Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Optimizing the Learning Rate Hyperparameter for Hybrid BiLSTM-FFNN Model in a Tourism Recommendation System

Aufa Ab'dil Mustofa[1], Erwin Budi Setiawan[2]

[1,2] Informatics, School of Computing, Telkom University
[1,2] Jl. Telekomunikasi 1 Terusan Buah Batu, Dayeuhkolot, Bandung, West Java 40257, Indonesia

Article:
Accepted: June 12, 2024
Revised: February 12, 2024
Issued: October 29, 2024

***Correspondence Address:**
aufaaslt@student.telkomuniversity.ac.id

ABSTRACT

Indonesia, with its abundant natural resources, is rich in captivating tourist attractions. Tourism, a vital economic sector, can be significantly influenced by digitalization through social media. However, the overwhelming amount of information available can confuse tourists when selecting suitable destinations. This research aims to develop a tourism recommendation system employing content-based filtering (CBF) and hybrid Bidirectional Long Short-Term Memory Feed-Forward Neural Network (BiLSTM-FFNN) model to assist tourists in making informed choices. The dataset comprises 9,504 rating matrices obtained from tweet data and reputable web sources. In various experiments, the hybrid BiLSTM-FFNN model demonstrated superior performance, achieving an accuracy of 93.36% following optimization with the Stochastic Gradient Descent (SGD) algorithm at a learning rate of about 0.193. The accuracy, after applying Synthetic Minority Over-sampling Technique (SMOTE) and fine-tuning the learning rate hyperparameter, showed a 14.3% improvement over the baseline model. This research contributes by developing a recommendation system method that integrates CBF and hybrid deep learning with high accuracy and provides a detailed analysis of optimization techniques and hyperparameter tuning.

Keywords: *BiLSTM; content-based filtering; feedforward neural network; TF-IDF; recommendation system; classification;*

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

1. INTRODUCTION

Indonesia's rich natural resources make it have numerous interesting tourist attractions in various regions. Tourism is currently one of the most influential and productive economic sectors for several regions and is regarded as important because it can affect the economy of the country [1]. This sector is very effective and plays a major role in the country's foreign exchange earnings, especially because Indonesia offers a variety of beautiful tourist destinations [2]. This is evidenced by data from Badan Pusat Statistika (BPS), cumulatively foreign tourist visits in 2023 increased 98.30 percent compared to the same period in 2022 [3]. A lot of changes have been made to advance the tourism sector, such as digitalization through various social media platforms, such as Twitter, which has had a good impact on the advancement of the tourism sector in Indonesia. However, the amount of information available can confuse tourists when choosing the place they want to visit. In addition, with social media algorithms that only bring up trends in tourist attractions that are viral, then, of course, information from tourism will only revolve around that place within a certain period of time [2]. Travelers often need advice and recommendations for places to visit when travelling to an area [4]. Therefore, a recommendation system is needed that can be used to provide recommendations for tourist attractions to overcome these problems.

Recommendation systems are useful in providing recommendations for tourist attractions that match the likes or preferences of tourists. The methods commonly used in recommendation systems are collaborative filtering (CF), content-based filtering (CBF), and hybrid approaches [5]-[7]. CBF is a recommender method that is performed based on the similarity of the user's preferred items [6]. In the CBF process, all item information is classified into different item profiles based on their respective descriptions [7].

In addition, the utilization of deep learning can support this, especially for classification tasks. Classification is one of the fundamental tasks in natural language processing with a wide field of applications [8]. Deep learning-based recommendation systems are increasingly in demand because they offer superior performance and can provide high-quality recommendations [6]. In deep learning, there are various methods for decision making, one of which is a recommendation system. Feedforward Neural Network (FFNN) is one of the simplest and most commonly used types of deep learning for classification tasks [9]. FFNN is a form of combined supervised learning and deep learning that belongs to an architecture consisting of multiple non-linear processing layers and is considered a good choice in recommendation systems. On the other hand, BiLSTM (Bidirectional Long Short-Term Memory) is an artificial neural network model that is able to consider context information from both directions, past and future, resulting in a more comprehensive representation of the text. The advantage of BiLSTM lies in its ability to capture temporal relationships in sequence data, which makes it highly effective in sentiment analysis tasks [10]. Recommender systems that use deep learning can provide more accurate and relevant recommendations because they are able to manage and analyze large with high complexity.

A literature review was conducted to support this research. Several studies that are aligned with the methods that have been done previously are used as references in this research. In research [11], a CBF model was developed to predict movie popularity based on initial features such as genre and director, achieving high accuracy with data from IMDb and TMDb. Research by S. Missaoui et al. [12] used CBF in the LOOKER application for filtering tourism content on social media based on multi-layered user profiles to recommend hotels, restaurants, and tourist attractions. CBF strategy was also used to recommend movies [13], with an analysis of previous user behaviour to provide more personalized recommendations. In addition, CBF was applied to an e-learning system to generate recommendations that match students' interests [14], showing significant improvements in MAE by 25.26% and accuracy by 93%.

On the other hand, research by Mingsheng Fu et al. [15] developed a FFNN model to describe the interaction between users and items, which can capture various types of relations. The model showed high performance with an RMSE of 0.843, outperforming previous methods. In addition, the research of B. Markapudi et al. [16] developed a video recommendation system to overcome semantic gaps by utilizing visual features and neural networks, using Motion Adaptive Gaussian Denoising Filtering and Multilayer Feed-Forward. Implementation with MATLAB r2020a showed a computation time of about 0.999 seconds and 94% accuracy. In addition, the sentiment analysis method proposed in this study integrates the degree of contribution of sentiment information into the TF-IDF algorithm for term weight calculation, resulting in a better word vector representation. The BiLSTM model is used to consider the context information thoroughly and obtain a more accurate text representation, with the sentiment results determined through FFNN and softmax mapping [10].

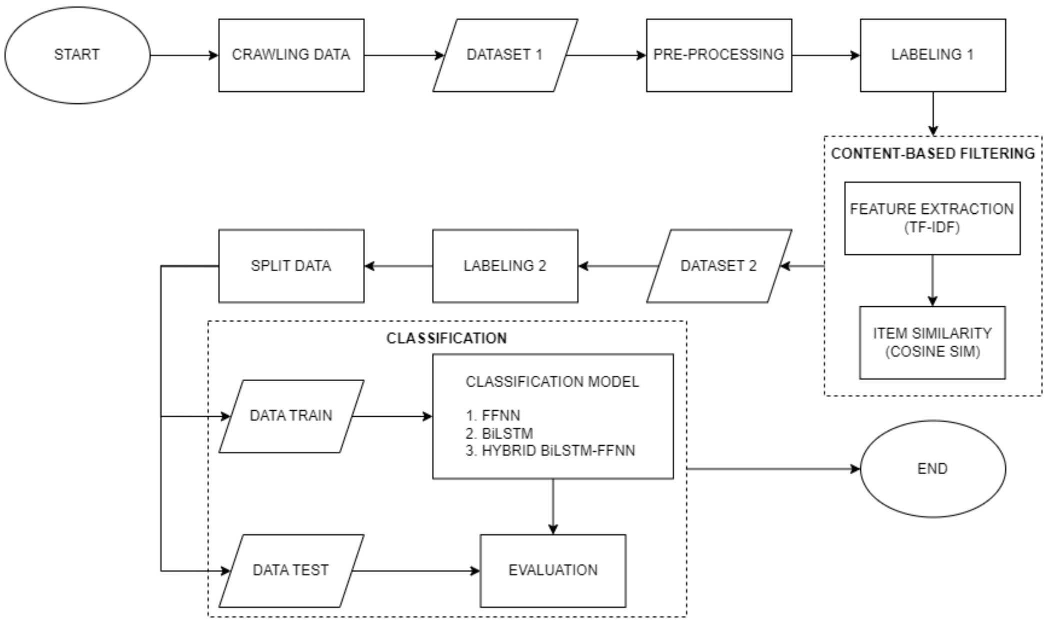

Figure 1. *The model development workflow*

Classification models in carrying out their tasks usually experience various problems, such as data balance problems. SMOTE (Synthetic Minority Over-sampling Technique) is a technique used to overcome the problem of class imbalance in datasets in the classification process [17]. This problem often occurs when the minority class in the dataset has a much smaller number of samples compared to the majority class, which can cause the classification model to favor the majority class and ignore the minority class. As in the research conducted by J. Shen et al., using SMOTE on the Support Vector Machine model optimized with Particle Swarm Optimization (PSOSVM) proposed successfully increased the accuracy by 38.29% from the model without SMOTE.

In this research, the tourism recommendation method will use a Content-Based Filtering and hybrid BiLSTM Feed-Forward Neural Network (BiLSTM-FFNN). The dataset used is tweet data from Twitter, because Twitter is considered a popular platform for expressing opinions, providing reviews, and providing recommendations [18]. This research aims to fill the knowledge gap, as there is no research on the use of CBF method with BiLSTM-FFNN hybrid in the development of travel recommendation system. This indicates the need for further experiments to explore the potential of this integration method in depth. This research contributes by developing a travel recommendation system method that integrates CBF and hybrid deep learning BiLSTM-FFNN with high accuracy and provides a comprehensive analysis of various hyperparameter optimization and tuning with Learning Rate Finder (LFR).

2. METHODS

In this research, a tourism recommendation system is built by applying a combination of two different methods. Based on Figure 1, CBF is performed by applying TF-IDF feature extraction, which then uses deep learning models FFNN, BiLSTM, and Hybrid BiLSTM-FFNN to perform the classification process.

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

2.1. Data Crawling

In this research, the data source used is information about tourist attractions in the Bandung area. The first dataset was obtained by submitting a request to the Ministry of Tourism and Creative Economy. In addition, data retrieval from the Twitter platform was carried out using crawling techniques on the Twitter API. The data retrieved includes tweets, user IDs, and information related to tourist attractions. Datasets were also obtained through crawling websites such as Traveloka, Google Maps, and others. This process produces datasets in the form of Comma Separated Values (CSV) files. The dataset obtained is divided into two parts, where dataset 1 consists of raw data that has not gone through the preprocessing stage. Meanwhile, dataset 2 is the result of applying content-based filtering that includes complete user ratings.

2.2. Data Preprocessing

Data preprocessing is an important stage to prepare it for model building [6]. The initial data goes through a series of processes to make it clean, informative, and ready to be used in the next stage. Some of the steps taken in data preprocessing include:

a. Data cleaning: The process of removing irrelevant or corrupted data, aiming to create a cleaner dataset, such as punctuation marks, irrelevant numbers, emoticons, URLs, and others.
b. Case Folding: The transformation of all letters into lowercase letters, aims to solve the problem of inconsistent writing.
c. Stopword removal: Removing words that do not provide special information or meaning.
d. Tokenization: The process of separating sentences into smaller units (tokens) to facilitate further processing.

Table 1. *Example of data preprocessing*

Preprocessing	Text
Original Tweet	@mhdarieff_ Kadang aku pulang kerja ato pulang dari mana, melipir dulu naik dan puter balik di Tangkuban perahu 😅😅😅😅 definitely udah gak keitung berapa kali
Data Cleaning	Kadang aku pulang kerja ato pulang dari mana melipir dulu naik dan puter balik di Tangkuban perahu udah gak keitung berapa kali
Case Folding	kadang aku pulang kerja ato pulang dari mana melipir dulu naik dan puter balik di tangkuban perahu udah gak keitung berapa kali
Stopword Removal	kadang pulang kerja ato pulang melipir naik puter balik tangkuban perahu udah keitung kali
Tokenization	['kadang', 'pulang', 'kerja', 'ato', 'pulang', 'melipir', 'naik', 'puter', 'balik', 'tangkuban', 'perahu', 'udah', 'keitung', 'kali']

2.3. Data Labeling

The process of rating each tweet, which includes user reviews or opinions about tourist attractions, is done by counting words with positive and negative connotations. This labeling process is divided into two parts. For labeling 1, the tweet data is given a rating value between 1 and 5. Values close to 1 indicate negative aspects, while values close to 5 reflect positive aspects. In labeling 2, the rating data that initially ranges from 1 to 5 is converted to 0 and 1 [6], [19]. This conversion is done to adjust the data into the input format required for the classification process.

2.4. Content-Based Filtering

Content-Based Filtering (CBF) is an approach or method that provides item recommendations to users by considering the characteristics of items that are similar to items that have been accessed or interacted with by users before [6], [20]. In order for the system to utilize the CBF method in the implementation of the recommendation system, information regarding user preferences is required. This method involves analyzing the dataset evaluated by each user, taking into account the rating and quality of the data and creating a user profile that becomes the basis for suggesting additional tourist attractions that match their preferences.

The CBF process uses the dataset generated from the pre-processing of dataset 1. The process starts by using TF-IDF (Term Frequency-Inverse Document Frequency) to extract features from item attributes such as name, category and description of tourist attractions. TF-IDF measures the value of how important a word is in a document based on its frequency but compensated by the frequency of the word in the whole dataset. The TF-IDF formula can be seen below [6], [21]:

$$W_{i,z} = TF_{i,z} \times IDF \quad (1)$$

$$\mathrm{IDF} = log\,\frac{N}{DF_z} \quad (2)$$

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

Term Frequency (TF) is the simplest way to give weight to terms (words). The weight of a word T in a document is given using equation (1). Whereas Inverse Document Frequency (IDF) is a count of word occurrences in a set of documents as in equation (2). Thus, the description of tourist attractions can be represented as a numerical feature vector that reflects the word weight.

Furthermore, the similarity between items is calculated using cosine similarity to overcome the condition where users have different rating schemes [22]. This process allows the system to recommend tourist attractions to users based on the similarity of positively labeled content. The calculation of cosine similarity is formulated in equation (3).

$$\cos \text{sim} = \frac{\sum_{i=1}^{n} A_i B_i}{\sum_{i=1}^{n} A_i^2 \ \sum_{i=1}^{n} B_i^2} \qquad (3)$$

Cosine similarity calculates the cosine angle between two feature vectors A and B, where A_i and B_i are the TF-IDF weights of word i in items A and B respectively. This formula allows the system to determine the similarity between two items based on both rated and unrated content, with the result value ranging from -1 to 1, where 1 indicates perfect similarity.

The result of the CBF process is a recommendation system that can recommend tourist attractions to users based on positive content similarity. By using TF-IDF feature vector representation and item similarity, the system can determine the extent to which a tourism is similar to the tourism preferred by the user. This allows users to get recommendations that are more personalized and in accordance with their preferences. In this research, the recommendation results from CBF in the form of a rating matrix will be labeled into binary to match the input of the classification model in the next process.

2.5. Feedforward Neural Network

In providing tourist attractions recommendations, Feedforward Neural Network (FFNN) has an important role as a method for classification. In recent research, it was found that in classification methods Neural Network (NN) is a promising choice as an alternative to various conventional classification methods [23]. FFNN is one type of NN that is considered the most frequently studied and used neural network classifier [23], [24]. FFNN is the basic model of a modified Recurrent Neural Network (RNN) [6], [24]. The common FFNN assumes that all inputs are independent of each other, but the hidden layer of the RNN depends on prior information caused by repeated edges [25].

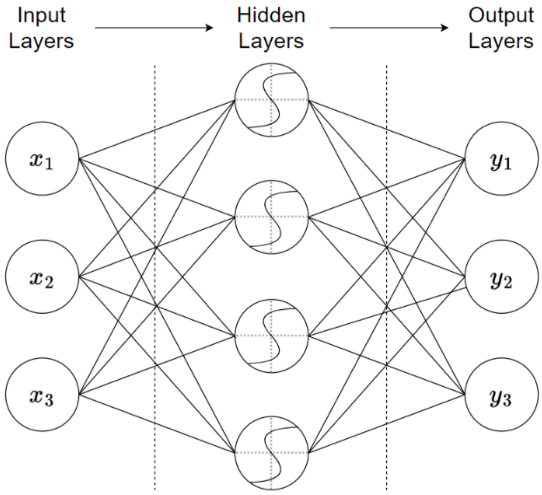

Figure 2. *Feedforward neural network architecture*

Based on Figure 2, the structure of the FFNN architecture itself consists of three main layers, namely the input layer to receive input data, the hidden layer as the hidden layer for processing, and the output layer that produces the final output [26]. In FFNN, training parameters such as the number of epochs, momentum, and learning rate are set based on training experience [16]. In the forward stage, the value of each hidden node is evaluated from the sum of the product of the corresponding connection weights and the neuron input values. This process is also applied to the output value by utilizing the connection weights and hidden node values. Furthermore, in the backward stage, the difference between the output response value and the target value is estimated and updates are made to the connection weights to minimize the error [16].

$$y_n = f(\sum_{j=1}^{H} f\,((\sum_{i=1}^{M} x_i w_{i,j} b_{oj}) w_{j,n}) + b_{on}) \qquad (4)$$

Equation (4) is used to calculate the feedforward according to Figure 2. The symbol y_n indicates the output value at the output layer. f represents the activation function applied to the layer. The symbols H and M refer to the hidden layer units 1 and 2, where j, i, and n are

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

the number of neurons in each layer. Input data is denoted by the symbol x, while w denote weight values, while b denote bias values [26]. Through training, FFNN could update its internal parameters based on new data. In this research, optimization was carried out using Adam, Stochastic Gradient Descent (SGD), and RMSprop to improve the accuracy of recommendations. The output of FFNN provided a prediction of the level of suitability of tourist attractions with user preferences and helped provide more personalized recommendations. The FFNN model learned complex patterns and optimized the classification of tourist attractions, providing users with recommendations that were increasingly tailored to their preferences.

2.6. Bidirectional Long Short-Term Memory

Bidirectional Long Short-Term Memory (BiLSTM) is an artificial neural network model that overcomes the limitations of conventional LSTMs that only process information in one direction. In a standard LSTM, information only flows forward in time order, so it cannot fully utilize information that lies ahead in a sequence. In contrast, BiLSTM adds a back propagation layer that allows the model to process information from both the past and the future simultaneously. By utilizing two recurrent neural networks in the positive and negative directions, BiLSTM can capture long-term dependencies in historical signals as well as future long-term information in the same signal [27].

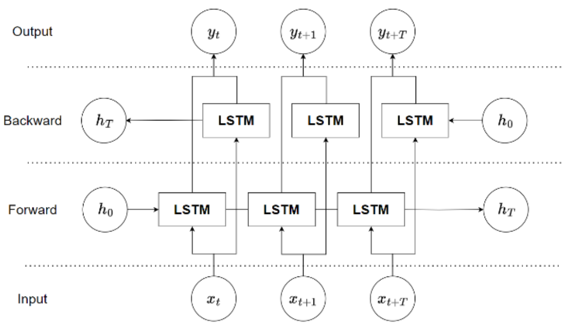

Figure 3. *Bidirectional long short-term memory structure*

Based on Figure 3, the BiLSTM model consists of two hidden layers, namely forward and backward LSTMs, which run in positive and negative directions [27]-[29]. The forward propagation process extracts long-term dependency information from historical signals, while the backward process extracts future information. The output of the same neuron connects the two LSTM meta cells, resulting in two combined hidden states. These hidden states are then passed to the fully connected layer for non-linear mapping and to the regression layer for final prediction. Thus, the BiLSTM can exploit information from two-time directions, past and future [28].

2.7. Hybrid BiLSTM-FFNN

Through the combination of several deep learning techniques, Hybrid Deep Learning builds more complex and efficient models to solve various problems [30]. In this research, a hybridization is performed by combining BiLSTM and FFNN. BiLSTM can capture long-term dependency information of words and positions in sentences. BiLSTM networks, which are often used today, can identify long-term relationships between words from beginning to end and from end to beginning. This bidirectional processing of input data creates additional workload calculations [31]. In addition, FFNN exhibits high architectural flexibility, making it an effective tool for excessive input data processing, which can usually result in poor test performance [32].

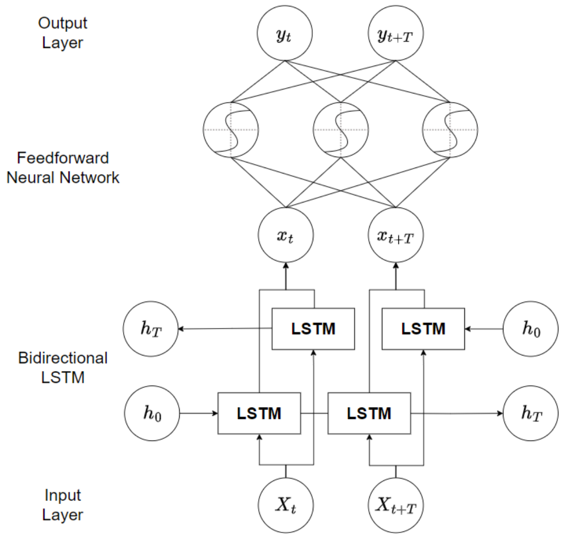

Figure 4. *Hybrid BiLSTM-FFNN architecture*

Based on Figure 4, hybrid BiLSTM-FFNN architecture combines the advantages of both types of neural networks to create more complex and effective models in dealing with different types of sequential problems. In the first stage, BiLSTM is used to extract and understand long-term dependency information

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

from historical and future data. BiLSTM consists of two LSTM layers: one for processing forward information and another for processing backward information. Each LSTM layer outputs two hidden states, which represent information from the forward and backward directions, respectively. Next, the outputs of BiLSTM, that is, the two hidden states of each direction, are passed to the FFNN layer. The FFNN layer is responsible for performing further non-linear mapping of the intermediate states generated by the BiLSTM.

The combination utilizes the BiLSTM's ability to capture information from both time directions (past and future) with reliability of feed-forward networks for multi-step non-linear prediction, resulting in more robust model for natural language processing and time series forecasting tasks [33]. Using this architecture, the model can utilize the power of BiLSTM in understanding sequential context and the power of feed-forward in providing stable and accurate predictions.

2.8. Model Optimization

Deep learning is evolving rapidly as various model designs ranging from simple to complex emerge. Therefore, neural network optimization becomes crucial to improve the performance and efficiency of the model. Neural network optimization is the process of improving the performance and efficiency of the model by making various adjustments and changes to achieve higher accuracy [34]. Optimization algorithms that have numerous variants utilize the weights and learning speed of the model during the training process to minimize the loss function and maximize accuracy. In this research, seven model optimization algorithms were used, which are described as follows [35]:

a. Adam: optimization that uses moving averages of the first and second gradients to adapt the learning rate, accelerate convergence and improve stability in training.
b. Stochastic Gradient Descent: optimization that updates model parameters iteratively based on random samples of training data, making calculations more efficient.
c. RMSprop: optimization that uses the exponential weighted average of the squares of the gradients to adapt the learning rate, reduce vertical oscillations and improve stability in training.
d. Nadam: optimization that combines Nesterov-accelerated gradient with Adam, updating parameters by utilizing momentum from the next step during the previous step's update to improve convergence speed and stability.
e. Adamax: variation of Adam that uses an infinite norm to control the learning rate, making it more stable under large gradient conditions.
f. Adagrad: optimization that adaptively adjusts the learning rate based on the sum of squares of the historical gradients of each parameter, enabling faster learning rate reduction.
g. Adadelta: optimization that updates the learning rate adaptively without requiring a fixed initial learning rate, by using a fixed window size to consider the sum of squares of past gradients.

2.9. Learning Rate

In the optimization process, the learning rate is a hyperparameter that controls the amount of change in the model's weights each time it is updated during training [6]. A small learning rate allows for faster training and good testing performance. Models with a large learning rate avoid overfitting on easy-to-remember patterns and generalize better on difficult patterns [36]. However, when the learning rate is set large, it causes undesirable divergent behavior in the loss function. The loss function is a metric used to measure how well the model predicts the true target value, with higher values indicating less accurate predictions. Therefore, when the highest learning rate is applied to complex and large problems, there is a negative impact on the training process and accuracy [37].

Therefore, in this research, hyperparameter tuning is performed to find the best learning rate using Learning Rate Finder (LFR). LFR is a technique used to determine the optimal learning rate for training, with the aim of ensuring fast and efficient convergence without causing divergence or poor performance issues [6]. The process begins by setting the model to undergo training with a wide range of learning rates. During training, the learning rate is changed exponentially at each iteration, and losses are recorded for each value of learning rate. The graph shows that the loss will decrease at a certain point when the optimal learning rate is chosen, whereas too high a learning rate will cause the loss to increase, signaling divergence. A learning rate value that provides a steady decrease in loss without divergence can be selected as the optimal one. This technique helps in identifying the learning rate that can improve the training speed and accuracy of the model [6].

2.10. Evaluation

In the evaluation stage, this research involves confusion matrix to evaluate the system's ability to classify and recommend or not recommend an item to the user. This confusion matrix combines the prediction results with the correct label values after the classification process. Accuracy describes the proportion of correct predictions out of all predictions made by the model [38]. Table 2 shows the confusion matrix and Equation 5 represents the calculation of accuracy.

Table 2. *Confusion matrix*

Recommendation	Actual	
	TRUE	**FALSE**
TRUE	True Positive (TP)	False Positive (FP)
FALSE	False Negative (FN)	True Negative (TN)

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

3. RESULTS AND DISCUSSION

This section is divided into three parts: data preparation, recommendation system, and classification. The data preparation part includes data crawling and preprocessing. In the recommendation system part, the main goal is to predict the ratings. In the last part, classification is performed with proposed models.

3.1. Data Preparation Result

In this research, tourism data was obtained from the Ministry of Tourism and Creative Economy with permission. After that, the list of tours was used as keywords to crawl through the Twitter API. The results of crawling tweets were 917,840 data entries with 16 columns, including username, full text, and others. Furthermore, crawling data was obtained from the web from six websites, including the rating of each tour. After that, preprocessing was done, and the three data sources were combined, resulting in 9,504 ratings. Tweets in the form of text were processed into numerical ratings using the TextBlob library. Twitter crawling results can be seen in Table 3.

Table 3. *Twitter crawled dataset*

username	cleaned text	rating
Indotravelista	tangkuban perahu tempat wisata dingin melegenda	3
Beritasatu	gunung tangkuban perahu diserbu ribuan wisatawan pengunjung rela naik motor	3
…	…	…
letsgoeat	annyeonghaseo chingu korean fans cafe chinguid jl prof eyckman no bandung open monsun	2
chinguID	iya baru di bandung aja chingu dmn main yuk kesini crstla chinguid chingu cafe cuman ada dibandung aja ya	3

3.2. Recommendation System Result

In the CBF method, this research used TF-IDF to calculate the occurrence of words in the dataset. The preprocessed dataset is converted into a matrix that associates users with tourist attractions. This research used important words from a combination of features such as place name, category, and description from tourism data. At the same time, IDF indicated the frequency of word occurrence. The output of the TF-IDF process produced a matrix that represented the tour and its features. After that, cosine similarity was applied, and rating prediction was done using the representation vector value of the similarity value. The results of the rating prediction represented the user's assessment of the tour, as shown in Table 4.

Table 4. *Recommendation system result*

Username	Place Name				
	Gunung Tangkuban Perahu	**Jalan Braga**	**…**	**Kota Mini**	**Chingu Cafe Little Seoul**
Trip Advisor	3.5	4.0	…	2.5	3.0
Google Maps	4.5	4.8	…	4.3	4.4
…	…	…	…	…	…
bandung 911	0.5	0.7	…	3.0	0.5
GEMAWA HYUH	0.9	1.0	…	0.8	0.8

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

3.3. Classification Result

This section presents the results of several experiments conducted on the proposed classification model, Hybrid BiLSTM-FFNN, and the independently built models namely BiLSTM and FFNN. The models used each have 64 units, a dropout of 0.3, activation using sigmoid, loss function using binary crossentropy, and trained for 50 epochs.

These classification models were tested, and several experimental scenarios were conducted to assess the accuracy results, where a significant increase in accuracy over the baseline model was expected. The first scenario determined the optimal test size for the model without optimization and hyperparameter tuning. In the second scenario, SMOTE was implemented to check the balance of the data. In the third scenario, the model was optimized using the default parameters and the most optimal test size obtained in the second scenario. In the fourth scenario, the model optimization was tuned with the Learning Rate Finder. The results are presented separately for each scenario.

3.3.1. First Scenario Result

In the first scenario, an experiment was conducted to determine the optimal test size for the model. This was done by comparing the accuracy values of each method against different test sizes, namely 10%, 20%, 30%, and 40%. The most optimal data size is the one that gives the highest accuracy. Table 5 shows the results obtained by testing the methods without optimization to obtain a baseline. The results show that the FFNN model obtained the highest accuracy of 78.65% at a ratio of 90:10, the BiLSTM model obtained the highest accuracy of 80.56% at a ratio of 80:20, and the Hybrid BiLSTM-FFNN model obtained the best accuracy at a ratio of 80:20 of 81.70%.

Table 5. *First scenario result*

Model	Data Split Ratio	Accuracy (%)
Feedforward Neural Network	**90:10**	**78.65**
	80:20	78.42
	70:30	78.06
	60:40	77.74
Bidirectional LSTM	90:10	80.15
	80:20	**80.56**
	70:30	80.11
	60:40	79.95
Hybrid BiLSTM-FFNN	90:10	81.28
	80:20	**81.70**
	70:30	80.71
	60:40	80.62

3.3.2. Second Scenario Result

In the second scenario, the model with all test sizes is tested again by applying SMOTE. SMOTE is an oversampling technique used to obtain optimal classification results [17]. SMOTE is applied with the aim of overcoming unbalanced data by resampling with oversampling techniques. SMOTE can improve classification performance for minority classes or classes with the least amount. The results show that there is an improvement in each accuracy for all test sizes. The 10% test size achieved the highest accuracy compared to the other test sizes. The FFNN model achieved an accuracy of 80.97%, which is up 2.95% from the baseline, while the BiLSTM model achieved an accuracy of 82.64%, which is up 3.11% from the baseline. In addition, the proposed hybrid BiLSTM-FFNN model obtained the highest accuracy of the other models at 84.48%, which was up 3.94% from the baseline in the first scenario. Each of the models with the best accuracy in implementing SMOTE will be used in the next scenario. The results of the second scenario can be seen in Table 6.

Table 6. *Second scenario result*

Model	Data Split Ratio	Accuracy (%)
Feedforward Neural Network	**90:10**	**80.97 (+2.95)**
	80:20	80.57 (+2.74)
	70:30	80.48 (+3.10)
	60:40	80.22 (+3.19)
Bidirectional LSTM	**90:10**	**82.64 (+3.11)**
	80:20	82.58 (+2.51)
	70:30	82.23 (+2.65)
	60:40	82.08 (+2.66)
Hybrid BiLSTM-FFNN	**90:10**	**84.48 (+3.94)**
	80:20	84.35 (+3.24)
	70:30	84.13 (+4.24)
	60:40	83.90 (+4.07)

3.3.3. Third Scenario Result

In the third scenario, the best model from the previous scenario is tested again by applying an optimization algorithm. Optimization algorithms are the basis of the machine's ability to learn from experience by calculating the Gradient to minimize the loss function through various optimization algorithms. There are several optimizations that can be used in classification models, including Adam, SGD, RMSprop, Nadam, Adamax, Adagrad, and Adadelta.

Table 7 shows the accuracy results of the optimized model. The FFNN model optimized with RMSprop produces the highest accuracy, which is 83.21%, up 5.8% from the baseline,

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

while the BiLSTM model optimized with Adamax produces an accuracy of 85.78%, up 6.48% from the baseline. In addition, the BiLSTM-FFNN hybrid model obtained the highest accuracy with Adam optimization compared to other optimizations, which was 86.59%, up 5.99% from the baseline. In contrast, the performance results of all models optimized with SGD decreased in accuracy. This is because the default learning rate setting of 0.001 for the optimization algorithm does not provide optimal performance, so hyperparameter tuning is required, which will be done in the next scenario.

Table 7. *Third scenario result*

Model	Optimization	Accuracy (%)
Feedforward Neural Network	Adam	82.52 (+4.92)
	SGD	75.41 (-4.12)
	RMSprop	**83.21 (+5.80)**
	Nadam	81.08 (+3.09)
	Adamax	81.13 (+3.15)
	Adagrad	79.60 (+1.21)
	Adadelta	79.58 (+1.18)
Bidirectional LSTM	Adam	83.95 (+4.21)
	SGD	80.47 (-0.11)
	RMSprop	83.68 (+3.87)
	Nadam	80.93 (+0.46)
	Adamax	**85.78 (+6.48)**
	Adagrad	84.36 (+4.72)
	Adadelta	84.96 (+5.46)
Hybrid BiLSTM-FFNN	**Adam**	**86.59 (+5.99)**
	SGD	80.24 (-1.79)
	RMSprop	84.87 (+3.88)
	Nadam	86.16 (+5.46)
	Adamax	85.37 (+4.49)
	Adagrad	85.27 (+4.37)
	Adadelta	84.06 (+2.89)

3.3.4. Fourth Scenario Result

The fourth scenario is the last experiment, which involves performing hyperparameter tuning to achieve improved accuracy. The model from the previous scenario is used in this scenario by tuning the hyperparameters using the Learning Rate Finder. The learning rate is a hyperparameter in the optimizer that controls how much the model weights change each time the model is updated during training [5]. During this process, the learning rate is changed exponentially, and the loss is recorded for each value. The resulting graph shows how the loss changes as the learning rate changes, helping to select the optimal learning rate to ensure rapid convergence and avoid divergence. Therefore, choosing the right learning rate is crucial to ensure efficient training and optimal model convergence [5]. The results of this scenario in the form of learning rate values for each optimization can be seen in Table 8. The results show that for all models, there is a significant increase in accuracy after hyperparameter tuning.

Based on Table 8, the FFNN model optimized with Adamax with a learning rate of 8.48342898244072e-07 resulted in the highest accuracy of 92.30%, up 17.3% from the baseline. The BiLSTM model optimized with Nadam experienced a 15% increase from the baseline with a learning rate of 6.10540229658532e-04 with an accuracy of 92.69%. In the BiLSTM-FFNN hybrid model, the highest accuracy was 93.36% using SGD optimization with a learning rate of 1.93069772888324e-01. This result increased by 14.3% from the baseline and outperformed models with other optimizations. Adjusting the learning rate in accordance with the proportion of the model will get maximum accuracy.

Table 8. *Fourth scenario result*

Model	Optimization	Best Learning Rate	Accuracy (%)
Feed forward Neural Network	Adam	1.63789370695406 e-02	89.88 (+14.2)
	SGD	2.68269579527972 e-04	91.26 (+16.0)
	RMSprop	7.19685673001152 e-03	89.75 (+14.1)
	Nadam	1.63789370695406 e-02	91.81 (+16.7)
	Adamax	**8.48342898244072 e-07**	**92.30 (+17.3)**
	Adagrad	5.17947467923121 e-05	91.43 (+16.2)
	Adadelta	1.17876863479358 e-04	89.89 (+14.2)
Bidirec tional LSTM	Adam	3.16227766016837 e-08	88.04 (+9.29)
	SGD	3.16227766016837 e-03	91.66 (+13.7)
	RMSprop	2.27584592607479 e-05	91.65 (+13.7)
	Nadam	**6.10540229658532 e-04**	**92.69 (+15.0)**
	Adamax	1.17876863479358 e-09	91.55 (+13.6)
	Adagrad	1.93069772888324 e-06	90.32 (+12.1)
	Adadelta	4.39397056076079 e-06	91.70 (+13.8)
Hybrid BiLSTM-FFNN	Adam	6.10540229658533 e-04	90.71 (+11.0)
	SGD	**1.93069772888324 e-01**	**93.36 (+14.3)**
	RMSprop	3.16227766016838 e-03	90.38 (+10.6)
	Nadam	3.72759372031494 e-02	92.08 (+12.7)
	Adamax	1.38949549437313 e-03	91.78 (+12.3)
	Adagrad	8.48342898244072 e-02	90.99 (+11.3)
	Adadelta	4.39397056076079 e-01	90.17 (+10.3)

3.4. Discussion

This research emphasizes the importance of meticulous data preparation and comprehensive preprocessing prior to the implementation of a recommendation and classification system. Tourism data collected from various sources, including Twitter API and websites, undergoes a processing process involving data crawling and preprocessing to ensure cleanliness and optimal data quality. Next, the recommendation system is implemented using the CBF method, using TF-IDF to extract important features from the tourism data. The main goal of the system is to predict tour ratings based on the text of tweets retrieved from the Twitter API and other data from the website, which are then converted into numerical values using the TextBlob library. The result of this recommendation system is a rating number in the form of a matrix between users and tourist attractions, which will then be labelled into binary form before entering the classification process.

Classification is performed by comparing several proposed models. The models used each have 64 units, a dropout of 0.3, activation using sigmoid, loss function using binary crossentropy, and trained for 50 epochs. The first experiment (first scenario) aims to determine the optimal test size without optimization and hyperparameter tuning. The tested models include the Hybrid BiLSTM-FFNN and the BiLSTM and FFNN models independently. The results of this scenario show that the Hybrid BiLSTM-FFNN model provides the best performance with accuracy reaching 81.70% at a data ratio of 80:20. While the FFNN model obtained the highest accuracy of 78.65% at a ratio of 90:10, and the BiLSTM model obtained the highest accuracy of 80.56% at a ratio of 80:20.

The second scenario involves applying the SMOTE oversampling technique to address data imbalance. The aim is to improve classification performance on minority classes or underrepresented data. The results show significant improvement in all models tested, with the Hybrid BiLSTM-FFNN model achieving the highest accuracy of 84.48%. The FFNN model obtained an accuracy of 80.97%, and the BiLSTM model obtained an accuracy of 82.64%. All three models get the highest accuracy at a data ratio of 90:10. This accuracy improved from the baseline model and was used in the next scenario by applying the optimization algorithm.

The third scenario experiment was conducted by applying model optimization using various optimization algorithms such as Adam, Stochastic Gradient Descent (SGD), RMSprop, Nadam, Adamax, Adagrad, and Adadelta. In this scenario, some optimization algorithms do not work well; even SGD optimization has decreased accuracy. This happens because the optimization algorithms are applied with default parameters. Therefore, the fourth experiment applied a hyperparameter tuning process to find the best learning rate for each optimization algorithm used. The experimental results showed that the optimized models were able to achieve a significant improvement in accuracy compared to the baseline, with the highest accuracy BiLSTM-FFNN hybrid model of 93.36% using SGD optimization with a learning rate of 1.93069772888324e-01. Up to 14.3% from the baseline. The FFNN model optimized with Adamax with a learning rate of 8.48342898244072e-07 resulted in an accuracy of 92.30%, up 17.3% from the baseline. In addition, the BiLSTM model optimized with Nadam has increased by 15% from the baseline with a learning rate of 6.10540229658532e-04 with an accuracy of 92.69%.

Furthermore, a statistical significance analysis was conducted to check whether the change in accuracy was indeed a significant change. This calculation was performed to check the increase in each scenario. The parameters used are P-Value and Z-Value. The P-Value assesses the possibility of a non-coincidental change. A P-Value of less than 0.05 indicates significance. In addition, at the 95% confidence level, a Z-Value greater than 1.96 indicates that the difference is significant [30]. The results of the statistical significance analysis are presented in Table 9, while Figure 5 visualizes the accuracy improvement of each tested scenario.

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (112-126)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.40250

Table 9. *Statistical significance test result*

Model	Parameters	Scenarios			
		S1→S2	**S2→S3**	**S3→S4**	**S1→S4**
Feedforward Neural Network	Z-Value	79.57	354.17	677.52	386.08
	P-Value	0.0	0.0	0.0	0.0
	Significant?	**True**	**True**	**True**	**True**
Bidirectional LSTM	Z-Value	556.43	641.35	1726.50	2379.67
	P-Value	0.0	0.0	0.0	0.0
	Significant?	**True**	**True**	**True**	**True**
Hybrid BiLSTM-FFNN	Z-Value	213.21	286.86	657.94	1586.99
	P-Value	0.0	0.0	0.0	0.0
	Significant?	**True**	**True**	**True**	**True**

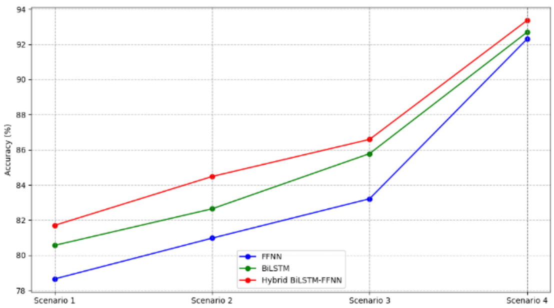

Figure 5. *Graph of increase in accuracy*

The results of the experiments conducted show that the approaches taken in the scenarios performed exhibit advantages in improving the accuracy of the classification model. In particular, the use of the SMOTE technique to address data imbalance and hyperparameter optimization by tuning the learning rate significantly improved the model performance. The Hybrid BiLSTM-FFNN model proved to be the most effective in this case, achieving the highest accuracy compared to the other models tested. Thus, this approach not only validates the research hypothesis but also provides a solid foundation for practical implementation in tourism recommendation.

CONCLUSION

This research develops a tourism recommendation system using Content-Based Filtering (CBF) method with the help of TF-IDF feature extraction and hybrid deep learning for classification. By utilizing tweet data from Twitter collected using the Twitter API, as well as data from trusted web sources, this system processes the data through a comprehensive process to ensure data hygiene and quality. The data used is 917,840 data entries, which are then converted into 9,504 matrix ratings of users and tourist attractions. The recommendation system using CBF with TF-IDF as feature extraction successfully predicts ratings that are 0 (not yet rated by the user). The results of this process are then labeled into binary form to match the input of the classification model. The models built are FFNN, BiLSTM and BiLSTM-FFNN hybrid. The models used each have 64 units, a dropout of 0.3, activation using sigmoid, loss function using binary crossentropy, and trained for 50 epochs. Experiments were conducted four times to improve accuracy compared to the baseline. The proposed Hybrid BiLSTM-FFNN model showed the best performance, achieving the highest accuracy of 93.36% after applying model optimization with SGD, SMOTE and hyperparameter tuning, with an improvement of 14.3% from the baseline and using a learning rate of 1.93069772888324e-01. The experiments also show that the use of SMOTE oversampling technique can improve the classification performance on underrepresented data, and the application of various optimization and hyperparameter tuning with Learning Rate Finder (LFR) can significantly improve the accuracy of the models. With satisfactory results and high accuracy, this research is able to provide more accurate and relevant tourist attraction recommendations for users. This research demonstrates that the integration of CBF with a Hybrid BiLSTM-FFNN model significantly enhances the accuracy of tourism recommendations. The use of SMOTE to handle data balance issues and hyperparameter tuning using LFR contributed greatly to the increase in accuracy compared to the baseline. As a suggestion for future research, it is recommended to explore the use of data from other social media platforms and consider individual user preferences to further improve recommendation relevance and accuracy. While this research focused on Twitter data, incorporating data from diverse sources could provide a more comprehensive understanding of tourist preferences.

REFERENCES

[1] M. Ridwan, "Jurnal Ekonomi Dan Bisnis Islam Jurnal Ecopreneur Upaya Masyarakat Dalam Publikasi Destinasi Wisata Untuk Peningkatan Ekonomi Desa Leuwikujang," *Jurnal Ekonomi dan Bisnis Islam*, vol. I, no. Ekonomi, pp. 30–45, 2020.

[2] D. M. Sari and A. S, "IMPLEMENTASI SISTEM PENGENALAN WISATA BAHARI BERBASIS WEBSITE KABUPATEN MAJENE," *Jurnal Teknik Informatika*, vol. 13, no. 2, pp. 123–132, Feb. 2021, doi: 10.15408/jti.v13i2.16956.

[3] B. P. S. Indonesia, "Kunjungan wisatawan mancanegara pada Desember 2023 mencapai 1,14 juta kunjungan, naik 20,17 persen (year-on-year).," *Badan Pusat Statistik Indonesia*, Feb. 01, 2024. https://www.bps.go.id/id/pressrelease/2024/02/01/2347. [Online].

[4] A. S. Ningrum, H. C. Rustamaji, and Y. Fauziah, "CONTENT BASED DAN COLLABORATIVE FILTERING PADA REKOMENDASI TUJUAN PARIWISATA DI DAERAH YOGYAKARTA," *Telematika*, vol. 16, no. 1, p. 44, Apr. 2019, doi: 10.31315/telematika.v16i1.3023.

[5] S. Supriyanto and J. Fahana, "Possible System Architecture for Travel Recommender," *JOIN (Jurnal Online Informatika)*, vol. 5, no. 1, pp. 1–8, Jul. 2020, doi: 10.15575/join.v5i1.573.

[6] B. M. G. A. Awienoor, and E.B. Setiawan, "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] D. Roy and M. Dutta, "A systematic review and research perspective on recommender systems," *J Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00592-5.

[8] N. Buslim, L. K. Oh, M. H. A. Hardy, and Y. Wijaya, "Comparative analysis of KNN, Naïve Bayes and SVM algorithms for movie genres classification based on synopsis.," *Jurnal Teknik Informatika*, vol. 15, no. 2, pp. 169–177, Dec. 2022, doi: 10.15408/jti.v15i2.29302.

[9] A. Z. Mustaqim, N. A. Fadil, and D. A. Tyas, "Artificial Neural Network for classification task in tabular datasets and image Processing: A Systematic Literature review," *JOIN (Jurnal Online Informatika)*, vol. 8, no. 2, pp. 158–168, Dec. 2023, doi: 10.15575/join.v8i2.1002.

[10] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BILSTM," *IEEE Access*, vol. 7, pp. 51522–51532, Jan. 2019, doi: 10.1109/access.2019.2909919.

[11] S. Sahu, R. Kumar, M. S. Pathan, J. Shafi, Y. Kumar, and M. F. Ijaz, "Movie Popularity and Target Audience Prediction Using the Content-Based Recommender System," *IEEE Access*, vol. 10, pp. 42030–42046, 2022, doi: 10.1109/ACCESS.2022.3168161.

[12] S. Missaoui, F. Kassem, M. Viviani, A. Agostini, R. Faiz, and G. Pasi, "LOOKER: a mobile, personalized recommender system in the tourism domain based on social media user-generated content," *Personal and Ubiquitous Computing*, vol. 23, no. 2, pp. 181–197, Jan. 2019, doi: 10.1007/s00779-018-01194-w.

[13] S. Reddy, S. Nalluri, S. Kunisetti, S. Ashok, and B. Venkatesh, "Content-Based Movie Recommendation System using genre correlation," *in Smart innovation, systems and technologies*, 2018, pp. 391–397. doi: 10.1007/978-981-13-1927-3_42.

[14] S. Bhaskaran and R. Marappan, "Enhanced personalized recommendation system for machine learning public datasets: generalized modeling, simulation, significant results and analysis," *International Journal of Information Technology (Singapore)*, vol. 15, no. 3, pp. 1583–1595, 2023, doi: 10.1007/s41870-023-01165-2.

[15] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel Deep Learning-Based Collaborative filtering model for recommendation system," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084–1096, Mar. 2019, doi: 10.1109/tcyb.2018.2795041.

[16] B. Markapudi, K. Chaduvula, D. N. V. S. L. S. Indira, and M. V. N. S. S. R. K. S. Somayajulu, "Content-based video

recommendation system (CBVRS): a novel approach to predict videos using multilayer feed forward neural network and Monte Carlo sampling method," *Multimedia Tools and Applications*, vol. 82, no. 5, pp. 6965–6991, Aug. 2022, doi: 10.1007/s11042-022-13583-8.

[17] J. Shen, J. Wu, M. Xu, D. Gan, B. An, and F. Liu, "A Hybrid Method to Predict Postoperative Survival of Lung Cancer Using Improved SMOTE and Adaptive SVM," *Comput Math Methods Med*, vol. 2021, 2021, doi: 10.1155/2021/2213194.

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

[19] G. Ramadhan and E. B. Setiawan, "Collaborative Filtering Recommender System Based on Memory Based in Twitter Using Decision Tree Learning Classification (Case Study : Movie on Netflix)," *In: Proc. of 2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, Bandung, pp. 1-6, 2022, doi: 10.1109/ICACNIS57039.2022.10055248.

[20] C. Channarong, C. Paosirikul, S. Maneeroj, and A. Takasu, "HybridBERT4Rec: A Hybrid (Content-Based Filtering and Collaborative Filtering) Recommender System Based on BERT," *IEEE Access*, vol. 10, pp. 56193–56206, 2022, doi: 10.1109/ACCESS.2022.3177610.

[21] A. Anindditya, M. A. Hasibuan, and E. Sutoyo, "Text Mining Approach Using TF-IDF and Naive Bayes for Classification of Exam Questions Based on Cognitive Level of Bloom's Taxonomy," *2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*, Nov. 2019, doi: 10.1109/iotais47347.2019.8980428.

[22] A. Yusmar, L. K. Wardhani, and H. B. Suseno, "RESTAURANT RECOMMENDER SYSTEM USING ITEM BASED COLLABORATIVE FILTERING AND ADJUSTED COSINE ALGORITHM SIMILARITY," *Jurnal Teknik Informatika*, vol. 14, no. 1, pp. 93–100, Sep. 2021, doi: 10.15408/jti.v14i1.21102.

[23] C. W. Kencana, E. B. Setiawan, and I. Kurniawan, "Hoax Detection System on Twitter using Feed-Forward and Back-Propagation Neural Networks Classification Method," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 4, no. 4, pp. 655–663, Aug. 2020, doi: 10.29207/resti.v4i4.2038.

[24] G. Srivastav and S. Kant, "Review on e-Learning Environment Development and context aware recommendation systems using Deep Learning," *2019 3rd International Conference on Recent Developments in Control, Automation & Power Engineering (RDCAPE)*, Oct. 2019, doi: 10.1109/rdcape47089.2019.8979066.

[25] B. Choe, T. Kang, and K. Jung, "Recommendation System with Hierarchical Recurrent Neural Network for Long-Term Time Series", *IEEE Access*, Vol. 9, pp. 72033–72039, 2021, doi: 10.1109/ACCESS.2021.3079922.

[26] 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*, vol. 6, no. 2, pp. 61–68, Sep. 2023, doi: 10.15408/aism.v6i2.26741.

[27] J. Xiang, Z. Qiu, Q. Hao, and H. Cao, "Multi-time scale wind speed prediction based on WT-bi-LSTM," *MATEC Web of Conferences*, vol. 309, p. 05011, Jan. 2020, doi: 10.1051/matecconf/202030905011.

[28] B. K. Yousafzai et al., "Student-Performulator: Student academic performance using hybrid deep neural network," *Sustainability*, vol. 13, no. 17, p. 9775, Aug. 2021, doi: 10.3390/su13179775.

[29] A. D. Safira and E. B. Setiawan, "Hoax Detection in Social Media using Bidirectional Long Short-Term Memory (Bi-LSTM) and 1 Dimensional-Convolutional Neural Network (1D-CNN) Methods," *11th International Conference on Information and Communication Technology (ICoICT)*, Aug. 2023, doi: 10.1109/icoict58202.2023.10262528.

[30] K. U. Wijaya, and E. B. Setiawan, "Hate Speech Detection Using Convolutional Neural Network and Gated Recurrent Unit with FastText Feature Expansion on Twitter," *JITEKI : Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 9.3 (2023): 619-631. 2024. doi: 10.26555/jiteki.v9i3.26532.

[31] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58080–58093, Jan. 2020, doi: 10.1109/access.2020.2982538.

[32] S. Moldovanu, C.-D. Obreja, K. C. Biswas, and L. Moraru, "Towards accurate diagnosis of skin lesions using feedforward back propagation neural networks," *Diagnostics*, vol. 11, no. 6, p. 936, May 2021, doi: 10.3390/diagnostics11060936.

[33] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos, Solitons & Fractals/Chaos, Solitons and Fractals*, vol. 139, p. 110045, Oct. 2020, doi: 10.1016/j.chaos.2020.110045.

[34] 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, 2020, doi: 10.14201/adcaij2020927990.

[35] S. Vani and T. V. M. Rao, "An Experimental Approach towards the Performance Assessment of Various Optimizers on Convolutional Neural Network," *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, doi: 10.1109/icoei.2019.8862686.

[36] Y. Li, C. Wei, and T. Ma, "Towards explaining the regularization effect of initial large learning rate in training neural networks," *Neural Information Processing Systems*, vol. 32, pp. 11669–11680, Jul. 2019, [Online].

[37] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "The effect of adaptive learning rate on the accuracy of neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, Jan. 2021, doi: 10.14569/ijacsa.2021.0120885.

[38] A. Luque, A. Carrasco, A. Martín, and A. De Las Heras, "The impact of class imbalance in classification performance metrics based on the binary confusion matrix," *Pattern Recognition*, vol. 91, pp. 216–231, Jul. 2019, doi: 10.1016/j.patcog.2019.02.023.