Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods

Arliyanna Nilla, Erwin Budi Setiawan
Telkom University, Jl. Terusan Buah Batu, Bandung 40257, Indonesia

ARTICLE INFO

Article history:

Received December 30, 2023
Revised January 27, 2024
Published February 12, 2024

Keywords:

Recommender System;
Twitter;
Content Based Filtering;
Word Embedding;
RoBERTa;
TFIDF;
Classification;
Convolutional Neural Network

ABSTRACT

Managing large amounts of data is a challenge faced by users, so a recommendation system is needed as an information filter to provide relevant item suggestions. Twitter is often used to find information about movie reviews that can be used a basis for developing recommendation systems. This research contributes to applying content-based filtering in the context of Convolutional Neural Network (CNN). To the best of the researcher's knowledge, there has been no research addressing this combination of method and classification. The main focus is to evaluate the development of a recommendation system by integrating and comparing similarity identification methods using the RoBERTa and TF-IDF approaches. In this research, Roberta and TF-IDF as vectorizer and classification methods are applied to form a model that can recognize patterns in data and produce accurate predictions based on its features. The total data used is 854 movies and 34086 film reviews from 44 Twitter accounts. The SMOTE method was applied as a technique to overcome data imbalance. The research was conducted three times with increasing accuracy results. The first experiment TF-IDF as baseline, SMOTE on CNN classification. The second experiment, applying baseline, SMOTE, embedding on CNN classification. The third experiment applied baseline, SMOTE, embedding, and optimizer to CNN classification. The experimental results show that TF-IDF as baseline, SMOTE, embedding and SGD optimizer with the best learning rate on CNN classification can provide optimal results with an accuracy rate of 86.41%. Thus, the system can provide relevant movie recommendations with good prediction accuracy and performance.

Corresponding Author:

Erwin Budi Setiawan, Telkom Univeristy, Jl. Terusan Buah Batu, Bandung 40257, Indonesia
Email: erwinbudisetiawan@telkomuniversity.ac.id

1. INTRODUCTION

Internet data production continues to increase with the rapid development of internet technology. According to a Statista.com report, internet data usage currently stands at 79 zettabytes, with predictions reaching 181 zettabytes by 2025 [1]. This growth leads to the problem of information overload for users, especially in choosing a movie amidst the many options. The movie industry continues to produce new works yearly; for example, in 2019, 835 movies were released in the United States and Canada, an increase of 70 movies from the previous year [2]. The many movies produced make it confusing for users to choose the movie they want. Therefore, a recommendation system is formed that can be used by users to utilize their previous preferences in selecting and recommending items efficiently [3].

Recommender systems have become very popular in the entertainment industry, especially in the movie industry, and have been implemented in various platforms such as Netflix [4]. As one of the significant platforms for movie streaming, Netflix offers a wide array of original series and movies [5]. Disney+ Hotstar, a popular streaming app in addition to Netflix, was developed by the Walt Disney Company with a focus on providing family-safe digital streaming services, in keeping with Disney's reputation as a company that places importance on family values. Therefore, the content on this app is not R-rated [6]. Beyond streaming platforms, social media facilitates users to write messages, news, or reviews that others can be sent and read globally. For example, Twitter allows users to search for information related to their interests, knowledge, news, recommendations, and latest movie reviews. Recommendation systems are essential in helping users choose the movie they want to see [7].

Research [8] aims to find similarities between movies that have been watched by utilizing synopsis and title data, focusing on research on content-based movie recommendation systems. This method involves title, genre, and synopsis features, and applies cosine similarity and TF-IDF weighting. Evaluation using the MAP@K method on three users resulted in a system accuracy of 0.823254 for single queries and 0.7500556 for multiple seeds queries [8]. In the following study, research [9] conducted a performance comparison of various word embedding schemes and deep learning architectures, such as CNN, BERT, and Roberta. This study applied two sentiment analysis approaches. The first used CNN with Glove and Word2Vec word embedding. They were second, using trained CNN models such as BERT and Roberta. The results showed that the CNN model applying Glove and BERT achieved the highest test accuracy of 0.967 [9]. Other research, namely Research [10], focuses on predicting IMDb movie ratings using RoBERTa Embeddings and Neural Networks. This research aims to develop a movie rating prediction model using Roberta Embeddings and Neural Networks. In addition, this study compares model performance and explores the potential application of a semantic search engine with the model. The results show that the Neural Network model using RoBERTa embedding can predict movie ratings with an accuracy of about 85%. Furthermore, another study utilizes Convolutional Neural Network (CNN) to evaluate sentiment on the Twitter social media platform, achieving an accuracy rate of 88.21% [11]. Other knowledge about Convolutional Neural Network (CNN), in the study, the performance of CNN classification in analyzing sentiment was compared, and the highest accuracy achieved was 90% [12]. Hans Prayogo's research, discussing the use of CNN in video classification, has the advantage of distinguishing the visual format of movies with a relatively high accuracy of around 89% on the dataset used [13]. From these studies, the use of CNN is superior in image and video classification and rating classification.

This research contributes to applying content-based filtering in the Convolutional Neural Network (CNN) context. The main focus is to evaluate the development of a recommendation system by integrating and comparing similarity identification methods using RoBERTa and TF-IDF word representation approaches. The results obtained is a rating matrix that will be classified using CNN regarding whether the movie is recommended. This is because, as far as the author knows, research has yet to combine these methods and classifications and utilize semantic features and feature extraction. The motivation of this research is to provide accurate accuracy results for recommending movies relevant to user preferences by combining classification methods and utilizing semantic feature extraction and extraction in the system. In its application in content-based filtering, TF-IDF is built to provide movie recommendations based on similarity calculations to produce rating prediction values, which are then evaluated. The evaluation shows the error rate experienced by the performance. Similar to TF-IDF, Roberta is built to provide recommendations based on similarity calculation. However, the Roberta process, each rating prediction from TF-IDF and Roberta is used as input to build the model of the recommendation system, which is expected to provide optimal prediction accuracy in recommending relevant movies.

This research has a structure consisting of several sections. In the second section, there is a summary of related studies that address similar research topics and sub-sections that are relevant to the system design. Furthermore, the third section will more specifically review the proposed methodology, mainly focusing on building a recommendation system using content filtering and CNN classification, accompanied by some experiments as part of the discussion. Then, the fourth section will conclude with the development of the system with the proposed method and classification, along with the accuracy results obtained.

2. METHODS

The system designed in this research is explained through the illustration in Fig. 1.These steps involve data collection, pre-processing, feature extraction and semantic features, rank prediction, cosine similarity, Top Recommendation N, evaluation using MAE and RMSE, data sharing, use of classification models, and accuracy assessment.

2.1. Recommender System

Recommender systems are computer software and algorithms that suggest items that are most interesting to users [14]. In principle, recommender systems help filter and recognize items such as products, services, or information likely to be of interest, purchased, or used by users [15], [16]. Created to make it easier to find items by utilizing information about relevant features, recommendations of the most suitable items for users can be made by analyzing user patterns and characteristics through various factors [17], [18]. Using input information, recommender systems can predict user interests, and ratings from previous users are a crucial component in such data sets. This research aims to find items that match the description of the genre for which the similarity is calculated to help rank recommendations and then combine them into a list of recommendations relevant to the user [19].

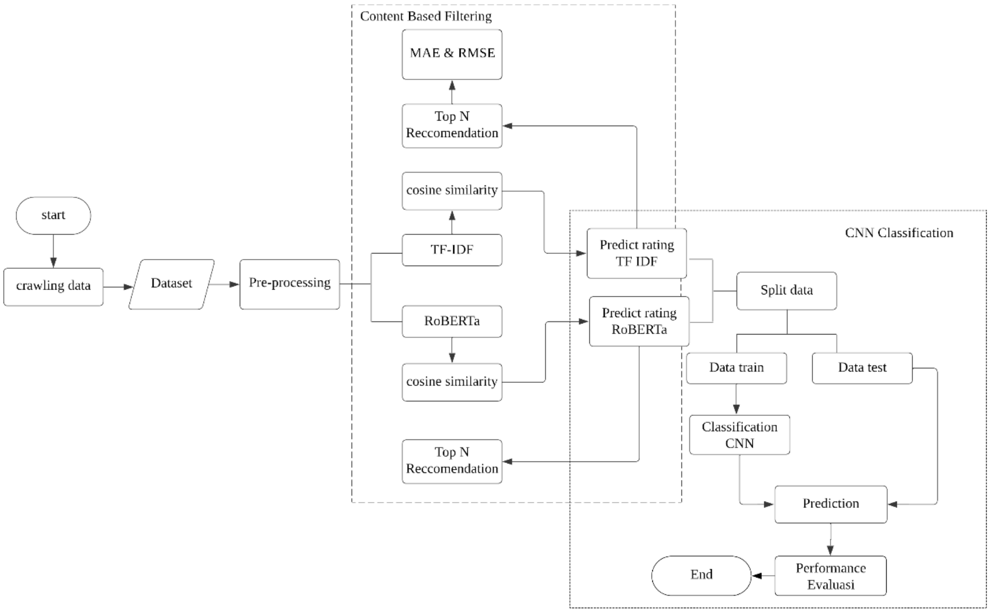

Fig. 1. CBF & CNN design system

2.2. Crawling Data

Using crawling techniques to gather data is an effective and efficient way to retrieve Twitter information. Designing research to automate data collection from Twitter can be more straightforward way to obtain information [20], [21]. In this study, two different types of data, which were collected from various sources were used. Initial crawling was done using the Netflix and Disney+ filters to retrieve film titles from the IMDB website. In addition, feature extraction was performed using the Python library PyMovieDb, which generates features such as name, description, and keywords. Some of the movies and their extracted features are shown in Table 1. The second crawling was done using Tweet-Harvest to obtain movie reviews from Twitter users known for their ability to evaluate movies; some of the reviews obtained from Twitter are shown in Table 2.

Table 1. Example of first crawling data

Film	Genre	Date Published
14 Cameras	[“Crime”, “Horror”, “Thriller”]	2018-07-27
17 Again	[“Comedy”, “Drama”, “Fantasy”]	2009-04-17
1BR	[“Drama”, “Horror”, “Thriller”]	2020-04-24

Table 2. Example of second crawling data

Username	Text
djaycoholyc	Grown Ups pertama ini masih asyik. Asyik banget
CenayangFilm	Film biografi indonesia yg paling pas emang baru azrax sih. Baru Gie dan Habibie Ainun.
danieldokter	Buset. Don't Knock Twice full house. Apa-apaan ini...

Previously, data collection was done by combining previously existing datasets with the addition of new datasets. There were 39 accounts without additional features in the previsious data and film title data without other features. Since content-based filtering requires genre, description, and others, a crawling process was carried out to add additional features. The research began by conducting an initial search on IMDB and applying a filter to filter out movies available on Netflix and Disney+ from January 1, 2022 to November 2, 2023. After successfully extracting the data, pyMovieDb was used to extract the required features. The results were then combined with data from previous research, resulting in 845 films withfeatures, namely type, name, URL, poster, description, review, content, genre, datePublished, keyword, duration, actor, director documented in Table 3. In addition, in the second crawling stage, we used a tweet collection technique to collect opinion about the films from 39 accounts with five additional recent account, namely IMDB, Metacritic metascore, Metacritic User Score, Rotten Tomatoes Tomato Meter, and Rotten Tomatoes Audience Score regularly provide film reviews by searching for film title keywords. Through keyword searches of film titles, this crawling process's results were combined with the review data that researchers had previously collected. The data that has been collected previously by researchers resulted in a total of 34,086 reviews, as shown in Table 4. The resulting tweet data was crawled using a total of 44 accounts, including 39 accounts and five additional recent accounts that have high credibility even over the years, providing recommendations based on preferences of experience in the film industry.

Table 3. Example of the first crawling result

Film	Genre	…	Date Published	Duration
14 Cameras	["Crime", "Horror", "Thriller"]	…	2018-07-27	PT1H30M
17 Again	["Comedy", "Drama", "Fantasy"]	…	2009-04-17	PT1H42M
…	…	…	…	…
3 Days to Kill	["Action", "Comedy", "Drama"]	…	2014-02-25	PT1H57M
3 Idiots	["Comedy", "Drama"]	…	2009-12-25	PT2H50M

Table 4. Example of the first crawling result

Username	Film	Text
AnakNonton	Thor: Ragnarok	Dengan $121 juta, 'Thor: Ragnarok' jadi film MCU dgn debut terbesar ke-7 sekaligus memuncaki box-office minggu ini!
AnakNonton	Headshot	Penata Efek Visual Terbaik #FFI2016 : Andi Novianto - 'Headshot' #MalamPuncakFFI2016
…	…	…
zavvi	Turning Red	Disney Pixar's #TurningRed hits @disneyplus today! Who else is watching this cute and cuddly coming of age film?
zavvi	What If	Well, new animated Marvel show What If...? looks like it will be plenty of fun #DisneyInvestorDay

2.3. Data Preprocessing

Data pre-processing is the initial stage in data management that involves a series of steps on text. Text pre-processing includes the steps of processing and preparing text data before analysis to removing unnecessary information from the statements [22]. Changing letters to lowercase, dividing text into words or tokens, filtering or removing irrelevant or unimportant words, and removing affixes to return words to their original form are all part of the data-cleansing process. All these steps are considered a crucial part of the process for users [23], [24]. In addition, the SMOTE method is needed to overcome the imbalance in the data [25].

This research, the preprocessing stage into three: cleaning text, translate text, and calculate blob. In cleaning text, cleaning includes removing "RT" characters, removing emojis, removing mentions, removing hashtags, removing links, removing non-alphanumeric characters except spaces, removing break rows, and converting text to lowercase. Then convert film review data from Twitter into a 0-5 rating system. This conversion process involves several steps, including translation, reviews are translated into English using the Python Deep Translator library. In addition, the text cleaning process aims to extract menaingful information from the reviews. In translate text, the language in the tweet is changed to English using the GoogleTranslator library. The polarity scoring process uses the Python TextBlob library to evaluate reviews on a 0-5 scale, to identify whether they are positive or negative. Textblob, a Python library, can be used to assess how authenticity is maintained after translation into English [26].The process generated 6,479 rows of data, including evaluations from five additional users: IMDB, Metacritic Metascore, Metacritic User Score, Rotten Tomatoes Tomato Meter, and Rotten Tomatoes Audience Score. Table 5 shows the results of the preprocessing stage.

In addition, additional pre-processing steps involve cleaning the data from too extreme values in the Polarity Score column in the dataset, performing column sets, handling missing values, and creating a new data frame to store film ratings from each account until normalization. Data normalization is a step taken before the data is processed. At this stage, values changes to make the processing easier. It is important to note that data normalization does not impact increasing memory usage or processing power [27]. In this case, it is implemented using MinMaxScaler to normalize the ranking values.

Table 5. Example of the first preprocessing result

Username	Film	Score
AnakNonton	3 Days to Kill	2.84
AnakNonton	65	2.89
…	…	…
zavvi	What If	3.26
zavvi	You People	2.68

2.4. Content Based Filtering

Implementing a content-based approach, the recommendation system provides recommendations for items that are similar to items previously favored or selected by the user. The degree of similarity between items is calculated based on the features of the compared items [28]. In addition, this method is commonly applied to filter based on the similarity of item descriptions in each description [29], [30]. Whether the item belongs to a category other consumer have never selected does not affect this approach. This method is suiting this final project research because it looks for recommendations based on certain similarities [31]. The content-based filtering methods recommend items based on description and genre. Applying the content-based filtering methods starts by building a TF-IDF vectorizer to calculate similarity to produce a rating prediction value for TF-IDF dan Roberta. This prediction value is used to find the Top N Recommendation from both approaches. TF-IDF is then evaluated.

2.4.1. Term Frequency – Inverse Document Frequency (TF-IDF)

To determine the relationship between words or phrases and documents, can use the Term Frequency-Inverse Document Frequency (TF-IDF) method [32]. TF-IDF is a method that measures the value or weight for each word (token) contained in a document in a particular document collection. Generally, this method is used for information retrieval and text development to evaluate the relationship between documents words [33], [34]. Terms frequently appearing in documents are valued through this normalization process. In other words, documents are converted into a metric that is calculated based on their frequency of occurrence [35]. This method is most commonly used to convert text into vectors, helping to calculate the frequency of use of certain words in documents [36]. This research usesTF-IDF as a vectorizer to convert text features into numeric vectors. The initial stage is to preprocess the text used. Next, the result is a matrix with the number of rows and columns. The initial stage is to preprocess the next used. Next. The result is a matrix with the number of rows and columns corresponding to the amount of data in dataset. Reach matrix element represents the weight value of the word in the next.corresponding to the amount of data in the dataset. Each element of the matrix represents the weight value of the word in the text. The methods used to obtain this weight value are methods (1) and (2) [37].

$$TF * IDF = TF(w_i, d) * IDF\ (w_i) \tag{1}$$

Method (2) is used to calculate the IDF value, where N is the total number of documents and DF (w_i) is the number of documents containing word (w_i).

$$IDF(w_i) = \frac{N}{DF(w_i)} \tag{2}$$

This research uses sci-kit-learn's "TfidfVectorizer" to convert the text in the "genre" and "description" columns into numerical vectors. As for the transition or flow, the cosine similarity matrix between films is calculated using the "cosine similarity" function. Table 6 shows the similarity of the cosine matrix. This method is used to evaluate the extent of similarity between items in the data by providing recommendations for a particular movie based on the cosine similarity value set and displayed on the corresponding row [38]. In making rating predictions at Table 7, the similarity is taken from the matrix by using TF-IDF which focuses on the content-based baseline. The TF IDF calculation matrix results will go through the cosine similarity process to calculate the similarity of items to provide top n film recommendations with the highest similarity value criteria. To assess the accuracy of the prediction, the performance of the model is assessed by Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) [39].

Table 6. Cosine Similarity Matrix TF-IDF result

	14Cameras	17Again	….	Ozel Ders
14Cameras	1.000000	0.000000	….	0.000000
17 Again	0.000000	1.000000	….	0.042062
….	……	…..	….	…..
Encontro	0.000000	0.587656	….	0.071575
Ozel Ders	0.000000	0.456880	….	1.000000

Table 7. TF-IDF with Content Based predictions result

Movie name	AnakNonton	----	zavvi
14Cameras	0.0421	…..	0.0335
17 Again	0.0896	…..	0.0495
….			
Encontro	1.4635	…..	0.0356
Ozel Ders	0.0324	….	0.3356

2.4.2. Word Embedding

Word embedding is a form of text data representation where each word is converted into a dense vector of numbers. The process of word embedding in text data results in a vector representation for each word. This semantic approach utilizes the Robustly Optimized BERT Pretraining Approach (RoBERTa) to convert words into number vectors. In this research, this method is applied together with the TF-IDF feature extraction technique. RoBERTa, which is an extension of Bidirectional Encoder Representations from Transformers (BERT), extends key hyperparameters to improve the quality of learning [40]. RoBERTa has the ability to predict parts of text that have been annotated. This model comes in two variants, namely RoBERTa large and RoBERTa base. As an alternative to the task of predicting the next sentence, RoBERTa recommends the use of dynamic masks, as described in previous research [41]. In addition to the dynamic mask method that creates a mask pattern each time a sequence is entered into the model, there is another approach that involves the removal of Next Sentence Prediction (NSP). NSP removal is followed by training with a large number of batches, aiming to speed up the optimization and improve the performance of the final task. In this context, the use of Byte-Pair Encoding (BEP), a technique that combines word and character representations, allows handling large vocabularies [42]. In this research, the RoBERTa model is used to create a tokeniser capable of converting text into a sequence of number tokens. This approach involves basic sequence token embedding and average RoBERTa basic structure embedding. Similar to TF-IDF, RoBERTa converts features into vectors that can be used to assess how similar items are to each other. The result is a matrix similar to TF-IDF, as described in Table 8. It is this matrix that will help recommend films with the highest rating value. However, Roberta will not be evaluated with MAE and RMSE because the next application of Roberta is used as an experiment [37]. The prediction results are presented in Table 9 [39].

Table 8. Cosine Similarity Matrix RoBERTa result

name	14Cameras	17Again	….	Ozel Ders
14Cameras	1.000000	0.965242	….	0.983266
17 Again	0.965242	1.000000	….	0.957397
….	……	…..	….	…..
Encontro	0.962303	0.994109	….	0.957875
Ozel Ders	0.983266	0.957397	….	1.000000

Table 9. Cosine Similarity RoBERTa result

Movie name	AnakNonton	----	zavvi
14Cameras	0.0421	…..	0.0335
17 Again	1.0246	…..	2.365
….			
Encontro	1.0242		02412
Ozel Ders	0.2461	…..	0.0356

2.5. Convolutional Neural Network (CNN) Classification

Deep learning models such as Convolutional Artificial Neural Networks (CNNs) are based on the principle of human brain function [43], [44]. CNNs effectively manage data with a network structure, such as grid [45]. The CNN structure has three dimensions: one dimension is used to process text and signals, two dimensions to process images or audio, and three to analyze video content. Although commonly used in computer vision to classify images [46], CNNs can also perform text classification well. In text classification, the input is a word vector formed through a word concatenation method [47], similar to the approach applied to image classification. Google created a framework called TensorFlow for machine learning because Tensors are crucial in machine learning. With the concept of tensors, we can generalize vectors and matrices to various dimensions more flexibly [48]. At CNN, data that has been labeled 0 - 1 is used. The characteristic of data labeled 0 is if the rating value is <0.5, while data labeled 1 is if the rating value is > 0.5. Using the Keras TensorFlow library, this research created several structures of the CNN model. The various levels of the structure involve input layer, convolution layer, max-poolinglayer, flatten layer, dense layer, dropout layer and output layer [49]. These functions act as steps where input data is included for processing through the convolution layer [50]. Since the nature of the data used is one-dimensional text, this study utilizes a 1D convolution layer. With a 1D CNN, the model can extract more complex patterns and features from the layers above it, allowing the identification of simple patterns in the observed data [51]. After performing Conv1D, a dropout level of 0.5 prevent overfitting by silencing a random number of neurons during the training process [52]. In the next step, Max Pooling1D is used to take the maximum value of the pooling window to order reduce the data's dimensionality. In anticipation of overfitting after MaxPooling, Dropout was added once again with a level of 0.5. Before entering the fully connected layer, a smoothing process converts the feature matrix into a vector. After the flatten stage, another Dropout was performed with a level of 0.5. Next, the model uses a dense layer with 64 units and applies ReLU activation. To minimize the risk of overfitting, an additional Dropout is applied. In order to generate class probabilities independently, the model is equipped with a dense layer of three units and uses sigmoid activation.CNN was chosen as a classification because, based on several related studies, CNN can provide accurate and optimal accuracy prediction result and the ability to recognize patterns even in rating. Fig. 2 shows an overview of the CNN arcithectural design.

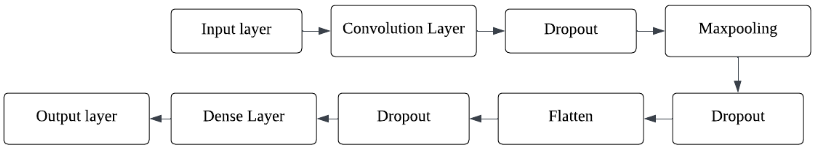

Fig. 2. CNN model architecture

3. RESULTS AND DISCUSSION

3.1. Content Based Filtering Result

The content-based filtering method integrates film genre and description fields into a dataframe. The TF-IDF method is used to extract relevant features from the "feature value" column, which considers both features. TF-IDF is applied to convert the text features in genre and description into numerical vectors, which are then calculated for cosine similarity, and the result is a matrix. In Table 10, results of item similarity calculation in dataframe "cosine_similarity_df" Table 11, the prediction value of the content-based filtering process by applying the TF-IDF Method is shown. TF-IDF uses this prediction value to obtain the top N recommendations. Evaluation of the results shows that the tested model has a low error rate, with Mean Absolute Error (MAE) of 0.28 and a Root Mean Square Error (RMSE) of 0.67.

Table 10. Cosine Similarity Matrix TF-IDF result

	14Cameras	17Again	….	Ozel Ders
14Cameras	1.000000	0.000000	….	0.000000
17 Again	0.000000	1.000000	….	0.042062
….	……	…..	….	…..
Encontro	0.000000	0.587656	….	0.071575
Ozel Ders	0.000000	0.456880	….	1.000000

Table 11. TF IDF with Content Based prediction results

Movie name	AnakNonton	----	zavvi
14Cameras	0.0421	…..	0.0335
17 Again	0.0896	…..	0.0495
….			
Encontro	1.4635	…..	0.0356
Ozel Ders	0.0324	….	0.3356

3.2. Classification Result

Classification is done using CNN. Classification data comes from the CBF processing results labeled 0 and 1 with the characteristics of 0 if the rating value < 5 and 1 if the rating value > 6, so the total label 0 data is 32.929 and the total label 1 data is 6.355. Due to this unbalanced label data, it is necessary to apply SMOTE. The focus of this research is on three experiments. The first experiment focused on applying SMOTE to the split ratio, and the optimal result of the scenario was used as the baseline for the next experiment. The second experiment evaluated the comparison between the baseline, SMOTE and Embedding, and analysed the percentage of improvement that occurred compared to the first experiment. The third experiment used the SMOTE method with the baseline, Embedding and optimizer, then compared with the results of the previous experiments and evaluated based on the previously obtained results. This step is carried out to obtain optimal results and the best performance by applying the CNN model.

3.2.1. CNN Baseline SMOTE Model

In this section, experiments are conducted on the baseline using CNN. This experiment uses input data to predict rating values from TF-IDF with the dataframe name “ResultCBFTFIDF”, labeled before and normalized to a scale of 0 – 1. This step aims to find optimal results by utilizing comparison as a divisor. In this research, the comparisons used include 90:10, 80:20, 70:30, and 60:40. The test results can be found in Table 12.

Table 12. Results of the first experiment

Split Ratio	Accuracy (%)
	CNN
90:10	**78.83**
80:20	77.90
70:30	78.33
60:40	77.53

The first experiment involved using the baseline method SMOTE. SMOTE plays a role in increasing the number of samples in the minority class or decreasing the number in the majority class, aiming to address data imbalance that can lead to inaccurate classification results [53]. Table 12 shows the baseline and SMOTE results for the CNN model. The table shows that the highest accuracy is achieved when the data is divided by a ratio 90:10, which is 78.83%. Therefore, the 90:10 ratio will be used consistently in the following scenarios.

3.2.2. RoBERTa Implementation as Word Embedding

The second experiment used baseline, SMOTE, and Embedding to compare the accuracy results. Roberta embedding was chosen in this study because it can convert text into number tokens. Its advantage is the ability to train the model for a longer duration, which can help the model extract and understand more complex patterns in the data so that the prediction results in the classification are more optimal. In the second experiment, the input data used is Roberta’s rating prediction value with the data frame name “Resultroberta”, which has labeled and the feature scale normalized using MinMaxScaler to make the feature scale range [0,1]. Details of the evaluation are recorded in Table 13. The results of the second experiment showed an increase in accuracy to 82.08%, an increase of about 4.12% from the first experiment, which only got an accuracy of 78.83%. The improvement seen from combining the baseline method,SMOTE with embedding can be found in Table 13.

Table 13. Results of the second experiment

Baseline	Accuracy (%)
	CNN
Baseline + SMOTE + Embedding	82.08 (+4.12)

3.2.3. Model Optimization

The third experiment included baseline, SMOTE, embedding, and optimizer implementation. These actions were performed to overcome the challenges in structured optimization while performing the analysis. The goal was achieved using various optimizer methods, such as Adam, SGD, Adadelta, Adagrad, RMSprop, and Nadam. The input data in the third experiment was “Resultroberta”, which was already labeled and feature-scaled. The study applied fifty epochs and was repeated six times to understand the performance better. The Details of the evaluation are recorded in Table 14.

Table 14. Results of the third experiment

	Accuracy (%)
	CNN
Baseline + SMOTE + Embedding	
	82.08 (+4.12)
Adadelta	59.43 (-24.61)
Adagrad	80.02 (+1.51)
RMSprop	82.12 (+4.17)
Nadam	82.97 (+5.25)
Adam	83.53 (+5.96)
SGD	**84.64 (+7.37)**

Table 14 shows that the SGD optimization resulted in an accuracy improvement of 84,64%, an increase of about 7.37% from the first experiment. Then, the most critical factor is finding the optimal value for the learning rate. This improvement shows that the SGD optimizer can control the model's performance at the classification stage. Table 15 shows a very significant improvement after finding the best learning rate. The findings show that the SGD optimization approach achieved the highest accuracy of 86.41%, improving 9.62% over the first experiment.

Table 15. Best Learning Rate of of SGD

Optimizer	Learning Rate	Accuracy (%)
SGD	3.906939937054621e-06	86.41 (+9.62)

3.3. Discussion

Applying the content-based filtering (CBF) method to the film recommendation system in this study provides significant results with a combination of classification so that it can recommend film with an accuracy of 86.41%. Previous research has examined how CBF can recommend films without applying classification and using around 4000 film titles with genre and other synopsis features. The research achieved an accuracy of 0.82% for a single query and 0.75% for multiple queries [8]. So, there is a comparison after the contribution of applying classification, semantic features, and feature extraction to CBF. The graph in Fig. 3 displays data illustrating the performance improvement for experiment one to three.

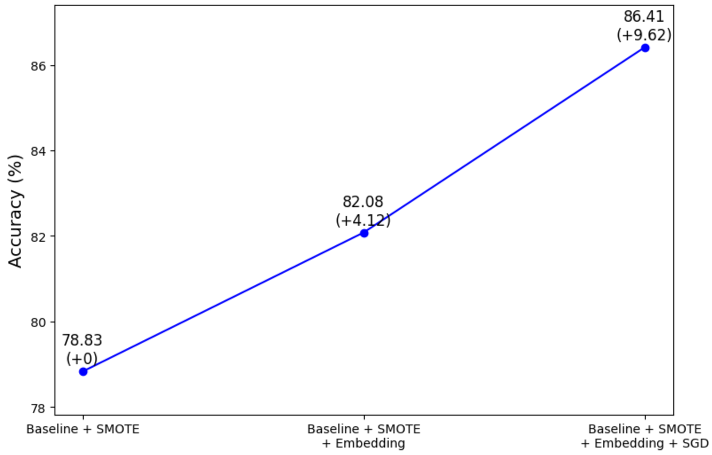

Fig. 3. Relative Increase in Accuracy

In this research, SMOTE is applied to handle the imbalance in the data. After the labeling process and normalizing the values to 0-1, the results obtained from the values 0 and 1 need to be balanced, namely, the total data label 0 as much as 32.929 and the total data 1 as much 6.355. This data imbalance can lead to inaccurate classification results [53]. SMOTE was applied together with feature extraction in the first experiment as a baseline by dividing four ratios of 90:10, 80:20, 70:30, and 60:30 on CNN classification. In the process, it was found that 90:10 ratio can be used as a data diriver in the following experiment with an accuracy of 78.83%. The second experiment applied the baseline, SMOTE, with Roberta embedding on CNN classification. The research obtained an accuracy of 82.08%, with an improvement of 4.12% from the first experiment.

The third experiment applied baseline, SMOTE, embedding, and optimizer to CNN classification. The are six optimizers, namely Adam, SGD, Adagrad, Adadelta, RMSprop, and Nadam, used to find which optimizer comparison can increase accuracy from the previous experiment. Based on Table 14, the SGD optimizer can provide an accuracy of 84.64%, with an increase of 7.37% from the first experiment. This is due to the optimizer. The SGD optimizer can control the model's performance during the classification accuracy performance. The SGD optimizer needs to find learning rate. Finding the learning rate is done with an epoch 50 with a period of 5 times, where epoch 50 will be repeated five times to find the best learning rate. By finding the best learning rate, it can see how far the method works on CNN classification to show improved results and provide film recommendations based on accurate predictions. Table 15 shows that by finding the best learning rate, the model's performance has increased prediction accuracy by 86.41% with an increase of 9.62% form the first experiment. Therefore, SGD was chosen as the optimizer algorithm in this study because of its ability to adjust the learning rate for each parameter during the training process to provide an accurate film recommendation, this has been investigated and documented in [54].

4. CONCLUSION

The results prove the accuracy of applying the CBF method to the recommendation system with classification, feature extraction, and semantic features. This research is superior to previous research recommending movies based on relevant user preferences [8]. The combination of extraction and semantic features for word representation and weight assessment approaches to produce numbers or tokens results from research that is rarely done, as far as the author's knowledge, still needs to be improved. With a limited dataset of 845 movies and 34,086 tweet reviews, the research was able to run three experiments and discover the best learning rate that increased the accuracy to 86.41%. Thus, the built recommendation system can recommend movies on streaming services or movie recommendation platforms because it successfully provides optimal prediction results to recommend movies based on user preferences.

It is hoped that this research in the future can be combined again with collaborative filtering or hybrid filtering methods, conduct experiments related to the use of different extraction features or semantic features and more datasets in order to be able to provide more optimal accuracy results from current research in order to be able to provide movie recommendations with user preferences accompanied by more optimal accuracy.

REFERENCES

[1] C. I. Rotună, M. Dumitrache, I. E. Sandu, "Assessment of Machine Learning algorithms for automated monitoring," *Romanian Journal of Information Technology & Automatic Control/Revista Română de Informatică şi Automatică*, vol. 32, no. 3, 2022, https://openurl.ebsco.com/EPDB%3Agcd%3A13%3A20837197/detailv2?sid=ebsco%3Aplink%3Ascholar&id=ebsco%3Agcd%3A159507519&crl=c.

[2] R. A. Abarja, "Movie Rating Prediction using Convolutional Neural Network based on Historical Values," *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 5, pp. 2156–2164, 2020, https://doi.org/10.30534/ijeter/2020/109852020.

[3] A. Chauhan, D. Nagar, and P. Chaudhary, "Movie recommender system using sentiment analysis," in *Proceedings of International Conference on Innovative Practices in Technology and Management, ICIPTM*, pp. 190–193, 2021, https://doi.org/10.1109/ICIPTM52218.2021.9388340.

[4] S. Tobias, "The Future of Film Talk Is on Letterboxd," *The Ringer,* Accessed: Dec. 26, 2023. [Online]. Available: https://www.theringer.com/movies/2020/9/18/21444082/letterboxd-filmdiscussion-site-streamingmovies.

[5] Z. Yu, "Netflix recommends: algorithms, film choice, and the history of the taste," *Inf Commun Soc*, pp. 1–3, 2023, https://doi.org/10.1080/1369118x.2023.2252485.

[6] J. Sturgill, "Beyond the Castle: An Analysis of the Strategic Implications of Disney+," *East Tinnessee State University,* 2019. [Online]. Available: https://dc.etsu.edu/honors/492.

[7] R. J. Limaye *et al.,* "Social media strategies to affect vaccine acceptance: a systematic literature review," *Expert review of vaccines,* vol. 20, 8, pp. 959-973, 2021, https://doi.org/10.1080/14760584.2021.1949292.

[8] A. H. J. P. J. Permana, 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.

[9] D. Suhartono, K. Purwandari, N. H. Jeremy, S. Philip, P. Arisaputra, and I. H. Parmonangan, "Deep neural networks and weighted word embeddings for sentiment analysis of drug product reviews," in *Procedia Computer Science*, pp. 664–671, 2022, https://doi.org/10.1016/j.procs.2022.12.182.

[10] J. Mathew, G. S. Kumar, D. P., and J. M. Jose. *Responsible Data Science*. in *Lecture Notes in Electrical Engineering*, vol. 940, 2022, https://doi.org/10.1007/978-981-19-4453-6.

[11] M. R. Aldiansyah and P. S. Sasongko, *Twitter Sentiment Analysis About Public Opinion on 4G Smartfren Network Services Using Convolutional Neural Network.*, in *International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 1-6, 2019, https://doi.org/10.1109/icicos48119.2019.8982429.

[12] S. Seo, C. Kim, H. Kim, K. Mo and P. Kang, "Comparative Study of Deep Learning-Based Sentiment Classification," in *IEEE Access*, vol. 8, pp. 6861-6875, 2020, https://doi.org/10.1109/ACCESS.2019.2963426.

[13] A. M. Conway, I. N. Durbach, A. McInnes, R. N. Harris, "Frame-by-frame annotation of video recordings using deep neural networks," *Ecosphere*, vol. 12, no. 3, p. e03384, 2021, https://doi.org/10.1002/ecs2.3384.

[14] A. A. S. Gunawan, and D. Suhartono, "Music recommender system based on genre using convolutional recurrent neural networks," in *Procedia Computer Science*, pp. 99–109, 2019, https://doi.org/10.1016/j.procs.2019.08.146.

[15] J. B. Schafer, J. A. Konstan, J. Riedl, "E-commerce recommendation applications," *Data mining and knowledge discovery*, vol. 5, pp. 115-153, 2021, https://link.springer.com/article/10.1023/A:1009804230409.

[16] Y. Deldjoo, M. Schedl, P. Cremonesi, G. Pasi, "Recommender systems leveraging multimedia content," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1-38, 2020,

[17] S. Jayalakshmi, N. Ganesh, R. Čep, and J. S. Murugan, "Movie Recommender Systems: Concepts, Methods, Challenges, and Future Directions," *Sensors*, vol. 22, no. 13, 2022, https://doi.org/10.3390/s22134904.

[18] S. Gupta, "A Literature Review on Recommendation Systems," *International Research Journal of Engineering and Technology*, vol. 7, pp. 3600-3605. 2020, [Online]. Available: www.irjet.net.

[19] B. Hawashin, M. Lafi, T. Kanan, A. Mansour, "An efficient hybrid similarity measure based on user interests for recommender systems," *Expert Systems*, vol. 37, no. 5, p. e12471, 2020, https://doi.org/10.1111/exsy.12471.

[20] D. Samalo, R. Martin, D. N. Utama, "Improved Model for Identifying the Cyberbullying based on Tweets of Twitter. Informatica," *Informatica*, vol. 47, no. 6, 2023, https://doi.org/10.31449/inf.v47i6.4534.

[21] C. Wintang 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, 2020, https://doi.org/10.29207/resti.v4i4.2038.

[22] H. T. Duong and T. A. Nguyen-Thi, "A review: preprocessing techniques and data augmentation for sentiment analysis," *Comput Soc Netw*, vol. 8, no. 1, 2021, https://doi.org/10.1186/s40649-020-00080-x.

[23] M. A. Rosid, A. S. Fitrani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi," in *IOP Conference Series: Materials Science and Engineering*, vol. 874, no. 1, p. 012017, 2020. https://doi.org/10.1088/1757-899X/874/1/012017.

[24] K. Demestichas, N. Peppes, T. Alexakis, E. Adamopoulou, "Blockchain in agriculture traceability systems: A review," *Applied Sciences*, vol. 10, no. 12, p. 4113, 2020, https://doi.org/10.3390/app10124113.

[25] X. W. Liang, A. P. Jiang, T. Li, Y. Y. Xue, G. T. Wang, "LR-SMOTE—An improved unbalanced data set oversampling based on K-means and SVM," *Knowledge-Based Systems*, vol. 196, p. 105845, 2020, https://doi.org/10.1016/j.knosys.2020.105845.

[26] 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)," *International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS)*, pp. 1-6, 2022, https://doi.org/10.1109/ICACNIS57039.2022.10055248.

[27] C. Garbin, X. Zhu, O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimedia Tools and Applications*, vol. 79, pp. 12777-12815, https://doi.org/10.1007/s11042-019-08453-9.

[28] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook: Third Edition*. Springer US, 2022, https://doi.org/10.1007/978-1-0716-2197-4.

[29] M. J. Mokarrama, S. Khatun, and M. S. Arefin, "A content-based recommender system for choosing universities," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 28, no. 4, pp. 2128–2142, 2020, https://doi.org/10.3906/ELK-1911-37.

[30] S. Natarajan, S. Vairavasundaram, S. Natarajan, A. H. Gandomi, "Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data," *Expert Systems with Applications*, vol. 149, p. 113248, 2020, https://doi.org/10.1016/j.eswa.2020.113248.

[31] D. Baxter *et al.,* "Recommended conventions for reporting results from direct dark matter searches," *The European Physical Journal C*, vol. 81, pp. 1-19, 2021, https://doi.org/10.1140/epjc/s10052-021-09655-y.

[32] M. I. Alfarizi, L. Syafaah, M. Lestandy, "Emotional Text Classification Using TF-IDF (Term Frequency-Inverse Document Frequency) and LSTM (Long Short-Term Memory)," *JUITA: Jurnal Informatika*, vol. 10, no. 2, pp. 225-232, 2022, http://dx.doi.org/10.30595/juita.v10i2.13262.

[33] S. W. Kim and J. M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, 2019, https://doi.org/10.1186/s13673-019-0192-7.

[34] T. D. Putra and E. B. Setiawan, "Sentiment Analysis on Social Media With Glove Using CNN Model and RoBERTa," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 3, pp. 457-563. 2023, https://doi.org/10.29207/resti.v7iX3.4892.

[35] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif Intell Rev*, vol. 52, no. 1, pp. 273–292, 2019, https://doi.org/10.1007/s10462-018-09677-1.

[36] E. D. Canedo, B. C. Mendes, "Software requirements classification using machine learning algorithms," *Entropy*, vol. 22, no. 9, p. 1057, 2020, https://doi.org/10.3390/e22091057.

[37] A. Boluki, J. P. R. Sharami, D. Shterionov, “Evaluating the Effectiveness of Pre-trained Language Models in Predicting the Helpfulness of Online Product Reviews,” *arXiv preprint arXiv:2302.10199*, 2023, https://doi.org/10.48550/arXiv.2302.10199.
[38] M. A. Budiman and Gst. A. V. M. G. Giri, “Song Recommendations Based on Artists with Cosine Similarity Algorithms and K-Nearest Neighbor,” *J. Elektron. Ilmu Komput,* vol. 8, no. 4, pp. 2654–5101, 2019, https://jurnal.harianregional.com/index.php/JLK/article/view/53174#google_vignette.
[39] T. O. Hodson, “Root mean square error (RMSE) or mean absolute error (MAE): when to use them or not,” in the *literature. Geoscientific model development*, vol. 7, no. 3, pp. 1247-1250, 2022, https://doi.org/10.5194/gmd-2022-64.
[40] A. Barua, S. Thara, B. Premjith, and K. P. Soman, “Analysis of Contextual and Non-contextual Word Embedding Models for Hindi NER with Web Application for Data Collection,” in *Communications in Computer and Information Science*, pp. 183–202, 2021, https://doi.org/10.1007/978-981-16-0401-0_14.
[41] R. Khusuma, W. Maharani, and P. H. Gani, “Personality Detection On Twitter User With RoBERTa,” *Jurnal Media Informatika Budidarma*, vol. 7, no. 1, pp. 542-553. 2023, https://doi.org/10.30865/mib.v7i1.5598.
[42] Y. Liu *et al.*, “RcoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv*, 2019, [Online]. Available: http://arxiv.org/abs/1907.11692.
[43] D. Alsaleh and S. Larabi-Marie-Sainte, “Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms,” *IEEE Access*, vol. 9, pp. 91670–91685, 2021, https://doi.org/10.1109/ACCESS.2021.3091376.
[44] J. Kufel *et al.,* “What is machine learning, artificial neural networks and deep learning?—Examples of practical applications in medicine,” *Diagnostics*, vol. 13, no. 15, p. 2582, 2023, https://doi.org/10.3390/diagnostics13152582.
[45] J. Heaton, “Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning,” *Genet Program Evolvable Mach*, vol. 19, no. 1–2, pp. 305–307, 2018, https://doi.org/10.1007/s10710-017-9314-z.
[46] S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos, “Convolutional Neural Networks for Toxic Comment Classification,” in *Proceedings of the 10th hellenic conference on artificial intelligence*, pp. 1-6, 2018, https://doi.org/10.1145/3200947.3208069.
[47] N. I. Widiastuti, “Convolution Neural Network for Text Mining and Natural Language Processing,” in *IOP Conference Series: Materials Science and Engineering*, no. 5, p. 052010,2019, https://doi.org/10.1088/1757-899X/662/5/052010.
[48] O. A. Montesinos López, A. M. López, and J. Crossa, “Convolutional Neural Networks,” in *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, pp. 533–577, 2022, https://doi.org/10.1007/978-3-030-89010-0_13.
[49] I. Dhall, S. Vashisth and G. Aggarwal, "Automated Hand Gesture Recognition using a Deep Convolutional Neural Network model," *10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 811-816, 2020, https://doi.org/10.1109/Confluence47617.2020.9057853.
[50] E. C. Nisa and Y. Der Kuan, “Comparative assessment to predict and forecast water-cooled chiller power consumption using machine learning and deep learning algorithms,” *Sustainability (Switzerland)*, vol. 13, no. 2, pp. 1–18, 2021, https://doi.org/10.3390/su13020744.
[51] M. Azizjon, A. Jumabek, and W. Kim, “1D CNN based network intrusion detection with normalization on imbalanced data,” in *International Conference on Artificial Intelligence in Information and Communication, ICAIIC*, pp. 218–224, 2020, https://doi.org/10.1109/ICAIIC48513.2020.9064976.
[52] W. Ramadhanti and E. B. Setiawan, “Topic Detection on Twitter Using Deep Learning Method with Feature Expansion GloVe,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 3, pp. 780–792, 2023, https://doi.org/10.26555/jiteki.v9i3.26736.
[53] A. Zahri, R. Adam, and E. B. Setiawan, “Social Media Sentiment Analysis using Convolutional Neural Network (CNN) dan Gated Recurrent Unit (GRU),” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 1, pp. 119–131, 2023, https://doi.org/10.26555/jiteki.v9i1.25813.
[54] 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, https://doi.org/10.14201/adcaij2020927990.

BIOGRAPHY OF AUTHORS

Arliyanna Nilla is an undergraduate student of the Faculty of Informatics at Telkom University, Bandung, Indonesia. Email: arliyannanilla@student.telkomuniversity.ac.id.

Erwin Budi Setiawan is a senior lecturer at the Faculty of Informatics at Telkom University, Bandung. He has more than ten years of research and teaching experience in Informatics. Currently, he is an Associate Professor with research interests in machine learning, people analytics and social media analytics. Email: erwinbudisetiawan@telkomuniversity.ac.id.