities with Bag of Words

	150 Coffee and Garden	Ambrogio Patisserie	…..	Xing Fu Tang	Yoshinoya
150 Coffee and Garden	1.000000	0.310179	…..	0.325950	0.372494
Ambrogio Patisserie	0.310179	1.000000	…..	0.358626	0.409701
…..	…..	…..	…..	…..	…..
Xing Fu Tang	0.325950	0.358626	…..	1.000000	0.316152
Yoshinoya	0.372494	0.409701	…..	0.316152	1.000000

Table 7. BOW with Content-Based predictions result

	BaseBDG	DraftAnakUnpad	…..	undipmenfess	velabllza
150 Coffee and Garden	0.674	0.069927	…..	0.052845	0.055221
Ambrogio Patisserie	0.092009	0.069519	…..	0.056871	0.45192
…..	…..	…..	…..	…..	…..
Xing Fu Tang	0.093739	0.068592	…..	0.064103	0.043854
Yoshinoya	0.596	0.625	…..	0.057851	0.042518

To present the top n recommendations, this study provides the top 5 restaurant recommendations for each user, as shown in Table 8. By leveraging the cosine similarity values, the system identifies and ranks the most similar restaurants for each user. These top 5 recommendations, based on the highest similarity scores, ensure personalized suggestions tailored to user preferences, enhancing the relevance and user experience.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

Table 8. BOW with Content-Based predictions result

	1	2	….	4	5
BaseBDG	Warung Sate Bu Ngantuk	Steak Warjo	…..	Lomie dan Bakmie Lombok	Sushi Hiro
DraftAnakUnpad	Kopi Kenangan	BBQ Mountain Boys	…..	Mie Gacoan	Baso Aci Ganteng
…..	…..	…..	…..	…..	…..
undipmenfess	Bebek Kaleyo	Kartika Sari	…..	Chattime	Golden Lamian
velabllza	Makmur Jaya Coffee Roaster	San.da.ran	…..	Merindu canteen & Coffee	Masagi Koffee

B. Deep Feed Forward Classification Result

Classification is performed using a Deep Feed Forward (DFF). Initially, the results from the Content-Based Filtering (CBF) were scaled to a 1-5 rating range using MinMaxScaler to ensure appropriate scaling for the classification process. Table 9 details this conversion process. The scaled CBF results were then labeled as 0 and 1, where 0 represented ratings below 2.5 and 1 represented ratings above 2.5. This resulted in 7,264 instances of label 0 data (82.55%) and 1,536 instances of label 1 data (17.45%), as shown in Table 10.

Table 9. BOW with Content-Based scaled to 1-5

	BaseBDG	DraftAnakUnpad	…..	undipmenfess	velabllza
150 Coffee and Garden	3.573078	1.015573	…..	1.012702	1.074277
Ambrogio Patisserie	1.025669	1.013826	…..	1.029651	1.032605
…..	…..	…..	…..	…..	…..
Xing Fu Tang	1.033242	1.009855	…..	1.060099	1.027046
Yoshinoya	3.231667	3.393502	…..	1.033778	1.021496

Table 10. Dataset for classification

	BaseBDG	DraftAnakUnpad	…..	undipmenfess	velabllza
150 Coffee and Garden	1	0	…..	0	0
Ambrogio Patisserie	0	0	…..	0	0
…..	…..	…..	…..	…..	…..
Xing Fu Tang	0	0	…..	0	0
Yoshinoya	1	1	…..	0	0

The next step involved evaluating the performance of the deep feedforward neural network (DFF) through a multi-stage process. Initially, we tested the model using different test sizes—

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

10%, 20%, 30%, and 40%—to identify the best-performing configuration. For each feature, we trained the model by excluding one column at a time, treating it as the label. The data was split accordingly, and the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training data to address class imbalance. We experimented with different layer configurations, testing models with 2, 3, 4, and 5 layers, each with varying of nodes and dropout layers to prevent overfitting. Once the optimal layer configuration was determined, we advanced to the next stage, where we applied various optimization algorithms, including Adam, SGD, RMSprop, Adagrad, Adadelta, and Nadam, to the baseline model to assess their effects on accuracy. The optimizer that yielded the highest accuracy was selected for the final stage, where we fine-tuned the model by experimenting with different learning rates to identify the most effective one. This comprehensive, systematic approach, which included SMOTE for addressing class imbalance and optimization of hyperparameters, enabled us to enhance the model's performance, resulting in a robust and well-optimized DFF that generalizes effectively across different feature sets.

Table 11. Deep Feed Forward Baseline

	Test Size	Accuracy	Precision	Recall	F1-Scores
2 Layers	10	80.64%	83.14%	76.70%	79.31%
	20	**81.08%**	**82.89%**	**76.93%**	**79.23%**
	30	80.96%	83.61%	76.93%	79.79%
	40	80.90%	82.99%	77.38%	79.84%
3 Layers	10	77.35%	83.97%	77.95%	80.53%
	20	77.46%	83.07%	76.59%	79.24%
	30	76.63%	83.19%	76.51%	79.39%
	40	76.91%	83.32%	77.18%	79.81%
4 Layers	10	74.87%	83.19%	76.47%	79.31%
	20	75.15%	82.09%	75.62%	78.25%
	30	74.18%	83.50%	76.66%	79.60%
	40	74.20%	82.93%	76.33%	79.15%
5 Layers	10	73.14%	83.29%	76.47%	79.22%
	20	73.59%	82.13%	75.39%	78.18%
	30	71.86%	82.57%	75.11%	78.31%
	40	72.04%	82.96%	75.90%	78.88%

From the experiments conducted on layers 2 to 5, as shown in Table 11, it was found that the configuration of layer 2 with 32/16 nodes and a dropout rate of 0.3, and a test size of 20, achieved the highest accuracy. This setup resulted in an accuracy of 81.08%, a precision of 82.89%, and a recall of 76.93% and 79.23%. Therefore, the researcher will proceed to the second phase using this model configuration.

INTENSIF, Vol.8 No.2 August 2024
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

Table 12. Deep Feed Forward with Optimizer

Optimizer	Accuracy	Precision	Recall	F1-Scores
ADAM	80.49% (-0.73%)	82.61% (-0.34%)	76.30% (-0.82%)	78.90% (-0.42%)
SGD	68.04% (-16.08%)	82.66% (-0.28%)	73.97% (-3.85%)	77.58% (-2.08%)
RMSPROP	**81.30% (+0.27%)**	**83.14% (+0.30%)**	**77.32% (+0.51%)**	**79.74% (+0.64%)**
ADAGRAD	54.97% (-32.22%)	81.48% (-1.70%)	59.31% (-22.89%)	66.49% (-16.07%)
ADADELTA	80.07% (-1.24%)	82.49% (-0.48%)	75.96% (-1.26%)	78.59% (-0.81%)

In the second phase, the researcher added different optimizers to the previous configuration. The optimizers compared were Adam, SGD, RMSprop, Adagrad, Adadelta, and Nadam. The results of this phase, shown in Table 12, indicate that the RMSprop optimizer achieved the highest accuracy at 81.30%, marking an improvement of a few percentage points over the baseline. Consequently, the researcher will proceed to the third phase using the RMSprop optimizer, incorporating additional tuning to further enhance accuracy.

Table 13. Deep Feed Forward with Optimizer + Learning Rate

Learning Rate	Accuracy	Precision	Recall	F1-Scores
0.09540954763 499943	86.06% (+6.15%)	82.71% (-0.22%)	72.67% (-5.53%)	76.46% (-3.49%)
0.07543120063 354622	88.06% (+8.62%)	83.12% (+0.28%)	74.94% (-2.59%)	78.06% (-1.48%)
0.05963623316 594648	**89.00% (+9.77%)**	**83.04% (+0.18%)**	**75.56% (-1.78%)**	**78.52% (-0.90%)**
0.12067926406 393200	83.19% (+2.61%)	82.52% (-0.45%)	71.53% (-7.02%)	75.08% (-5.23%)

In the third phase, the researcher focused on selecting the optimal learning rate for the RMSprop optimizer. The learning rate, a key hyperparameter that affects the number of steps taken in each learning iteration, can significantly influence model stability and accuracy [30]. Experiments were conducted with learning rates ranging from 1e-10 to 1e0, tested over 50 epochs and 5 periods. Additional parameters, such as exponential moving average, beta, epsilon, and momentum, were incorporated to improve optimizer performance and model convergence. The experiments conducted in this phase led to a notable increase in accuracy, rising from the previous baseline of 81.08% to 89.00%, reflecting an improvement of 9.77%, with a learning rate of 0.05963623316594648. These results demonstrate the effectiveness of this approach in enhancing accuracy and indicate a significant improvement, showing a good advancement over previous outcomes [25]. Table 13 presents the results, highlighting the optimal learning rate and the corresponding accuracy achieved with the RMSprop optimizer.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

The next step involved conducting a statistical significance test of the experimental scenarios. This stage aimed to demonstrate statistically significant changes in accuracy from the conducted experiments. The P-value and Z-value were used as parameters, where the P-value indicated the likelihood of no significant change (if less than 0.05), while the Z-value indicated that the difference between the two scenarios was significant at the 95% confidence level. Based on Table 14, there was a significant change in accuracy in all scenarios. The S1→S3 change indicated that the proposed model provided better accuracy compared to the baseline. The results of the increase in accuracy obtained from this research can be seen in Fig. 4.

Table 14. Accuracy significant improvement

Parameters	Scenarios		
	S1→S2	S2→S3	S1→S3
Z-Value	15.66	309.00	199.00
P-Value	0.0	0.0	0.0
Significant	**True**	**True**	**True**

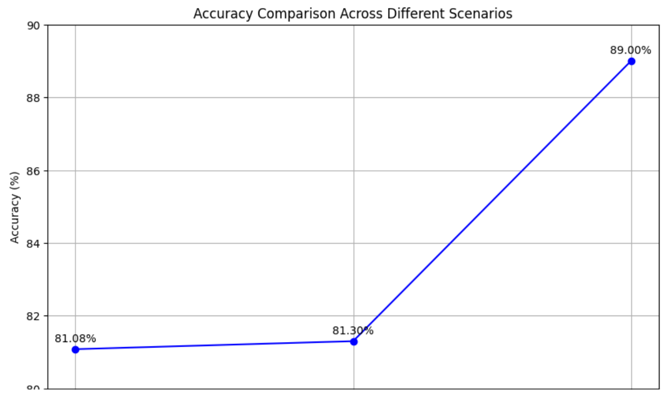

Fig 4. Accuracy Improvement Graph

IV. CONCLUSION

In this study, the researchers successfully developed a restaurant recommendation system for Bandung by combining Content-Based Filtering (CBF) using Bag of Words (BoW) and Cosine Similarity for recommendations with Deep Feed Forward Neural Networks (DFF) for classification. Various scenarios were explored, including determining the optimal number of layers, comparing different optimization techniques, and identifying the most effective learning rate for fine-tuning the model. Based on experiments, it can be concluded that this approach resulted in a Mean Absolute Error (MAE) of 0.0614 and a Root Mean Square Error (RMSE) of 0.0934 for CBF. The classification process employed a DFF, optimized in three phases. To

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

address data imbalance in the training data, the Synthetic Minority Over-sampling Technique (SMOTE) was applied before model training. After balancing the data, the optimal layer configuration was determined to be a two-layer setup with 32/16 nodes, a dropout rate of 0.3, and a test size of 20%. This configuration achieved an accuracy of 81.08%, with precision at 82.89% and recall rates of 76.93% and 79.23%. Subsequently, utilizing the RMSprop optimizer increased the accuracy to 81.30%, and tuning the learning rate for RMSprop to 0.0596 further enhanced accuracy to 89%, representing a significant 9.77% improvement. This study highlights the effectiveness of combining CBF with DFF classification, particularly when addressing class imbalances using SMOTE, in enhancing restaurant recommendation accuracy. However, this approach has limitations due to its reliance on textual features extracted using Bag of Words (BoW) and Cosine Similarity, which may overlook subtle aspects of user preferences that could be better captured through more advanced natural language processing techniques, such as word embeddings or transformers. To address this limitation, future research should consider incorporating word embeddings, which can better capture the contextual meaning of words. Additionally, exploring additional contextual factors such as user preferences and social media interactions, along with experimenting with neural network architectures like recurrent or convolutional networks and advanced optimization techniques, could further enhance the accuracy and performance of the recommendation system.

Author Contributions: *Fikri Maulana:* was responsible for writing the original draft, data curation, formal analysis, and conducting the experiments. *Erwin Budi Setiawan:* contributed to the conceptualization, methodology, and review for editing of the manuscript.

All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: The source code for data crawling and analysis is publicly available at https://github.com/fikrimln16/crawling-pergikuliner/blob/main/code.py.

Informed Consent: The data used in this study were collected from publicly available sources, specifically Twitter (X) and PergiKuliner, and did not involve direct interaction with human subjects. Therefore, informed consent was not applicable. The collection process complied with the terms of service of the respective platforms.

Animal Subjects: There were no animal subjects.

ORCID:
Fikri Maulana: https://orcid.org/0009-0008-3523-8445
Erwin Budi Setiawan: https://orcid.org/0000-0002-2121-8776

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

REFERENCES

[1] A. N. Faadhilah and E. B. Setiawan, "Content-Based Filtering in Recommendation Systems Culinary Tourism Based on Twitter (X) Using Bidirectional Gated Recurrent Unit (Bi-GRU)," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 10, no. 2, pp. 406–418, 2024, **doi:** 10.26555/jiteki.v10i2.29010.

[2] D. Turgarini, "Bandung City as A Sundanese Gastronomy Foodscape," *Digital Press Social Sciences and Humanities*, vol. 4, p. 00004, Jul. 2020, **doi:** 10.29037/digitalpress.44351.

[3] "The Representation of Culinary Experience as the Future of Indonesian Tourism Cases in Bandung City, West Java," *International Journal of Business and Economic Affairs*, vol. 2, no. 5, Oct. 2017, **doi:** 10.24088/IJBEA-2017-25001.

[4] N. Ben-Lhachemi, E. H. Nfaoui, and J. Boumhidi, "Hashtag Recommender System Based on LSTM Neural Reccurent Network," in *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, IEEE, Oct. 2019, pp. 1–6. **doi:** 10.1109/ICDS47004.2019.8942380.

[5] A. Nilla and E. B. Setiawan, "Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 1, p. 17, Feb. 2024, **doi:** 10.26555/jiteki.v9i4.28113.

[6] "Movie Recommendation System Based on Tweets Using Switching Hybrid Filtering with Recurrent Neural Network," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 2, pp. 277–293, Apr. 2024, **doi:** 10.22266/ijies2024.0430.24.

[7] I. H. Arsytania, E. B. Setiawan, and I. Kurniawan, "Movie Recommender System with Cascade Hybrid Filtering Using Convolutional Neural Network," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 10, no. 2, pp. 188–200, 2024, **doi:** 10.26555/jiteki.v9i4.28146.

[8] Y. M. Arif, D. Wardani, H. Nurhayati, and N. M. Diah, "Non-Rating Recommender System for Choosing Tourist Destinations Using Artificial Neural Network," *Applied Information System and Management (AISM)*, vol. 6, no. 2, pp. 61–68, Sep. 2023, **doi:** 10.15408/aism.v6i2.26741.

[9] A. Pal, P. Parhi, and M. Aggarwal, "An improved content based collaborative filtering algorithm for movie recommendations," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, IEEE, Aug. 2017, pp. 1–3. **doi:** 10.1109/IC3.2017.8284357.

[10] P. Wang, Q. Qian, Z. Shang, and J. Li, "An recommendation algorithm based on weighted Slope one algorithm and user-based collaborative filtering," in *2016 Chinese Control and Decision Conference (CCDC)*, IEEE, May 2016, pp. 2431–2434. **doi:** 10.1109/CCDC.2016.7531393.

[11] S. Natarajan, S. Vairavasundaram, S. Natarajan, and A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using Linked Open Data," *Expert Syst Appl*, vol. 149, p. 113248, Jul. 2020, **doi:** 10.1016/j.eswa.2020.113248.

[12] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim, and R. Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities," *Applied Sciences*, vol. 10, no. 21, p. 7748, Nov. 2020, **doi:** 10.3390/app10217748.

[13] A. Fernandez, S. Garcia, F. Herrera, and N. V. Chawla, "SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary," *Journal of Artificial Intelligence Research*, vol. 61, pp. 863–905, Apr. 2018, **doi:** 10.1613/jair.1.11192.

[14] S. T. Jishan, R. I. Rashu, N. Haque, and R. M. Rahman, "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique," *Decision Analytics*, vol. 2, no. 1, p. 1, Dec. 2015, **doi:** 10.1186/s40165-014-0010-2.

[15] J. Son and S. B. Kim, "Content-based filtering for recommendation systems using multiattribute networks," *Expert Syst Appl*, vol. 89, pp. 404–412, Dec. 2017, **doi:** 10.1016/j.eswa.2017.08.008.

[16] B. R. Cami, H. Hassanpour, and H. Mashayekhi, "A content-based movie recommender system based on temporal user preferences," in *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, IEEE, Dec. 2017, pp. 121–125. **doi:** 10.1109/ICSPIS.2017.8311601.

[17] G. Geetha, M. Safa, C. Fancy, and D. Saranya, "A Hybrid Approach using Collaborative filtering and Content based Filtering for Recommender System," *J Phys Conf Ser*, vol. 1000, p. 012101, Apr. 2018, **doi:** 10.1088/1742-6596/1000/1/012101.

[18] C. Fiarni and H. Maharani, "Product Recommendation System Design Using Cosine Similarity and Content-based Filtering Methods," *IJITEE (International Journal of Information Technology and Electrical Engineering)*, vol. 3, no. 2, p. 42, Sep. 2019, **doi:** 10.22146/ijitee.45538.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v8i2.22904

[19] S. Akuma, T. Lubem, and I. T. Adom, “Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets,” *International Journal of Information Technology (Singapore)*, vol. 14, no. 7, pp. 3629–3635, Dec. 2022, **doi:** 10.1007/s41870-022-01096-4.

[20] N. Valentino and E. B. Setiawan, “Movie Recommender System on Twitter Using Weighted Hybrid Filtering and GRU,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, pp. 159–172, May 2024, **doi:** 10.22219/kinetik.v9i2.1941.

[21] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, “Movie Recommendation System using Cosine Similarity and KNN,” *Int J Eng Adv Technol*, 2020, **doi:** 10.35940/ijeat.E9666.069520.

[22] Y. Hasbi, W. Budi, and S. Rukun, “Feed Forward Neural Network Modeling for Rainfall Prediction,” *E3S Web of Conferences*, vol. 73, p. 05017, Dec. 2018, **doi:** 10.1051/e3sconf/20187305017.

[23] C. C. Lee, M. H. F. Rahiman, R. A. Rahim, and F. S. A. Saad, “A Deep Feedforward Neural Network Model for Image Prediction,” *J Phys Conf Ser*, vol. 1878, no. 1, p. 012062, May 2021, **doi:** 10.1088/1742-6596/1878/1/012062.

[24] N. Fatima, “Enhancing Performance of a Deep Neural Network: A Comparative Analysis of Optimization Algorithms,” *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 9, no. 2, pp. 79–90, Jun. 2020, **doi:** 10.14201/ADCAIJ2020927990.

[25] T. Takase, S. Oyama, and M. Kurihara, “Effective neural network training with adaptive learning rate based on training loss,” *Neural Networks*, vol. 101, pp. 68–78, May 2018, **doi:** 10.1016/j.neunet.2018.01.016.

[26] “Destinations Ratings Based Multi-Criteria Recommender System for Indonesian Halal Tourism Game,” *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 1, Feb. 2022, **doi:** 10.22266/ijies2022.0228.26.

[27] H. S. Ramadhan and E. Budi Setiawan, “Social Media Based Film Recommender System (Twitter) on Disney+ with Hybrid Filtering Using Support Vector Machine,” *sinkron*, vol. 8, no. 4, pp. 2215–2225, Oct. 2023, **doi:** 10.33395/sinkron.v8i4.12876.

[28] T. Takase, S. Oyama, and M. Kurihara, “Effective neural network training with adaptive learning rate based on training loss,” *Neural Networks*, vol. 101, pp. 68–78, May 2018, **doi:** 10.1016/j.neunet.2018.01.016.

[29] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature,” *Geosci Model Dev*, vol. 7, no. 3, pp. 1247–1250, Jun. 2014, **doi:** 10.5194/gmd-7-1247-2014.

[30] N. Fatima, “Enhancing Performance of a Deep Neural Network: A Comparative Analysis of Optimization Algorithms,” *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 9, no. 2, pp. 79–90, Jun. 2020, **doi:** 10.14201/ADCAIJ2020927990.