Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

JURNAL TEKNIK INFORMATIKA

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

Genetic Algorithm Optimization of Hybrid LSTM-AutoEncoder in Tourism Recommendation System

Bayu Surya Dharma Sanjaya[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: July 24, 2024
Revised: March 12, 2024
Issued: October 29, 2024

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

ABSTRACT

The tourism industry has rapid growth and has become one of the world's leading economic industries in recent years due to advances in information technology, such as the internet and social media. However, the overwhelming amount of information often makes it difficult for travelers to decide on their preferred travel destination. To address these issues, this research proposes a tourism recommendation system that combines Content-Based Filtering and Hybrid LSTM-AE, which is optimized using Genetic Algorithm (GA). There is no research that has developed a recommendation system using a combination of these methods and optimized using GA. So that this research can contribute to providing personalized recommendations and higher accuracy. The dataset consists of 9,504 ratings collected from the Ministry of Tourism and Creative Economy, Twitter, and web sources. The system was able to achieve a rating prediction accuracy of 96.82% by applying SMOTE to handle data imbalance and implementing a GA approach to the Hybrid LSTM-AE model. Accuracy has increased by 18.7% from the baseline model without using SMOTE and optimization. These results underscore that a strong integration between natural language processing and genetically optimized deep learning provides more accurate recommendations.

Keywords : *recommendation system; content-based filtering; auto encoder; LSTM; classification;*

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

1. INTRODUCTION

In recent years, the tourism industry has become one of the most rapidly growing sectors. Tourism has grown significantly almost all around the world, becoming the largest industry and significant contributor to the global economy [1]. Indonesia is a country that is officially recognized by the government as a country that makes tourism one of the main sectors, thus having a significant impact on the country's economy [2]. Along with the growth of information technology, particularly the internet and social media, travelers are increasingly using digital platforms to search for information, share experiences, and plan their trips. Social media has become a major platform for interaction between users, enabling information sharing and creating social relationships in digital form [3]. Platforms such as Twitter allow users to give opinions, opinions and recommendations through short messages or tweets [3], [4]. However, along with the abundance of information available, tourists often need help finding tourist attractions that match their preferences. To address this challenge, technology-based tourism recommendation systems are becoming increasingly important. These systems are designed to filter and present relevant information to users, helping them make more informed and personalized decisions regarding the tourism attractions they want to visit [5].

Some relevant research that has been done is the primary reference in this research. Based on research conducted by S. Bhaskaran and R. Marappan [6] shows that an improved vector space recommendation system model to address criticisms of the proposed e-learning recommendation system achieves better performance metrics in runtime of computation time, Mean Absolute Error (MAE), ranking score, precision, recall, and accuracy benchmarked to existing leading strategies. The model achieves MAE between 5.08% and 25.26% and accuracy between 80% and 93% for the regarded as a benchmark example. In another research, Permana and Wibowo [7] proposed a content-based movie recommendation system using the cosine similarity and the Term Frequency-Inverse Document Frequency (TF-IDF) with Gensim library to extract keywords. The evaluation results showed that the performance of the system, when user preferences were included, improved significantly with precision, recall, and F1-score reaching 0.3, 0.1, and 0.12 for the top 10 recommendations in genre-based scenarios, respectively. Further refinement of the system to provide more relevant and accurate recommendations is still needed in this research.

In research by C. Wang et al. [8], the proposed method is anomaly detection with an autoencoder neural network model that performs prediction and reconstruction on input data simultaneously. The performance of the process is verified through experiments on the Secure Water Treatment (SWaT) dataset, which shows that the proposed method provides improved performance with a recall of 88.5% and an F1-score of 87.0%. On the other hand, Shafqat and Byun [9] proposed a hierarchical model based on Long-Short Term Memory (LSTM) enriched with context. The model consists of two hierarchical levels, where each level has a deep LSTM network. In the first stage, the LSTM learns the user's traveling history and predicts the probability of the next location. In the second stage, a different LSTM framework learns context features embedded with probability and generates the most recommended places. The performance of the hierarchical LSTM approach shows improvement with an accuracy higher than the Gated Recurrent Unit (GRU) model and the Bidirectional LSTM. Hybridization of deep learning models can be done to get higher performance, even if the use of independent models is enough to work optimally.

Furthermore, based on research conducted by Fatimatuzzahra et al. [10], they developed an optimized neural network algorithm using Genetic Algorithms (GA) to analyze and predict local and international tourist arrivals. Optimized parameters include the number of neurons for the input layer, hidden layer, gradient, alpha, momentum, and another parameter. The Root Mean Square Error (RMSE) for the GA optimized neural network algorithm is 0.044. This research shows that GA can be used to define the hidden layer, corresponding features, momentum, and weight optimization of neural networks. GA used as an artificial neural network parameter optimization provides superior performance, so it can be applied to more complex deep learning models.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

Based on the literature review, this research proposes a tourism recommendation system that uses combination of Content-Based Filtering (CBF) with LSTM-AutoEncoder optimized using Genetic Algorithm. The combination of these methods aims to fill the gaps of related research that has been done before.

CBF allows the system to recommend items based on content information from the data and to deliver more tailored recommendations based on user preferences and previous interactions [1], [3-7]. The use of hybrid LSTM-AutoEncoder helps in mapping complex data and handling incomplete data problems, overcoming long-term or high-dimensional dependency problems, thus improving the system's ability to provide accurate and meaningful recommendations to users. Not only that, the use of GA to optimize classification models can produce more efficient and relevant models by filtering out irrelevant information and retaining important information.

To the best of our knowledge, no research has proposed a combination of LSTM-AutoEncoder optimized with GA in developing a recommendation system. The motivation of this research is to improve the accuracy of tourism prediction in accordance with user preferences by using a combination of these methods. This research contributes to the development of a tourism recommendation system that produces relatively low errors and excellent accuracy, so this model is hypothesized to provide accurate tourism recommendations to users.

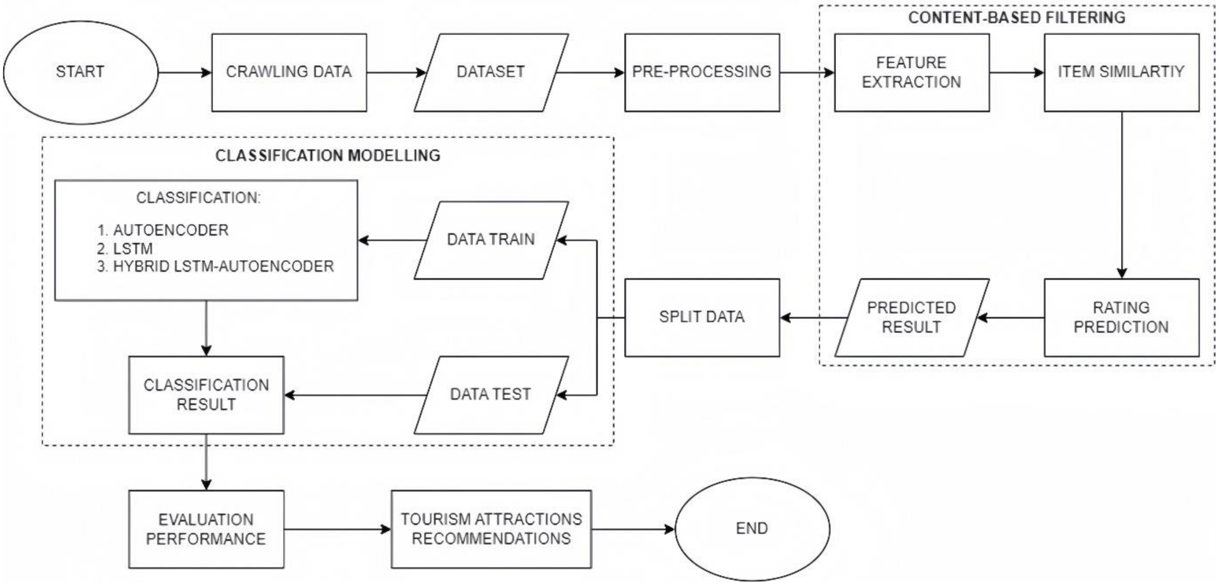

Figure 1. *Flowchart of proposed method*

2. METHODS

This research builds a tourism attractions recommendation system that uses the Content-Based Filtering method integrated with a hybrid deep learning algorithm, namely AutoEncoder with long short-term memory (LSTM) and is optimized using the genetic algorithm method. The overall system design in this research is shown in Figure 1, which illustrates the overall general design flow in this research.

2.1. Crawling Data

This research uses three different types of data, namely data from the Ministry of Tourism and Creative Economy, tweet data, and web data. Tourism data from the Ministry of Tourism and Creative Economy was obtained with permission and certain conditions. The tweet data was obtained through crawling using the Twitter API, which leads to tweets containing tourism attractions in the Bandung area, including opinions, reviews, and recommendations. In addition, data from websites such as TripAdvisor, Google Maps, Traveloka, and others have been crawled in addition to more verified data.

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

2.2. Preprocessing

At this stage, raw data processing is carried out, which comes from the results of crawling data from Twitter. Since the crawled tweet data is still in its raw form, where the data has not been processed or formatted, it often contains errors, duplications, and irrelevant information. Data processing aims to transform text data into a more structured and cleaner format through cleaning and transformation processes so that it becomes more organized, error-free, and ready for use. Steps in data processing include cleaning data from noise, case folding, stopword removal, stemming, and tokenization. After that, the polarity score of the tweet is calculated using TextBlob so that it becomes a rating form with a range of 1 to 5. TextBlob is a Python library that provides a simple API to dive into Natural Language Processing (NLP) tasks [11]. TexBlob is used to process textual data by identifying sentiment from text that can be categorized into positive, negative or neutral sentiment [11], [12]. Preprocessing ensures that the data fed into the recommendation system is of good quality. The following are the steps in preprocessing tweet data:

a. Data cleaning involves removing punctuation, numbers, emoticons, and URLs from tweet sentences.
b. Case Folding: involves changing all letters in the document to lowercase.
c. Stopword removal involves the removal of auxiliary words by removing less important words and keeping words that are considered important.
d. Stemming involves removing words that are bound to the base form.
e. Tokenization involves splitting words in a sentence for text analysis purposes.

2.3. Content-based filtering (CBF)

Content-based filtering (CBF) is an approach that relies on similarities between items based on users' previous preferences [1], [3-7]. This method is based on analyzing the features or attributes of each item, such as text, to look for similarities. By understanding the characteristics and content of each item, CBF can generate more suitable and personalized recommendations for users [4]. Each attraction has a profile that includes its unique characteristics, such as name, description, and other attributes. If user data is available, CBF can also be used to create user profiles based on preferences or history of visits to attractions. With this approach, the recommendation system can deliver more personalized and relevant recommendations for each user. In this research, CBF will utilize IndoBERT as a feature extraction of preprocessed tweet data. Furthermore, the feature extraction result vector will be used for item similarity calculation so that CBF can predict ratings that are 0 (not yet rated by the user).

2.3.1. Feature Extraction

In this research, IndoBERT is used as a tool to classify text data that has gone through a pre-processing stage. IndoBERT, as a variant of Bidirectional Encoder Representations from Transformers (BERT), has been tested using Indonesian datasets [9]. It is a BERT-based trained model on top of a large corpus of Indonesian language and has a similar structure to BERT, consisting of a stack of Transformer encoders that allows it to present a robust and precise representation of Indonesian texts.

Therefore, IndoBERT is strongly compatibility with BERT as both adopt the Transformer encoder stack in their structure [13]. IndoBERT's advantage in classifying Indonesian text data lies in its ability to unearth contextual nuances and deep semantic meanings. The self-attention mechanism integrated into the Transformer architecture gives IndoBERT an added advantage, allowing it to grasp well the complex language structures and linguistic relations in Indonesian texts. Moreover, IndoBERT continues to undergo refinement through a process of adaptation and fine-tuning on specialized classification tasks, which significantly improves its performance. IndoBERT can provide more robust, consistent and accurate classification results.

In this research, IndoBERT is used to extract features from pre-processed tweet text data. The input is a collection of tweet texts, then goes through a tokenization process that is fed into the IndoBERT model to produce vector representations as features extracted from the text. The output of this process is a numerical vector that can be used to calculate the similarity between items and predict ratings in a content-based recommendation system.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

2.3.2. Item Similarity

In this research, cosine similarity becomes an essential method in content-based filtering (CBF) to measure how similar the user profile is to the tourism attraction profile. The concept of cosine similarity is used to assess the similarity between items based on the attributes in the rating matrix [14]. To calculate the similarity value between 2 items, a set of users is required to rate the items [15]. Cosine similarity provides a robust measure of how close or similar the items in the recommendation system are by calculating the cosinus of the angular distance between two vectors in multidimensional space. The following is the formula for calculating cosine similarity:

$$Sim(a,b) = \frac{\sum_{n=1}^{m}(R_{n,a}-\hat{R}_a)\,(R_{n,b}-\hat{R}_b)}{\sum_{n=1}^{m}(R_{n,a}-\hat{R}_a)^2\,(R_{n,b}-\hat{R}_b)^2} \qquad (1)$$

$Sim(a,b)$ denotes the similarity between items a and b, where $\hat{R}_a$ and $\hat{R}_b$ represent the average probability values of items a and b, respectively. In addition, $R_{n,a}$ and $R_{n,b}$ are the probability values of items a and b in cluster n, while m is number of predefined clusters [14]. The result of the cosine similarity calculation helps CBF to predict the value of 0 rating, so that it can provide suitable tourism attractions to the user.

2.4. AutoEncoder (AE)

AutoEncoder (AE) is an unsupervised neural network model used to reassemble input data at the output layer [8], [16-19]. The latent space layer, which is the middlemost layer in the autoencoder structure, acts as a feature description of the input data that can be used for various purposes, such as dimensionality reduction, anomaly detection, or feature extraction [16]. Through this process, the autoencoder compresses the input data into a more miniature representation in the latent space layer, which enables the extraction of essential features and the removal of irrelevant information. The architecture of the AE can be seen in Figure 2.

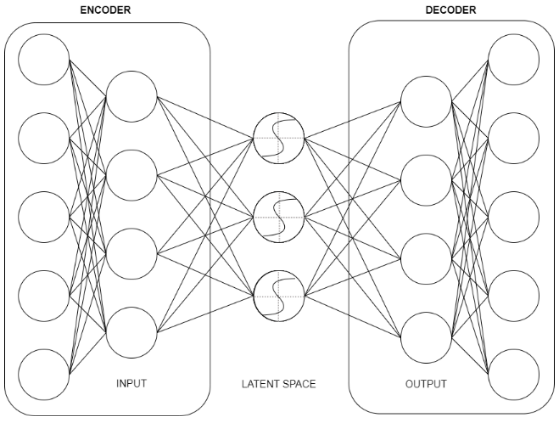

Figure 2. *Auto encoder neural network*

Similar to a multilayer perceptron, the autoencoder is a feed-forward neural network, but the difference lies in the primary function of autoencoder, which reassemble the input, while the purpose of a multilayer perceptron is to predict the target value with a specific input [18]. Based on Figure 2, AE consists of two main parts: encoder and decoder [8]. The input layer receives the raw data, while the encoder encodes the input into a more compact representation. Latent space is the middle layer that stores the essential features of the data in a lower dimension. The decoder then reassembles the data from the latent space back to its original form, producing an output like the original input.

Autoencoder variants include denoising AE that removes noise from the input data, marginalized AE that accounts for data imprecision, sparse AE that has a limited number of active units in the representation, contractive AE that maintains a stable representation, and variational AE that uses a probability approach to generate a continuous representation of the data [16].

2.5. Dataset Splitting

Long Short-Term Memory (LSTM) a variant of recurrent neural network (RNN) introduced to solve the problem of long-term dependency [20-23]. LSTM has the strength of overcoming the missing gradient problem when handling data sets containing long sequential data [9, 22, 24]. The LSTM structure consists of a series of memory cells and three gates, namely input gate (i), output gate (O), and forget gate (f). Each of these gates serves to regulate the flow of information, with the input gate controlling the incoming data, the output gate determining the outgoing information, and the

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

forget gate managing which information should be forgotten or stored in memory cells. Figure 3 shows the architecture of the LSTM model [23].

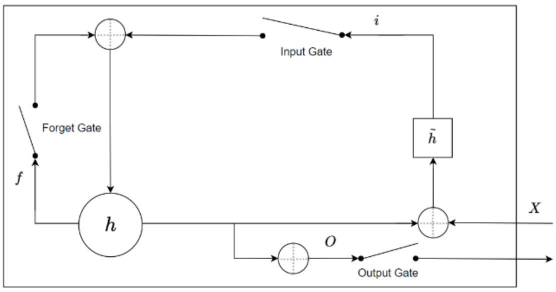

Figure 3. *Long short-term memory*

Each gate uses a sigmoid layer and a dot multiplication operation, the input value is passed to the input gate to update and remove old information, and the sigmoid layer outputs (0 and 1) to determine how much information is passed. This architecture allows the LSTM to effectively manage information and overcome the problem of long-term dependencies in sequence data. Here is the formula of the LSTM.

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + B_i) \qquad (2)$$

What new information is stored in the memory cell is decided by the input gate. The value of i_t is calculated by applying a sigmoid function to the sum of the weight multiplication with the concatenation between the previous output and the current input, plus the bias B_i.

$$f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + B_f\right) \qquad (3)$$

What information is discarded from the memory cell is decided by the forget gate. The value of f_t is obtained by applying a sigmoid function to the summation result of the weight multiplication with the concatenation between the previous output and the current input, plus the bias B_f.

$$\widetilde{h_t} = \tanh(w_c \cdot [h_{t-1}, x_t] + B_c) \qquad (4)$$

The candidate memory cell $\widetilde{h_t}$ represents the new value to be added to the memory cell. This value is calculated by applying the activation function tanh to the sum of the product of the weights with the concatenation between the previous output and the current input, plus the bias B_c.

$$h_t = f_t \cdot h_{t-1} + i_t \cdot \widetilde{h_t} \qquad (5)$$

The new memory cell h_t is the result of the combination of old and new information. This value is calculated by multiplying the value of f_t by the previous memory cell and adding the result of multiplying the value by the candidate memory cell.

$$O_t = \sigma(w_o \cdot [h_{t-1}, x_t] + B_o) \qquad (6)$$

The output gate sets the part of the cell's memory that will be output. The value of O_t is obtained by applying a sigmoid function to the sum of the product of the weights with the concatenation between the previous output and the current input, plus the bias B_o.

$$h_t = O_t \cdot \tanh(h_t) \qquad (7)$$

The final output h_t is the result obtained by multiplying the value of the output gate O_t with the activation function tanh of the memory cell h_t. This step ensures that only relevant and filtered information is transferred to the next stage in the network.

2.6. Hybrid LSTM-AutoEncoder

Hybrid Deep Learning combines two or more different deep learning techniques to create models that are more complex and effective in handling various types of problems [24]. This method combines the strengths of each technique used to improve the performance and flexibility of the model. In this research, we will combine two different types of models, namely AutoEncoder and LSTM. LSTM-AutoEncoder is a type of AE that uses an LSTM network as an encoder and decoder [25]. LSTM's ability to understand patterns in sequential data makes it practical for tasks such as classification that work sequentially. To overcome the complexity of data in classification that is influenced by various factors, a hybrid LSTM-AutoEncoder (AE) is used. Figure 4 shows the architecture of the hybrid LSTM-AutoEncoder.

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

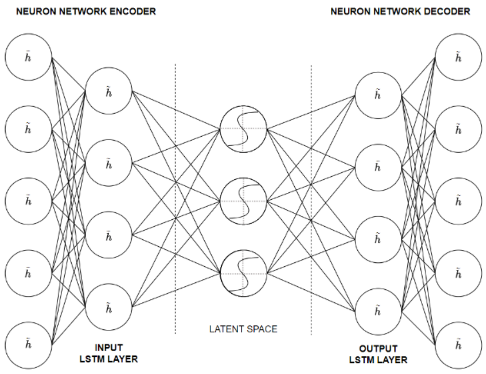

Figure 4. *Hybrid LSTM-autoencoder*

Based on Figure 4, the encoder and decoder comprise an LSTM network, which enables the model to capture and reconstruct patterns in sequential data effectively. The process begins with the LSTM encoder converting the sequential input into a latent space representation, or code, that summarizes the essential information of the input data. This latent space representation is then passed to the LSTM decoder, which aims to reconstruct the original input from this representation. The AE reduces the dimensionality of the data by extracting key characteristics, so the input data for the LSTM consists of these characteristics and relevant historical data [26]. In this model, the LSTM processes more straightforward input data while still considering pertinent various influences. LSTM-AutoEncoder can perform classification more efficiently, taking into account the impact of multiple factors without increasing the complexity of calculations [26]. Thus, LSTM-AutoEncoder combines the power of AE in feature extraction and the ability of LSTM to process long sequence data, resulting in more accurate and efficient classification.

2.7. Genetic algorithm (GA)

Genetic algorithm (GA) is an algorithm inspired by Darwin's theory of evolution, where the process begins with the random generation of a population of chromosomes representing potential solutions. Through iteration, the GA selects the most suitable chromosomes for the next generation using crossover and mutation operations, similar to the concept of natural selection that leads to an increase in the suitability of solutions in the population. In recommendation problem solving, GA is widely used to search for the optimal solution, adopting the principle of continuity of the best solution [27].

GA apply the principles of biological evolution by representing potential solutions as individuals in a population. GA operates through a series of iterations, or generations, where individuals undergo selection, reproduction (crossover) and mutation to produce the next generation. Each individual in the population presents a possible solution to the problem at hand, and the effectiveness of the solution is assessed by a fitness function [28]. Here is the GA process flow.

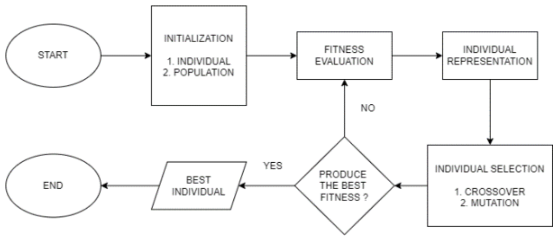

Figure 5. *Genetic algorithm flow*

Each individual in the population is insured for survival in the next generation. To create a new individual by combining genetic information from its parents, a crossover step is performed. A frequently used parental selection process is using the roulette wheel technique. Meanwhile, the mutation process occurs by replacing less suitable individuals with new individuals, where the replacement rate is influenced by the mutation rate parameter [29].

In this research, GA is used to optimize a hybrid LSTM-AutoEncoder model for rating classification. The process begins with the generation of an initial population of potential solutions, representing various model parameters. Through iteration, the GA performs selection, crossover, and mutation to generate a new generation of improved solutions, using a fitness function to assess the effectiveness of each solution. Thus, GA helps to find the optimal configuration of parameters, which results in significant improvements in the accuracy and performance of the proposed recommendation system.

2.8. Evaluation Performance

The evaluation of experimental results is an integral part of the overall process of measuring the efficiency of the model. One of the traditional evaluation methods often used is through a confusion matrix. Accuracy is often used as the primary metric to measure experimental results. Accuracy is obtained by

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

comparing the actual classification results with the classification predictions made by the model, which are then calculated based on a formula involving the elements in the confusion matrix [30]. In this way, the accuracy provides an overview of how good the model is at identifying the correct classification in the entire data set. The accuracy calculation is shown in formula (8), and confusion matrix can be seen in Table 1.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Table 1. *Confusion matrix*

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

3. RESULTS AND DISCUSSION

This research consists of three main stages: data preparation, recommendation system development, and classification. Data preparation includes data crawling and pre-processing. In the recommendation system stage, rating prediction is performed using CBF with the help of IndoBERT feature extraction. The classification stage focuses on labeling each rating and uses an optimized LSTM-AE hybrid with GA as classification.

3.1. Data Preparation Result

This research conducted three processes in retrieving data from different sources. The first dataset was taken from the Ministry of Tourism and Creative Economy with applicable permissions, containing 124 tourism attractions in Bandung complete with descriptions, categories, ratings, and other information. This tourism data became the reference for the following crawling process. Table 2 shows an example of the first dataset.

Table 1. *Ministry of Tourism and Creative Economy Dataset*

Place Id	Place Name	…	Category	Rating
211	Gunung Tangkuban Perahu	…	Cagar Alam	4.5
212	Jalan Braga	…	Budaya	4.7
…	…	…	…	…
333	Kota Mini	…	Taman Hiburan	4.4
334	Chingu Café Little Seoul	…	Taman Hiburan	4.5

Furthermore, the second dataset was obtained by using TwitterAPI to collect tourism-related tweets. Crawling was performed using tourism keywords as per the first dataset. Twitter crawling produced 57,365 results, which were then cleaned so that the text could be labeled using TextBlob based on the sentiment of the tweet text. Table 3 shows an example of the second dataset.

Table 2. *Twitter dataset*

Username	Full Text	Label
DolanYok	Melihat Kawah Indah Di Objek Wisata Tangkuban Perahu	5
kompasiana	Eko-Wisata Tangkuban Perahu: Keindahan Alam dan Pelestarian Lingkungan di Satu Tempat	5
…	…	…
Feilatan1304	view of new chingu cafe nice korean cafe in bandung jalan sawunggaling 10	4
RedDoorzID	Kamu penggemar K-Pop dan K-Drama? Gak usah jauh-jauh ke Korea Di Chingu Cafe Bandung, kamu bisa makan beragam hidangan khas Korea, foto-foto di banyak area yang terinspirasi oleh Dongdaemeun, bahkan memakai hanbok	3

The third dataset is collected from several trusted web sources that provide tourism information, reviews and ratings as additional data for verified data, such as TripAdvisor, Google Maps, Traveloka, and others. Furthermore, the three datasets are then combined to form a matrix between users and tourism with a size of 95 x 100. Table 4 shows the final dataset that is ready to be utilized for the recommendation system.

Table 4. *Final Dataset*

Username	Tourism Attractions				
	Gunung Tangkuban Perahu	**Jalan Braga**	**…**	**Kota Mini**	**Chingu Cafe Little Seoul**
Trip Advisor	3.50	4.00	…	0.00	3.00
Google Maps	4.50	4.80	…	4.30	4.40
…	…	…	…	…	…
bandung 911	0.00	0.00	…	3.00	0.00
GEMAWA HYUH	0.00	0.00	…	0.00	0.00

3.2. Recommendation System Result

Content-based filtering (CBF) using IndoBERT in tourism recommendation systems involves several critical stages. The first stage is the extraction of text features such as place names, descriptions, and categories from the data of the Ministry of Tourism and Creative

Economy. IndoBERT, a transformation model that has been trained in Indonesian, is used to process this text and generate a vector representation that considers the context. This process converts the text information from the columns into embeddings that can be analyzed further. The embedding from IndoBERT is then used to calculate the content similarity between tourism attractions using cosine similarity so that the system can recommend similar places based on the existing content. Finally, the system provides rating predictions for tourism attractions that still need to have a rating (0) by considering similarity and previous user preferences. Table 5 shows the rating prediction results.

Table 5. *Rating prediction result*

Username	**Tourism Attractions**				
	Gunung Tangkuban Perahu	**Jalan Braga**	**…**	**Kota Mini**	**Chingu Cafe Little Seoul**
Trip Advisor	3.50	4.00	…	2.54	3.00
Google Maps	4.50	4.80	…	4.30	4.40
…	…	…	…	…	…
bandung 911	0.48	0.49	…	3.00	0.48
GEMAWA HYUH	0.90	0.91	…	0.90	0.90

3.3. Classification Result

Classification begins by labelling the predicted ratings into binary form, where a value of 1 indicates that the prediction is recommended and a value of 0 is not recommended. The binary labelled data is then used as data in the construction of AutoEncoder (AE), Long Short-Term Memory (LSTM), and Hybrid LSTM-AE models. The classification stage process is carried out in several experiments, namely scenario 1 for the baseline model, scenario 2 for the baseline with the application of SMOTE, and scenario 3 baseline with the application of SMOTE and optimization. The three scenarios are carried out to determine the accuracy comparison of each model, and to increase accuracy until maximum results are obtained.

3.3.1. Scenario 1

The baseline model was built using unit parameters of 32 for unit 1 and 2, dropout of 0.3, activation with sigmoid, batch size 64, and model trained for 50 epochs. The first scenario was conducted to test baseline model with variations in ratio of test data and training data, namely 10:90, 20:80, 30:70, and 40:60. The results of first scenario can be seen in Table 6.

Table 6. *First scenario result*

Model	**Data Split Ratio**	**Accuracy (%)**
AutoEncoder	10:90	80.21
	20:80	**80.87**
	30:70	80.03
	40:60	79.27
LSTM	**10:90**	**80.48**
	20:80	80.37
	30:70	79.88
	40:60	78.92
Hybrid LSTM-AutoEncoder	10:90	80.87
	20:80	**81.56**
	30:70	80.10
	40:60	80.03

Experimental results of the first scenario show that each model achieves good accuracy, with values between 78% and 81% for various ratios of test and train data. The highest accuracy of the AutoEncoder model is 80.87% at a test size of 20%, the LSTM model is 80.48% at a test size of 10%, and the Hybrid LSTM-AutoEncoder model gets the highest accuracy at a test size of 20%, with an accuracy of 81.56%. This accuracy indicates that the developed models are able to train on the rating data effectively.

3.3.2. Scenario 2

The second scenario involved model testing experiments with re-sampling using SMOTE (Synthetic Minority Over-sampling Technique) to address unbalanced data. SMOTE randomly selects a line segment that connects the sample with its nearest neighbor to sample data from the minority class. The results of the second scenario can be seen in Table 7.

Table 7. *Second scenario result*

Model	**Data Split Ratio**	**Accuracy (%)**
AutoEncoder	**10:90**	**83.39 (+3.96)**
	20:80	83.07 (+2.72)
	30:70	82.76 (+3.41)
	40:60	82.55 (+4.14)
LSTM	**10:90**	**84.18 (+4.60)**
	20:80	84.12 (+4.67)
	30:70	83.23 (+4.19)
	40:60	83.18 (+5.40)
Hybrid LSTM-AutoEncoder	10:90	85.20 (+5.35)
	20:80	**85.66 (+5.03)**
	30:70	84.94 (+6.04)
	40:60	84.22 (+5.24)

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

The use of SMOTE shows increased accuracy, indicating that training the model with balanced data is better than the baseline. The AutoEncoder model produced the highest accuracy of 83.39%, an increase of 3.96%, and the LSTM model produced an accuracy of 84.14%, an increase of 4.60%. At the same time, the Hybrid LSTM-AutoEncoder model produces the highest accuracy among to the two models, with an accuracy of 85.66%, up to 5.03% from the baseline. The model with the highest accuracy of each model is tested again in the following scenario.

3.3.3. Scenario 3

The third scenario involved model testing experiments using Adam, SGD, RMSprop, Adagrad, and Genetic Algorithm optimizations. The model used in the third scenario is the model with the highest accuracy from the second scenario. This scenario aims to compare the accuracy of models optimized with genetic algorithm with models optimized with commonly used optimizations. The genetic algorithm process starts with the initialization of a population consisting of various configurations of model parameters. Each individual in the population represents a different set of parameters. The following are the Genetic Algorithm parameters used in this research.

Table 8. *Genetic algorithm parameters*

Parameter	Value	Description
FitnessMax	1.0	A fitness function used to maximize the objective score (fitness) of individuals in the population.
initCycle	2	Initialize individuals with two attributes using the cyclic method, where one attribute is an integer, and the other is a float.
initRepeat	5	Initialize a population of 5 individuals using a predefined method.
crossxUniform	0.5	Apply the uniform crossover method with a probability of 0.5 during the reproduction process for genetic combination.
mutGaussian	1, 0.2	Apply Gaussian mutation with a standard deviation of 1, with a probability of 0.2 for each mutation that occurs.
setTournament	3	Selecting individuals using the tournament selection method with a tournament size of 3 to determine the best.

The genetic algorithm used a population size of 5 individuals, stores the best individuals in the Hall of Fame, with a crossover probability of 0.5, mutation probability of 0.2, and runs for 10 generations. The algorithm applies selection, crossover, and mutation to generate new generations to maximize model accuracy. Selection selects the best individuals based on accuracy performance, crossover combines pairs of individuals to produce new offspring, and mutation randomly changes some parameters to maintain diversity in the population. This process is repeated until convergence, or a specified number of generations is reached. Table 9 shows the genetic algorithm selection results with the best fitness compared to commonly used optimization.

Table 9. *Third scenario result*

Model	Optimization	Accuracy (%)
AutoEncoder	Adam	85.68 (+5.96)
	SGD	81.71 (+1.04)
	RMSprop	84.56 (+4.58)
	Adagrad	84.39 (+4.36)
	Genetic Algorithm	**90.47 (+11.8)**
LSTM	Adam	88.09 (+9.45)
	SGD	82.78 (+2.86)
	RMSprop	85.00 (+5.61)
	Adagrad	86.35 (+7.29)
	Genetic Algorithm	**95.23 (+18.3)**
Hybrid LSTM-AutoEncoder	Adam	88.69 (+8.78)
	SGD	82.63 (+1.31)
	RMSprop	88.88 (+8.97)
	Adagrad	85.98 (+5.42)
	Genetic Algorithm	**96.82 (+18.7)**

Table 10. *Genetic algorithm best individual of each model*

Model	Best Individual		Best Fitness (%)
	Parameters	**Values**	
AutoEncoder	Unit 1	128	90.47 (+11.8)
	Dropout 1	0.6515819256	
	Unit 2	64	
	Dropout 2	0.9469636892	
	Activation	Sigmoid	
LSTM	Unit 1	128	95.23 (+18.3)
	Dropout 1	0.2513474172	
	Unit 2	16	
	Dropout 2	0.1175876053	
	Activation	Sigmoid	
Hybrid LSTM-AutoEncoder	Unit 1	128	96.82 (+18.7)
	Dropout 1	0.2007236117	
	Unit 2	32	
	Dropout 2	0.2846718589	
	Activation	Sigmoid	

Based on Table 9, each model optimized with Genetic Algorithm produces the highest accuracy compared to models with other optimizations. Table 10 shows the individual best of each model optimized with GA. The AutoEncoder model with the best individual parameters of unit 1 of 128, unit 2 of 64,

Jurnal Teknik Informatika Vol. 17 No. 2, Oktober 2024 (98-111)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

dropout 1 of 0.6515, and dropout of 0.9469 produces the highest fitness of 90.47%. This AE model requires a higher dropout to achieve maximum accuracy. The LSTM model found the best individual with unit 1 of 128, dropout 1 of 0.2513, unit 2 of 16, and dropout 2 of 0.1175. This individual produced a fitness of 95.23%. Just like the LSTM model, the Hybrid LSTM-AutoEncoder does not require many dropout values to deliver maximum accuracy. This model, with individual units 1 of 128, units 2 of 32, dropouts 1 of 0.2007, and dropouts 2 of 0.2846, produces the highest fitness at 96.82%. This experiment demonstrates the effectiveness of Genetic Algorithm as a neural network model optimization in classification tasks compared to commonly used optimization algorithms.

3.4. Discussion

In this research, the data preparation process was conducted in three main stages to obtain a comprehensive dataset. The first stage involved collecting data from the Ministry of Tourism and Creative Economy, which contains complete information on 124 tourism attractions in Bandung, including descriptions, categories, and ratings. The second dataset was obtained through TwitterAPI by crawling using the keyword tourism, resulting in 57,365 results that were then sentiment-scored using TextBlob. Finally, data from various trusted web sources, such as TripAdvisor and Google Maps, were combined to form a matrix between users and tourism attractions with a size of 95 x 100. The integration of these three datasets enabled the development of a more accurate and comprehensive recommendation system.

The implementation of content-based filtering (CBF) with IndoBERT is successful in providing more relevant tourism attractions based on content similarity. The use of cosine similarity measures also strengthens the system's ability to suggest places that match user preferences, as evidenced by the prediction of unrated columns using cosine similarity vector values from IndoBERT. The results of this system provide rating predictions for tourism attractions that still need to have a rating, considering previous user preferences.

In the classification experiments conducted, the three models, AutoEncoder (AE), Long Short-Term Memory (LSTM), and Hybrid LSTM-AutoEncoder, were evaluated in three different scenarios. In the first scenario without resampling (SMOTE), all three models achieved accuracies between 78% and 81%, with the Hybrid LSTM-AutoEncoder recording the highest accuracy of 81.56%. Then, in the second scenario, with the application of SMOTE to handle data imbalance, there was a significant improvement in accuracy for all models, with the Hybrid LSTM-AutoEncoder improving from 81.56% to 85.66%. The third scenario involved optimization using various algorithms, where the model optimized with Genetic Algorithm (GA) achieved the highest accuracy, reaching 96.82% for Hybrid LSTM-AutoEncoder. These results show that the use of SMOTE and GA is efficacious in improving the performance of classification models to achieve maximum accuracy.

In this research, a statistical significance test was conducted to validate the change in accuracy after all scenarios. The parameter for this test is the P-Value, which indicates the likelihood of no significant change. Statistically, a P-Value < 0.05 signifies significance. Whereas at the 95% confidence level, a Z-Value > 1.96 indicates that the difference is statistically significant [24]. Table 11 shows the results of the statistical significance test, and Figure 6 shows the graph of the increase in accuracy of each scenario.

Table 11. *Statistical significance test result*

Model	Parameters	Scenarios		
		S1→S2	**S2→S3**	**S1→S3**
AutoEncoder	Z-Value	12.04	149.62	42.77
	P-Value	0.0	0.0	0.0
	Significant?	**True**	**True**	**True**
LSTM	Z-Value	22.93	110.52	77.14
	P-Value	0.0	0.0	0.0
	Significant?	**True**	**True**	**True**
Hybrid LSTM-AutoEncoder	Z-Value	18.65	91.70	74.92
	P-Value	0.0	0.0	0.0
	Significant?	**True**	**True**	**True**

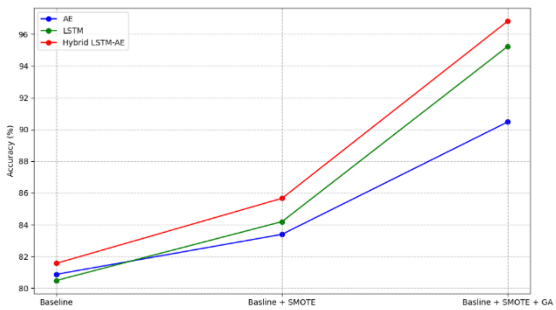

Figure 6. *Graph of increase in accuracy*

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

CONCLUSION

Based on the results of this research, an approach that combines Content-Based Filtering with feature extraction using IndoBERT for tourism recommendation systems, as well as the use of Hybrid LSTM-AE optimized with Genetic Algorithm (GA) for rating classification, yields high accuracy. The dataset used consists of 9,504 ratings from 96 users and 99 tourism attractions. The recommendation system implemented with the use of SMOTE and GA optimization achieves an accuracy of up to 96.82% on the Hybrid LSTM-AE model, an increase of 18.7% from the base model that does not use both. The implications of this research suggest that the integration of optimized natural language processing and deep learning techniques can significantly improve prediction accuracy in tourism recommendation systems. This research can thus have a wider impact and contribute to the tourism industry with the potential to improve user experience through more accurate and personalized recommendations, which in turn can increase user satisfaction. Suggestions for future research are to explore the use of other methods or integration with different approaches to enhance the diversity and accuracy of recommendations, as well as to test this model in a broader dataset for further validation.

REFERENCES

[1] W. Xia, "Digital transformation of tourism industry and smart Tourism recommendation algorithm based on 5G background," *Journal of Mobile Information Systems*, vol. 2022, pp. 1–13, Sep. 2022, doi: 10.1155/2022/4021706.

[2] D. M. Lemy, F. Teguh, and A. Pramezwary, "Tourism development in Indonesia," *in Bridging tourism theory and practice*, 2019, pp. 91–108. doi: 10.1108/s2042-144320190000011009.

[3] N. R. Vajjhala, S. Rakshit, M. Oshogbunu, and S. Salisu, "Novel User Preference Recommender System based on Twitter Profile analysis," *in Advances in intelligent systems and computing*, 2020, pp. 85–93. doi: 10.1007/978-981-15-7394-1_7.

[4] 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.

[5] L. Quijano-Sánchez, I. Cantador, M. E. Cortés-Cediel, and O. Gil, "Recommender systems for smart cities," *Information Systems*, vol. 92, p. 101545, Sep. 2020, doi: 10.1016/j.is.2020.101545.

[6] 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*, vol. 15, no. 3, pp. 1583–1595, Feb. 2023, doi: 10.1007/s41870-023-01165-2.

[7] A. H. J. P. J. Permana and A. T. Wibowo, "Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity," *International Journal on Information and Communication Technology (IJoICT)*, vol. 9, no. 2, pp. 1-14, 2023, https://doi.org/10.21108/ijoict.v9i2.747.

[8] C. Wang, B. Wang, H. Liu, and H. Qu, "Anomaly detection for industrial control system based on autoencoder neural network," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–10, Aug. 2020, doi: 10.1155/2020/8897926.

[9] W. Shafqat and Y.-C. Byun, "A Context-Aware Location Recommendation System for tourists using hierarchical LSTM model," *Sustainability*, vol. 12, no. 10, p. 4107, May 2020, doi: 10.3390/su12104107.

[10] F. Fatimatuzzahra, R. Hammad, A. Z. Amrullah, and P. Irfan, "Optimasi neural network dengan menggunakan algoritma genetika untuk prediksi jumlah kunjungan wisatawan," *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, vol. 3, no. 4, pp. 227–235, Feb. 2022, doi: 10.35746/jtim.v3i4.190.

[11] I. G. S. M. Diyasa, N. M. I. M. Mandenni, M. I. Fachrurrozi, S. I. Pradika, K. R. N. Manab, and N. R. Sasmita, "Twitter Sentiment Analysis as an evaluation and service base on Python Textblob," *IOP Conference Series. Materials Science and Engineering*, vol. 1125, no. 1, p. 012034, May 2021, doi: 10.1088/1757-899x/1125/1/012034.

[12] R. Bose, P. S. Aithal, and S. Roy, "Sentiment Analysis on the Basis of Tweeter Comments of Application of Drugs by Customary Language Toolkit and TextBlob Opinions of Distinct Countries," *International Journal of Emerging Trends in Engineering Research*. 8. 3684-3696. 2020. 10.30534/ijeter/2020/129872020.

[13] L. Geni, E. Yulianti, and D. I. Sensuse, "Sentiment Analysis of Tweets Before the 2024 Elections in Indonesia Using Bert Language Models," *JITEKI: Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, 9(3), 746-757. 2023. doi:http://dx.doi.org/10.26555/jiteki.v9i3.26490.

[14] Q. R. Arvianti, Z. K. A. Baizal, and D. Tarwidi, "Tourism Recommender System Using Item-Based Hybrid Clustering Method (Case Study: Bandung Raya Region)," *Journal of Data Science and Its Applications (Online)*, vol. 2, no. 2, pp. 95–101, Nov. 2019, doi: 10.34818/jdsa.2019.2.35.

[15] 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.

[16] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep Learning Based Recommender System: A Survey and New Perspectives". *ACM Comput. Surv*. 52, 1, 2019. https://doi.org/10.1145/3285029.

[17] K. R, P. Kumar, and B. Bhasker, "DNNRec: A novel deep learning based hybrid recommender system," *Expert Systems With Applications*, vol. 144, p. 113054, Apr. 2020, doi: 10.1016/j.eswa.2019.113054.

[18] H. Tahmasebi, R. Ravanmehr, and R. Mohamadrezaei, "Social movie recommender system based on deep autoencoder network using Twitter data," *Neural Computing & Applications*, vol. 33, no. 5, pp. 1607–1623, Jun. 2020, doi: 10.1007/s00521-020-05085-1.

[19] Y. Pan, F. He, and H. Yu, "Learning social representations with deep autoencoder for recommender system," *World Wide Web*, vol. 23, no. 4, pp. 2259–2279, Mar. 2020, doi: 10.1007/s11280-020-00793-z.

[20] N. Ben-Lhachemi, E. H. Nfaoui, and J. Boumhidi, "Hashtag Recommender System Based on LSTM Neural Reccurent Network," *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, Oct. 2019, doi: 10.1109/icds47004.2019.8942380.

[21] S. Bansal and N. Baliyan, "Remembering past and predicting future: a hybrid recurrent neural network based recommender system," *Journal of Ambient Intelligence & Humanized Computing/Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 12, pp. 16025–16036, Sep. 2022, doi: 10.1007/s12652-022-04375-x.

[22] B. A. Putri and E. B. Setiawan, "Topic Classification Using the Long Short-Term Memory (LSTM) Method with FastText Feature Expansion on Twitter," *In Proceedings of the International Conference on Data Science and Applications*, Aug. 2023, doi: 10.1109/icodsa58501.2023.10277033.

[23] 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, Jan. 2021, doi: 10.1109/access.2021.3079922.

[24] 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.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v17i2.39760

[25] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management," *International Journal of Information Management*, vol. 57, p. 102282, Apr. 2021, doi: 10.1016/j.ijinfomgt.2020.102282.

[26] W. Wei, H. Wu, and H. Ma, "An AutoEncoder and LSTM-Based Traffic Flow Prediction method," *Sensors*, vol. 19, no. 13, p. 2946, Jul. 2019, doi: 10.3390/s19132946.

[27] S. Kulkarni and S. F. Rodd, "Context Aware Recommendation Systems: A review of the state of the art techniques," *Computer Science Review*, vol. 37, p. 100255, Aug. 2020, doi: 10.1016/j.cosrev.2020.100255.

[28] B. S. Neysiani, N. Soltani, R. Mofidi, and M. H. Nadimi-Shahraki, "Improve Performance of Association Rule-Based Collaborative Filtering Recommendation Systems using Genetic Algorithm," *International Journal of Information Technology and Computer Science*, vol. 11, no. 2, pp. 48–55, Feb. 2019, doi: 10.5815/ijitcs.2019.02.06.

[29] A. Rafdi, H. Mawengkang, and S. Efendi, "Sentiment Analysis Using Naive Bayes Algorithm with Feature Selection Particle Swarm Optimization (PSO) and Genetic Algorithm," *International Journal of Advances in Data and Information Systems*, vol. 2, no. 2, Nov. 2021, doi: 10.25008/ijadis.v2i2.1224.

[30] W. Dang et al., "Increasing Text Filtering Accuracy with Improved LSTM," *Computing and Informatics*, vol. 42, no. 6, pp. 1491–1517, Jan. 2023, doi: 10.31577/cai_2023_6_1491.