Research Article

Targeted Advertising in Social Media Platforms Using Hybrid Convolutional Learning Method besides Efficient Feature Weights

Seyed Mohsen Ebadi Jokandan,1 Peyman Bayat,1 and Mehdi Farrokhbakhht Foumani2

1Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Iran
2Department of Computer Engineering, Fouman and Shaft Branch, Islamic Azad University, Fouman, Iran

Correspondence should be addressed to Peyman Bayat; drbayat.iau0@gmail.com

Received 11 May 2022; Revised 12 September 2022; Accepted 17 September 2022; Published 10 October 2022

Academic Editor: Mohamed Louzazni

Copyright © 2022 Seyed Mohsen Ebadi Jokandan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Advertising has been one of the most effective and valuable marketing tools for many years. Utilizing social media networks to market and sell products is becoming increasingly prevalent. The greatest challenges in this industry are the high cost of providing content and posting it on social networks, maximizing ad efficiency, and limiting spam advertisements. User engagement rate is one of the most frequently employed metrics for measuring the effectiveness of social media advertisements. Previous research has not comprehensively analyzed the factors influencing engagement rate. To this end, it is necessary to investigate the impact of various factors (such as user characteristics, posts, emotions, relationships, images, and backgrounds, among others) on engagement rate because assessing these influential factors in different networks can increase the engagement of users with advertising posts and thereby increase the success rate of targeted advertising. To predict the user engagement rate, we extract the significant attributes of posts and introduce an adaptive hybrid convolutional model based on FW-CNN-LSTM. We cluster the selected data based on the weight and significance of their attributes using the FCM and XGBoost algorithms and then apply CNN- and LSTM-based methods to select similar features. Using accuracy, recall, F-measure, and precision metrics, we compared our algorithm to standard techniques such as SVM, Logistic regression, Naïve Bayes, and CNN. According to the findings, hashtag, brand ID, movie title, and actors achieve the highest scores, and the values for actual training time in various data ratios are relatively linear, which confirms the scalability of the proposed model for large datasets. The results also demonstrate that our proposed method outperforms others and can lead to targeted ads on social media.

1. Introduction

The majority of advertising budgets are allocated to social media advertising. By 2021, global advertising expenditures are projected to reach 757.44 billion USD. The three principal players in online advertising are as follows: (i) the company desiring to serve online advertisements, known as the advertiser; (ii) the company delivering the advertisements via its website, known as the publisher; and (iii) the individual browsing the publisher’s website and being exposed to advertisements, known as the customer. Social media marketing has become very appealing to advertisers due to the growing number of active users and the ability to establish one-to-one relationships with customers [1]. Consequently, advertising companies are used to enhance campaign targeting and define better advertisement placement strategies [2].

Companies that invest significant funds in advertising expect a successful promotion and increased sales. Effective advertising is a challenge for advertisers and companies alike. Businesses consider each advertisement that influences the audience and becomes popular with them. Influencers and well-known locations can be utilized in advertisements to attract the audience’s attention.

Recent research has employed popularity metrics to predict and analyze Instagram popularity. These metrics
include the popularity prediction [3, 4], the fake users [5], the number of likes [6–9], the engagement rate [10], the intrinsic popularity [11, 12], and virality [13]. Numerous factors, including hashtags [14], metadata, visuals [7, 9], and sentiments [15], were considered during the analysis. The engagement rate, which is the number of likes per follower, is the most popular metric for measuring popularity. Another report suggested alternative metrics, including the growth, reach, and impression of followers [16]. The engagement rate represents the reaction of users based on quantifiable metrics such as likes and comments. Increasing engagement rate on social networks, particularly Instagram, is a factor of competition for pages that advertise products. Prior research highlighted the evaluation of post quality and engagement rate for user-generated text content. However, they failed to consider image attributes and additional contexts, including emotional attributes, captions, and hashtags of published posts.

Because of massive gains in processing capability and data storage, deep learning (DL) and machine learning (ML) have attracted a lot of interest in recent years. The approaches may be applied to a variety of real-world systems [17–19]. Clustering is a technique for analyzing market data. Data clustering is used to identify a classification in which data share hidden characteristics [20]. These techniques are highly effective during the early stages of data analysis when the dataset is unknown. One of the most significant clustering methods is fuzzy c-means clustering, which faces difficulties with high-dimensional datasets and numerous archetypes [21]. Moreover, the performance of the fuzzy c-means algorithm (FCM) is strongly influenced by the selection of the primary centroid clusters [22]. Recently, deep learning algorithms, particularly convolutional neural (CNN) and long short-term memory (LSTM) networks, have been applied to marketing for more accurate analysis. Deep learning is utilized in various marketing fields, including marketing intention detection, advertising, and data mining [23]. For instance, a great number of articles have studied the application of semantic analysis with deep learning techniques on the texts communicated in social networks. Through the Seq2Seq dialect normalization and the transformers analysis, Chennafi et al. [24] analyzed the Arabic texts. By decreasing the number of OOV (out of vocabulary) words in the current technique, a Seq2Seq model for dialect normalization may operate as a pre-processing stage for the classification task of ABSA (aspect-based sentiment analysis). As a result, the accuracy of the model is improved. Recognizing the types of advertising texts, that is, offensive language, racism, hatred, and entire types of verbal violence, is of utmost importance. The classification system developed by Aldjanabi et al. [25] aimed at determining hate and offensive speech via multitask learning (MTL) model, generated on the basis of a pre-trained language model. The developed multitask learning model indicated a noticeable performance, which was much better compared to that of the other models on three out of four datasets for the purpose of detecting hate and offensive speeches. A similar study was also carried out by Fan et al. [26] on the tweets published on the Twitter social network platform in order to categorize and identify the hashtags written about UK Brexit. According to the results, their suggested model could analyze and classify toxic tweets efficiently. In order to comprehend the ad complicity criteria on social media platforms, Ghanbarpour et al. [27] employed statistical techniques. According to their results, consumers featuring higher perceived ad complicity experienced higher perceived ad intrusiveness.

This study demonstrates the feasibility and utility of a hybrid convolutional model based on FCM and Extreme Gradient Boosting (XGBoost) for evaluating and estimating the accuracy of user engagement with offered advertisements, preventing spam advertisements, and lowering the cost for advertisers. We cluster the selected data based on the weight and significance of their attributes using the FCM algorithm and the XGBoost method, and then, we use CNN- and LSTM-based methods to train and predict the user engagement rate with a published advertisement. Innovations in our method are as follows:

(i) Utilizing and extracting a dataset of the promoted post’s primary attributes and context

(ii) Determining the weight of effective attributes on user engagement rate and providing a clustering technique based on XGBoost and FCM methods

(iii) Employing a hybrid convolutional algorithm based on the feature-weight (FW) and CNN-LSTM algorithms to learn and predict the user engagement rate in promotions

This research is structured as follows: in the first section, we examine the existing literature on evaluating the efficacy of advertising in social networks and machine learning algorithms. Section 2 evaluates the user engagement rate with advertising using a deep learning methodology. The final section discusses the results, outcomes, and future works.

2. Background

2.1. Engagement Rate. In practice, advertisers may not be able to participate in every ad auction due to their limited budget and a large number of online users. They must first identify potential user interests and then bid on audiences whose interests align with the advertiser’s advertising campaigns [28]. Numerous studies use the engagement rate as an essential criterion for measuring the success rate of targeted advertising. Accurate prediction of engagement rate reveals user interest and response to published advertisements.

In several studies, the engagement rate with online advertising as measured by the click-through rate (CTR) has been analyzed [29]. Tree structures and parameters used to extract the CTR of a new impression are constructed using historical data. Commonly used generative models include hierarchical Bayesian frameworks [30] and CTR hierarchy trees [31]. The natural advantage of a generative model is its interpretability, which enables businesses to determine which factor(s) improves CTR values the most. However, due to model limitations, these methods can frequently only
estimate a limited number of parameters (e.g., by using several chosen factors to break down the tree hierarchy). In addition, they cannot accurately estimate CTR by incorporating a significant amount of useful information from websites, publishers, and users. On the other hand, the increasing use of machine learning (deep learning in particular) has given rise to a group of predictive modeling techniques that view user clicks as binary events and train a classifier through supervised learning to estimate the probability of an advertisement impression being clicked [32]. Several CTR estimation methods based on deep neural networks are included in these techniques [33]. Generally, these techniques can manage tens of thousands of features and are typically more robust than generative models. However, when it comes to interpretability, they are not entirely transparent. Hong et al. [34] used a combined convolutional method based on a recurrent neural network (RNN) to classify and extract the users’ interests in social networks. The results showed that combining text and image information could improve classification precision. Yap [35] evaluated Facebook engagement rates to determine the efficacy of library advertisements. To this end, they analyzed users’ average Facebook page engagement rates kept in university libraries. Luke and Suharjito [36] investigated the use of Twitter to post product advertisements. Their proposed method measured the effectiveness of advertisements by analyzing user engagement with advertising tweets. Consequently, they utilized the Naive Bayes algorithm to categorize and estimate the engagement rate of Twitter followers based on the products or services they promoted.

In their study, Bonilla-Quijada et al. [37] measured the effectiveness of urban tourism advertising using the engagement rate. They analyzed the number of likes and comments posted on Instagram to determine user engagement rate and then measured the effectiveness of published ads. According to Zheng et al. [38], user engagement directly affects brand loyalty. They evaluated user engagement by creating online communities that encouraged other users to engage. In a separate study, Kim et al. [39] examined the effects of various Facebook content types on the engagement rate of online users. It was observed that educational posts had the greatest impact on online users. In addition, engagement was identified as a significant factor in disseminating knowledge through online tools. Stefko et al. [40] observed that the success of a promotional post is contingent on maximizing users’ engagement with that post via likes, comments, and sharing, among other activities. Gasparoni [41] examined the correlation between user age and level of engagement with various social media platforms. Their findings indicated that different age groups utilize social media differently. Accordingly, they observed that people over 45 prefer Facebook, whereas younger people (those aged 18 to 34 years) are more active on Instagram.

The impact of various factors (including user characteristics, posts, emotions, relationships, images, and backgrounds, among others) on the engagement rate must be investigated because assessing these influential factors in different networks can increase the engagement of users with advertising posts and thus increase the success rate of targeted advertising. The following section describes relevant methods.

2.2. Feature-Weight Learning Algorithm: Extreme Gradient Boosting. XGBoost [42] is an improved algorithm derived from the gradient boosting decision tree. It can efficiently construct boosted trees and execute parallel operations. These boosted trees are categorized into classification and regression trees. The central idea behind the algorithm is to optimize the objective function. While feature vectors compute the similarity of the history and forecasting days, gradient boosting builds boosted trees that can extract the feature scores intelligently, demonstrating to the training model the importance of each feature. Increased usage of a feature for making important decisions with boosted trees results in a higher score for that feature. The algorithm counts importance via “cover,” “frequency,” and “gain” [43]. The cover is defined as the relative value of observing a feature. Frequency, a simplified version of gain, is defined as the occurrence of a given feature in all constructed trees. Moreover, the gain is the most significant reference factor for determining the importance of a particular feature in the tree branches. “Gain” is used to set the feature importance in this study. For a single decision tree T, Breiman et al. [44] proposed the following as a score of importance for each predictor feature $X_i$:

$$W_i^2(T) = \sum_{t=1}^{I-1} r_i^2. \quad (1)$$

There are J − 1 internal nodes in the decision tree, which divide the region into two subregions over each node t using the prediction feature $X_i$. The selected feature is the one that provides the most significant estimated improvement $r_i^2$ in the squared error risk over a constant fit across the entire region. The squared importance of feature $X_i$ is equal to the sum of similar squared improvements at all the J − 1 nodes, for which this feature was selected as the splitting feature. The importance is calculated over the additive M trees as follows:

$$W_i^2(T) = \frac{1}{M} \sum_{m=1}^{M} r_i^2(T_m). \quad (2)$$

2.3. Fuzzy Clustering Algorithm. Lotfi Zadeh pioneered the fuzzy set theory to eliminate linguistic ambiguities and event uncertainties [45, 46]. In the traditional set theory, set boundaries are defined so that a given element is either definitely included or definitely excluded from the set. A similar clear distinction exists in classical logic, which states that a given proposition is either entirely accurate or incorrect. However, many propositions and complexes may not conform to such distinct and clear boundaries.

A fuzzy set results when a feature is not fully clear, that is, when the membership of all or some members belonging to a set is not completely clear. In hard clustering or classic strategies, such as K-means clustering, a sample or object
either definitely belongs to a set or not, with a respective membership degree of either 1 or 0. In the fuzzy or soft membership theory, an object may belong to a cluster with a membership degree between 0 and 1 [47]. Fuzzy clustering may refer to precise or fuzzy data analysis using fuzzy methods. However, this article used fuzzy techniques to analyze specific data. In practice, FCM-based fuzzy clustering algorithms are the most prevalent [48, 49]. These algorithms, such as K-means clustering, belong to a class of clustering algorithms with objective functions; they aim at minimizing the cost function shown as follows:

\[ J(U, V) = \sum_{j=1}^{C} \sum_{i=1}^{N} \left( \mu_{ij} \right)^{m} \left\| x_{i} - v_{j} \right\|^{2}. \]  

(3)

Here, \( V = \{ v_{1}, v_{2}, \ldots, v_{c} \} \) represents cluster center set, \( U = (\mu_{ij}) \) denotes a fuzzy partition matrix, \( m \in [1, \infty) \) is a weighting factor that indicates cluster fuzziness, and \( \mu_{ij} \in [0, 1] \) represents the degree of membership of data point \( j \) within the cluster. The cost function \( J(U, V) \) is minimized by calculating the cluster centers as follows:

\[ v_{j} = \frac{\sum_{i=1}^{N} (\mu_{ij})^{m} x_{i}}{\sum_{i=1}^{N} (\mu_{ij})^{m}}. \]  

(4)

Using the computed distance norms \( d_{ik} = \| x_{i} - v_{j} \| \), we update the fuzzy partitioning matrix, \( \mu \), as follows:

\[ \mu_{ij} = \frac{1}{\sum_{k=1}^{C} \left( d_{ij}/d_{ik} \right)^{2m-1}} \quad \mu_{ij} = 1 \text{ if } d_{ij} > 0 \left( x_{i} \neq v_{j} \right), \]  

where \( C \) represents the number of clusters in the equation above.

2.4. Convolutional Neural Network (CNN). The basis of a convolutional neural network (CNN) is a series of convolution and subsampling processes conducted across multiple layers [50]. Subsequently, one or several fully connected layers follow the CNN. All CNN operations pass through three successive layers, as depicted below.

2.4.1. Convolution Layer. The name CNN originates from the performed convolution operation. This process primarily contributes to the extraction of features from the input data. For instance, if the input is an image, the convolution operation extracts image features while maintaining the interpixel spatial relationship by learning the image features via small squares of input data (2-D filters). Convolution contributes to extracting the feature matrix when applied to text classification by maintaining a high-level word or phrase representation.

2.4.2. Pooling Layer. A good practice is to decrease the number of trainable parameters when the input size is too large. The feature dimension must be decreased with no loss of significant information. Pooling layers periodically appear between subsequent convolution layers. Pooling (also known as down-sampling or subsampling) decreases the spatial size of the feature maps but preserves the most significant information. There are different types of spatial pooling, including max, average, and sum. In max-pooling, a spatial neighborhood (e.g., a 22-window) is defined, and the largest element of the rectified feature map within that window is extracted. Instead of the greatest element, the sum or average of all elements within that window can be taken for sum and average pooling, respectively. This article considers the max-pooling strategy.

2.4.3. Fully Connected Layer. This layer is a classical multilayer perceptron with an output layer that employs a softmax activation function. The term “fully connected” indicates that every neuron in the preceding layer is connected to every neuron in the next layer. The convolution and pooling layers output high-level features from the input data. The fully connected layer uses these features to classify the input into different classes according to the training dataset. The majority of the features of convolution and pooling layers appear to be appropriate for classification.

2.5. LSTM-Based RNN for Prediction. Hochreiter et al. proposed LSTM in 1997 [51] as an efficient recurrent neural network (RNN) architecture, extensively utilized in numerous fields. Additionally, LSTM is a well-known model for time series prediction that can handle long-term dependency data effectively. RNNs were designed to work in nonlinear time-varying problems [52]. The internal connections of an RNN allow signals to travel back and forth, making RNNs suitable for time series prediction. RNNs can mine rules from time sequences for predicting data that have not yet occurred. The features result from the feedback connections enabling weight updates based on the residual at each forward step (Figure 1):

(i) \( x_{t} \): is taken as the input to the network at time step \( t \)
(ii) \( h_{t} \): represents a hidden state at time \( t \) and acts as “memory” of the network
(iii) \( W \): hidden-to-hidden recurrent connections parameterized by a weight matrix \( W \)

RNN has demonstrated suitability for the specified problem [53]. However, RNNs are likely heavily impacted by
gradient vanishing, which can increase indefinitely and lead to network failure. Consequently, simple RNNs may not be optimal for predicting problems with long-term dependencies.

LSTM was initially designed to deal with the problem of vanishing gradients in standard RNN when handling long-term dependencies. This section concludes with the long-term, short-term memory (LSTM) neural network. The LSTM model augments the RNN neurons with input, output, and forget gates. This structure can effectively circumvent the problem of vanishing gradients [54].

This capability makes LSTM a suitable architecture for problems that contain long-term dependencies. The main novelty of LSTM lies in the memory cell, which essentially acts as a state information collector. LSTM uses the forget gate to determine which information to discard from the cell state, as shown in Figure 2. The forget gate \(f_t\)'s activation is calculated using a sigmoid function:

\[
f_t = \sigma(W_f[h_{t-1}, x_t] + b_f).
\]  

The subsequent step involves determining which new information to store in the cell state. Initially, a sigmoid layer called the “input gate layer” decides what information must be updated. Subsequently, a tanh layer produces a vector \(c_t\) containing new candidate values to be updated next:

\[
i_t = \sigma(W_i[h_{t-1}, x_t] + b_i),
\]

\[
c_t = \tanh(W_c[h_{t-1}, x_t] + b_c).
\]  

Afterward, the previous cell state \(c_{t-1}\) is updated to the new cell state \(c_t\), \(c_{t-1}\) is multiplied by \(f_t\) to discard the information from the old cell. Next, \(i_t * c_t\) is added. The resulting candidates are scaled according to the amount of information that must be updated for each state value:

\[
c_t = f_t * c_{t-1} + i_t * c_t.
\]

Finally, the output must be decided. There are two parts to this: first, a sigmoid layer is run as the output gate for filtering the cell state; next, the cell state is placed through \(\tanh(.)\) and is multiplied by the output \(o_t\) for computing the desired information:

\[
o_t = \sigma(W_o[h_{t-1}, x_t] + b_o),
\]

\[
c_t = o_t * \tanh(c_t).
\]  

In Equations (10)–(15), \(W_f, W_i, W_c,\) and \(W_o\) denote the suitable weight matrices. Moreover, the vectors \(b_f, b_i, b_c,\) and \(b_o\) represent the associated bias vectors. Figure 3 depicts the model framework.

3. Proposed Method

We propose a hybrid convolutional network model for learning and predicting user engagement rates based on the FW-FCM and CNN-LSTM algorithms. We extract the effective attributes using FCM XGBoost methods. Selected data are clustered based on the attributes' weight, and our model learns similar features by employing a convolutional neural network. The output of this algorithm is the prediction of the level of the user engagement rate in a promotion. The overall process of our approach is shown in Figure 4.

3.1. Data Collection and Preprocessing. This study aims at providing an analytical model independent of the dataset. Consequently, we evaluate our method using three datasets from different domains. The following is a detailed description of each dataset’s selected features.

3.1.1. Taobao Dataset. Taobao is an advertising dataset provided by Alibaba, containing eight days of ad click-through data (26 million records) randomly sampled from 1,140,000 users (Table 1). Following the original data split, the first seven days (i.e., 20170506–20170512) of samples are used for training, while the last day’s samples (20170513) are used for testing [55].

3.1.2. Instagram Dataset. The second dataset originated from a Python-written crawler that extracted it from Instagram. The dataset was compiled in 2019 by crawling 6,605 Instagram posts containing the hashtag #mcdonalds and mentioning McDonald’s. The hashtags are represented by nodes, and their relationship is represented by a link [56] (Table 2).

3.1.3. IMDB Dataset. The Large Movie Review Dataset (commonly referred to as the IMDB dataset) contains 25,000 top-rated movie reviews for training and an equal number for testing. The issue is determining whether a particular movie review is positive or negative. The data were collected by Stanford researchers and published in 2011 in a paper in which the ratio of training data to testing data was 70:30 [57] (Table 3).

4. Implementation

The proposed FW-CNN-LSTM model’s entire procedure is outlined below and illustrated in Figure 5:

1. Selection of similar features: Calculate the feature-weights using the XGBoost method, and combine them with the FCM algorithm to identify similar clusters
2. Construction of the feature matrix: Each feature within the text list is embedded in a 128-element vector trained via the backpropagation procedure
3. Decomposition of data: Take a similar feature load as input data and decompose the input data into several intrinsic mode functions
4. Prediction: Use separate CNN-LSTM convolution networks to compute the disparity between the actual and predicted engagement labels
4.1. **Feature Importance.** The factor significance of the 10 top features in the three databases is shown in Figure 6. Based on the data analysis presented in the previous section, multiple features are fed into the XGBoost algorithm to calculate feature importance. Hashtag-related features achieve the highest scores in the Instagram database. These features

![Diagram](image-url)

**Figure 2:** The architecture of LSTM memory block.

![Diagram](image-url)

**Figure 3:** Structure of the proposed approach.

![Diagram](image-url)

**Figure 4:** The flowchart of the FW-CNN-LSTM model.
### Table 1: The dataset’s selected features (Taobao).

| Advertising features                                                                 |
|-----------------------------------------------------------------------------|
| N1: Timestamp;                                                             |
| N2: Category ID;                                                            |
| N3: Campaign ID;                                                            |
| N4: Brand ID;                                                               |
| N5: Advertiser ID;                                                          |
| N6: Item price;                                                             |

| User features                                                               |
|-----------------------------------------------------------------------------|
| N7: Micro group ID;                                                         |
| N8: CMS group ID;                                                           |
| N9: Gender; 1 for male, 2 for female;                                       |
| N10: Consumption grade, 1: low, 2: mid, 3: high;                            |
| N11: Shopping depth, 1: shallow user, 2: moderate user, 3: deep user;       |
| N12: City level;                                                            |

| Behavioral trait | Click-through rate (CTR) |
|------------------|--------------------------|
| N13: Types of behavior, including cart/fav/buy;                            |
| N14: 1 for click, 0 for not click;                                         |

### Table 2: The dataset’s selected features (Instagram).

| Post features                                                                 |
|------------------------------------------------------------------------------|
| F1: Type of media, carousel, or image                                         |
| F2: Post date                                                                |
| F3: Post time                                                                |
| F4: Location tag                                                             |
| F5: User count tag                                                           |

| Image features                                                               |
|------------------------------------------------------------------------------|
| F6: Image size                                                               |
| F7: Image quality                                                            |
| F8: Text on the image                                                        |
| F9: Length of text on the image                                              |

| Caption features                                                             |
|------------------------------------------------------------------------------|
| F10: Caption text                                                            |
| F11: Caption text length                                                     |
| F12: Number of keywords in the caption                                       |
| F13: Total number of characters in the caption keywords                       |

| Hashtag features                                                             |
|------------------------------------------------------------------------------|
| F14: Number of hashtags                                                      |
| F15: Hashtag length                                                          |
| F16: Hashtag text                                                             |

| Engagement features                                                          |
|------------------------------------------------------------------------------|
| F17: Like count                                                              |
| F18: Comment count                                                           |
| F19: Engagement rate, equal to (lik + com)/flr                               |
| F20: Engagement rate classification according to the Likert scale A: Very good, B: good, C: medium, D: low, E: very low |

| Sentiment feature                                                            |
|------------------------------------------------------------------------------|
| F21: Image sentiment analysis (object detection)                             |
| F22: Image text sentiment analysis                                           |
| F23: Caption text sentiment analysis                                         |
| F24: Hashtag text sentiment analysis                                         |

### Table 3: The dataset’s selected features (IMDB).

| Title features                                                               |
|------------------------------------------------------------------------------|
| M1: Text title                                                              |
| M2: A number to identify rows uniquely for a specific title ID               |
| M3: The more well-known title/the title on the promotional materials at the time of release |
| M4: Title sentiment analysis                                                 |
| M5: Title language                                                           |
| M6: Enumerated attribute set for the alternative title. One or more of the following: “DVD,” “alternative,” “TV,” “festival,” “working,” “video,” “imdbDisplay,” and “original.” New values can be added prospectively without warning. |
| M7: Length of the title text                                                 |
| M8: Number of keywords in the title                                          |

| Movie features                                                               |
|------------------------------------------------------------------------------|
| M9: Video format/type (e.g., short, movie, short, TV series, tv episode, and video) |
| M10: This feature represents the title release year. For a TV series, it is the start year of the series |
| M11: Primary runtime (in minutes)                                            |
| M12: Genres related to the title                                             |
include the number of hashtags, the hashtag’s length, and the hashtag’s text. In the IMDB database, movie title and actors who achieve the highest scores, and in the Taobao dataset, timestamp and brand ID achieve the highest scores. In addition, the supplement features are significant for prediction. This outcome is consistent with the data analysis results. The significant values of every feature have now been extracted. These will be employed as a priori knowledge for the next clustering algorithm.

4.2. FCM Clustering Based on Feature Weight. This section improves FCM clustering by calculating the original cluster centers and employing a new distance computation method.
The core idea underlying the selection of SDs is to determine the attribute weights using the XGBoost algorithm and compute the distance between the selected engagement rate, which depends on measuring various attributes with varying weights. This process comprises the following steps:

1. The predicting engagement rate is chosen as the initial center $c_0$, which aims at minimizing the cost function below.

2. The subsequent center $c_j$ is selected, where $c_i$ denotes the furthest point from the cluster centers chosen previously, that is, $\{c_0, c_1, \ldots, c_j\}$. Steps 1 and 2 are repeated until all the K centers are found.

3. The feature-weights are calculated via the XGBoost method. Next, a weight is attributed to every feature, giving them different importance levels. Assuming $w_p$ to be the weight related to the feature $p$, we present the norm as follows:

(a) We assign each data point to the closest cluster.
(b) We update the clusters by calculating the cluster centroid again. The algorithm repeatedly executes steps (1) and (2) until convergence occurs.

4.3. Setting the Convolutional Neural Network (CNN)

4.3.1. Construction of the Feature Matrix. Each feature in the text list is embedded in a 128-element vector trained through the backpropagation process. Figure 7 depicts a sample feature matrix created from the representation embedding model. The index values for each feature are displayed as an example in Figure 6. These values can vary with datasets. The values in the matrix are weights assigned randomly to the embedding layers adjusted via the backpropagation procedure.

4.3.2. The Convolution Layer. In the convolutional layer, the input feature matrix is traversed by eight filters with a size of $7 \times 1$ and a stride of 1 to obtain the required features. We utilize multiple filters to extract different feature types. For instance, if a matrix with a size of $26 \times 130$ is traversed using a filter with dimensions of $7 \times 1$, the convolution process will deliver $8 \times 26 \times 130$ features matrix. The matrix includes every local hidden feature, as shown in Figure 8. The input of the last layer is the output from the last layer in LSTM.

4.3.3. The Max-Pooling Layer. After a feature matrix with a size of $m \times n$ is obtained from the convolutional layer, max-pooling is applied using a filter with dimensions of $2 \times 1$. In the max-pooling process, the highest feature value is selected at each filter position during the traverse. The resulting feature matrix has dimensions of $m/2 \times n$. The process of max-pooling is performed independently for each convolution filter. The max-pooling layer’s structure is schematically depicted in Figure 8.

4.3.4. The Sigmoid Layer. The output feature vectors from the LSTM layer are transferred to a fully connected sigmoid layer to determine each category's probability distribution. This is mathematically expressed as follows:

$$D_{\text{sig}}(V_j) = \frac{e^{z_j}}{1 + e^{z_j}}.$$  

Here, $D_{\text{sig}}(V_j)$ represents the probability distribution of category $j$ and $z_j$ denotes the output related to category $j$. We use the sigmoid activation function to normalize the confidence score of the classifier between 0 and 1. After the probability distribution is obtained from the sigmoid layer, we apply binary cross-entropy as a loss function to calculate the disparity between the actual and predicted engagement labels [58]:

$$L = -\sum_{j=1}^{k} R(V_j) \log D_{\text{sig}}(V_j).$$

Figure 6: Feature matrix.
Here, $k$ represents the number of categories. Moreover, $R(V_i)$ denotes the actual sentiment related to the text. This parameter can adopt discrete values belonging to the set $T = \{A, B, C, D, E\}$, with $T$ being the label for the engagement rate. $R(V_i)$ resembles the likelihood function, which seeks to minimize the difference between the probability distribution within the training set and that predicted by the model for the testing dataset.

4.4. Setting the Long Short-Term Memory (LSTM) Network. The LSTM layer receives the output of the max-pooling layer for sequential left-to-right analysis of the created feature vectors. Because the significant local features were extracted at the max-pooling layer output, the LSTM network can examine long-term dependencies for global features. The LSTM layer output is flattened in order to reduce the
4.4.1. Quantifying the Number of Hidden Layers in the LSTM. Attention-based LSTM has varying output sequences and learning effects depending on the number of hidden layers. This section seeks to determine the optimal number of hidden layers for classifier performance. This section includes three experiments. The experiment variable equals the number of hidden layers. We compute the time consumption and precision of the model by keeping the other variables unchanged and setting the hidden layer number to 1, 2, and 3. The number of hidden layer neurons in all three experiments is 128. The results are displayed in Table 4.

As shown in the table, there is a strong correlation between the number of hidden layers and time consumption. The more hidden layers, the more time required to train identical data. The reason is that the number of neurons that must be trained grows exponentially as the number of hidden layers increases. An increase in parameters will result in increased time and resource consumption. However, there is no strong correlation between precision and the number of layers; increasing the number of layers will not improve the model’s precision. Consequently, more hidden layers are not necessarily better. As shown in the table, there is a moderate time consumption for a hidden layer number of 2. Most importantly, the two-layer neural network has the highest precision. Therefore, the LSTM models listed below employ two hidden layers.

4.4.2. Quantifying the Number of Neurons in the LSTM. The previous section’s experiment determined the number of model layers by adjusting only the number of hidden layers. Nevertheless, for convenience, the experiment assumes the number of neurons to be 128 in each layer. In this section, after determining the number of hidden layers in the model, we identify the number of neurons in each layer through experiments to improve the attention-based LSTM formulation. Since there are two hidden layers in LSTM, the number of neurons must be tested in the network with two layers. We designed three sets of tests with neuron numbers of (128, 128), (256, 128), and (128, 64). The time consumption and accuracy of each set of tests were computed, and the results are presented in Table 5.

As shown in the table, the training time is only slightly affected by the number of neurons. Consequently, the current experiment utilizes the group of parameters with the highest precision, corresponding to 128 hidden layers.

5. The Performance Evaluation Parameters

We randomly split the dataset into training (70%), validation (20%), and test (10%) sets. The validation set was used for tuning hyper-parameters, and the final performance comparison was conducted on the test set. Figures 9 demonstrates the efficiency of the proposed model in terms of accuracy and the number of epochs. The learning of the proposed network was overfitted on the Taobao dataset (with more than 100 epochs). Also, the presented method has a good fit in 150 epochs on the IMDB dataset and 100 epochs on the Instagram dataset. In this study, we have employed the dropout (0.3) method after the convolutional layers to avoid overfitting. Then, to determine the proper optimization algorithm for our proposed model, the standard methods of Adam, RMSProp, and SGD were compared based on their cost function, and the Adam algorithm was selected with a learning rate of 0.0001, which obtained better results.

All the experiments in this article used the Python programming language, and PyCharm was utilized as the development environment. The FW-CNN-LSTM model was compared to the other machine learning models, such as CNN, logistic regression, Naïve Bayes, and SVM, for validation purposes. The proposed algorithm was evaluated regarding the F-measure, recall, precision, and accuracy obtained from the confusion matrix. In the following, the criteria set in line with the research objective, identifying and suggesting suitable users for targeted advertisements, are introduced.

The precision criterion indicates how accurately the algorithm predicts user engagement rates. The following equation shows how this criterion is calculated:

\[
\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}
\]

The precision criterion has several weaknesses. For instance, it does not differentiate the prediction error of the engagement rate of included users from that of excluded users. The precision and recall criteria were employed to solve this problem. The main focus of the precision criterion is on positive classifier outputs and the extent to which the engagement rate is correctly predicted in positive responses. The following equation shows how this criterion is calculated:

\[
\text{Recall} = \frac{\text{TP}\_\text{in}}{\text{TP}\_\text{in} + \text{FP}\_\text{in}}
\]

The recall criterion is used to verify the algorithm’s coverage in introducing users to be selected for targeted advertisements. In this criterion, the main focus is on the prediction error of included users’ engagement rates. The recall criterion is more important in issues related to sending
targeted advertisements. The following shows how this criterion is calculated:

\[
\text{Recall} = \frac{TP}{TP + FN}
\]  \hspace{1cm} (15)

The values of recall and precision can be significantly different, in which case the best algorithm is selected using the following equation from the average recall and precision values under the criterion \(F\)-measure:

\[
F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]  \hspace{1cm} (16)

(i) True positive (TP): TP represents the reviews initially labeled positive and predicted by the classifier as positive. (ii) False positive (FP): FP represents the reviews initially labeled negative but predicted by the classifier as positive. (iii) True negative (TN): TN represents the reviews initially labeled negative and predicted by the classifier as negative. (iv) False negative (FN): FN represents the reviews initially labeled positive but predicted by the classifier as negative.

6. Analysis and Discussion

To evaluate the scalability of the proposed method for large datasets, the presented method was assessed based on various ratios of training data (0.1, 0.2, 0.3, \ldots, 1) and their corresponding training time. Figure 10 illustrates this process. As can be seen, increasing the ratio of training data from 0.1 to 1 increased the training time from 97 seconds to 675 seconds. However, the values for actual training time in various data ratios are relatively linear, which confirms the scalability of the proposed model for large datasets.

We demonstrate the efficacy of the proposed FW-CNN-LSTM model by comparing it to classical machine learning techniques widely used today, such as CNN, Naïve Bayes, logistic regression, and SVM. The comparison indicators are the \(F\)-measure, precision, recall, and accuracy. Table 6 presents a comparative analysis of the Taobao dataset. As shown in the recall criterion, the FW-CNN-LSTM algorithm could recognize over 91% of the users suitable for targeted advertisements, meaning that less than 18,400 of the customers were not covered by the proposed method. Regarding the precision criterion, the proposed method (FW-CNN-LST) successfully selected 94% of the users suitable for targeted advertisements, which is acceptable compared to other methods. The Naive Bayes algorithm also performed acceptably in the precision criterion. The \(F\)-measure indicates that the FW-CNN-LSTM algorithm outperformed the other examined methods. The FW-CNN-LSTM algorithm successfully
predicted the customer engagement rate (91%) in the accuracy criterion, displaying its superiority.

In the IDMP dataset, the Naive Bayes algorithm displayed the best performance (97%) in the recall criterion. Thus, this algorithm could optimally include the users suitable for targeted advertisements. However, the algorithm displayed 84% precision by wrongly sending advertisements to 870 users. The FW-CNN-LSTM algorithm outperformed the other algorithms by displaying 96% precision. The algorithm has assigned advertisements to users more accurately. Nevertheless, the algorithm covers a smaller range of users (about 87%). Given the significant difference in the performance regarding the precision and recall criteria, the proposed methods can be compared more accurately using the $F$-measure criterion. The $F$-measure indicates that the FW-CNN-LSTM algorithm outperformed the other examined methods. In the accuracy criteria, the FW-CNN-LSTM method successfully predicted the user engagement rates by 91%, outperforming the other algorithms (Table 7).

In the Instagram dataset, the FW-CNN-LSTM method could optimally (93%) cover the selected users for targeted advertisements in the recall criteria, failing to recognize only 79 included users. The CNN algorithm displayed the best performance (92%) in the precision criterion. The FW-

![Figure 10: Training time of the proposed method on different data proportions.](image)

| Models        | Confusion matrix          | Evaluation parameter |
|---------------|---------------------------|----------------------|
|               | Predicted positive | Predicted negative | Precision | Recall | $F$-measure | Accuracy |
| CNN           | Actually positive | 170530 | 28400 | 0/8944 | 0/8572 | 0/8754 | 0/8581 |
|               | Actually negative    | 20125 | 122945 |  |  |  |  |
| Logistic regression | Actually positive | 191025 | 40600 | 0/8966 | 0/8247 | 0/8591 | 0/8288 |
|               | Actually negative    | 22035 | 112160 |  |  |  |  |
| Naive Bayes   | Actually positive | 181252 | 28173 | 0/9471 | 0/8655 | 0/9044 | 0/8880 |
|               | Actually negative    | 10125 | 122450 |  |  |  |  |
| SVM           | Actually positive | 185325 | 18400 | 0/9213 | 0/9097 | 0/9155 | 0/8999 |
|               | Actually negative    | 15825 | 122450 |  |  |  |  |
| FW-CNN-LSTM   | Actually positive | 191025 | 18400 | 0/9497 | 0/9121 | 0/9305 | 0/9166 |
|               | Actually negative    | 10125 | 122450 |  |  |  |  |

| Models        | Confusion matrix          | Evaluation parameter |
|---------------|---------------------------|----------------------|
|               | Predicted positive | Predicted negative | Precision | Recall | $F$-measure | Accuracy |
| CNN           | Actually positive | 4076 | 741 | 0/9488 | 0/8462 | 0/8945 | 0/8719 |
|               | Actually negative    | 220 | 2463 |  |  |  |  |
| Logistic regression | Actually positive | 5012 | 750 | 0/9766 | 0/8698 | 0/9201 | 0/9089 |
|               | Actually negative    | 120 | 3665 |  |  |  |  |
| Naive Bayes   | Actually positive | 4756 | 122 | 0/8454 | 0/9750 | 0/9056 | 0/8833 |
|               | Actually negative    | 870 | 2752 |  |  |  |  |
| SVM           | Actually positive | 4256 | 750 | 0/9119 | 0/8502 | 0/8800 | 0/8563 |
|               | Actually negative    | 411 | 2665 |  |  |  |  |
| FW-CNN-LSTM   | Actually positive | 3865 | 550 | 0/9699 | 0/8754 | 0/9202 | 0/9107 |
|               | Actually negative    | 120 | 2965 |  |  |  |  |
CNN-LSTM method displayed 91% precision by sending 108 advertisements to excluded users. It can be stated that regarding the accuracy criterion, the proposed method outperformed other methods by successfully predicting the user engagement rate by 90% (Table 8).

The overall performance of the proposed algorithm was assessed and ranked using Friedman's test. Friedman's test computes the adjusted chi-square ($\chi^2$) between the proposed algorithm and the other algorithms [59]. The larger the adjusted $\chi^2$, the more different the ranks. This study considered a significance level of 0.05. If the $p$-value is smaller than 0.05, the null hypothesis (the mean ranks of the variables are not significantly different) is rejected. Consequently, the variables have different ranks. Tables 9–11 present the Friedman test’s ranking results on all datasets.

According to Friedman’s test, the FW-CNN-LSTM algorithm with the highest rank on datasets outperforms in the evaluation criteria compared to the other classification approaches. Logistic regression in the IMDB dataset, the CNN algorithm in the Instagram dataset, and the SVM algorithm in the Taobao dataset are of the ranks presented in Tables 9–11.

Figure 11 displays the performance of the presented algorithm on the top-$k$ users with a high engagement rate. The figures indicate the superior performance of the proposed algorithm compared with the other algorithms. The result showed that an increase in the number of predicting level of engagement will decrease the accuracy criterion. Accordingly, it can be concluded that sending advertisements for the first $k$ primary users can be significantly

### Table 8: Evaluation parameters for the Instagram dataset.

| Models          | Confusion matrix | Evaluation parameter |
|-----------------|------------------|----------------------|
|                 | Predicted positive | Predicted negative | Precision | Recall | F-measure | Accuracy |
| CNN             | Actually positive | 1258                | 142       | 473    | 0/9209     | 0/8986 | 0/9096 | 0/8738 |
|                 | Actually negative | 108                 |           |        |           |        |        |        |
| Logistic regression | Actually positive | 987                 | 80        | 713    | 0/8308     | 0/9250 | 0/8753 | 0/8582 |
|                 | Actually negative | 201                 |           |        |           |        |        |        |
| Naive Bayes     | Actually positive | 995                 | 132       | 736    | 0/8940     | 0/8829 | 0/8883 | 0/8738 |
|                 | Actually negative | 118                 |           |        |           |        |        |        |
| SVM             | Actually positive | 1058                | 342       | 473    | 0/9074     | 0/7728 | 0/8246 | 0/7728 |
|                 | Actually negative | 108                 |           |        |           |        |        |        |
| FW-CNN-LSTM     | Actually positive | 1221                | 79        | 573    | 0/9187     | 0/9392 | 0/9288 | 0/9056 |
|                 | Actually negative | 108                 |           |        |           |        |        |        |

### Table 9: Average ranking of Friedman’s test for the comparing algorithms of the Taobao dataset.

| Test results | CNN | Logistic regression | Naive Bayes | SVM | FW-CNN-LSTM |
|--------------|-----|---------------------|-------------|-----|-------------|
| Average ranking | 1.75 | 1.25 | 3.25 | 3.75 | 5 |
| Ranking | 2 | 1 | 3 | 4 | 5 |

Chi-square = 14.800, $p$ value = 0.0051, and $\alpha$ = 0.05.

### Table 10: Average ranking of Friedman’s test for the comparing algorithms of the IMDB dataset.

| Test results | CNN | Logistic regression | Naive Bayes | SVM | FW-CNN-LSTM |
|--------------|-----|---------------------|-------------|-----|-------------|
| Average ranking | 2 | 4 | 3 | 1.50 | 4.50 |
| Ranking | 2 | 4 | 3 | 1 | 5 |

Chi-square = 10.400, $p$ value = 0.0342, and $\alpha$ = 0.05.

### Table 11: Average ranking of Friedman’s test for the comparing algorithms of the Instagram dataset.

| Test results | CNN | Logistic regression | Naive Bayes | SVM | FW-CNN-LSTM |
|--------------|-----|---------------------|-------------|-----|-------------|
| Average ranking | 3.88 | 2.25 | 2.63 | 1.50 | 4.75 |
| Ranking | 4 | 2 | 3 | 1 | 5 |

Chi-square = 10.987, $p$ value = 0.0267, and $\alpha$ = 0.05.
effective, and the users will be more likely to interact with the advertisements.

7. Conclusion and Future Works

This article proposes a hybrid convolutional method, termed FW-CNN-LSTM, to predict the user engagement rate. The algorithm selects SDs by first defining the attribute weights with XGBoost and then estimating the distance between the selected engagement rate and the one that calculates various attributes with varying weights. The FCM algorithm then combines the feature-weight and XGBoost distance measures into a single cluster for further projection. Subsequently, the selected comparable feature-based data are utilized as an input into a hybrid convolutional CNN-LSTM network to predict the engagement rate. In addition, we develop a model capable of producing accurate results regardless of the dataset or domain. As a result, our algorithm is evaluated on three distinct datasets. The IMDB, Taobao, and Instagram datasets are associated with the film, advertising, and food industries, respectively. Accuracy, recall, F-measure, and precision are used to analyze the results of basic methods, such as SVM, Logistic regression, Naive Bayes, and CNN. The presented algorithm achieved a 0/9056 accuracy level on the Instagram dataset, a 0/9107 accuracy level on the IMDB dataset, and a 0/9166 accuracy level on the Taobao dataset. Consequently, the results demonstrated that our algorithm outperforms others. According to the findings, hashtag, brand ID, movie title, and actors who achieve the highest scores, and the values for actual training time in various data ratios are relatively linear, which confirms the scalability of the proposed model for large datasets. Our method analyzes the user engagement rate with advertising content, resulting in more targeted advertisements and less spam on social media platforms. It is suggested that relational features and post links be utilized to enhance our context-based strategy and predict and analyze the user engagement rate with advertisement content.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.
References

[1] M. A. Shareef, B. Mukerji, Y. K. Dwivedi, N. P. Rana, and R. Islam, “Social media marketing: comparative effect of advertisement sources,” Journal of Retailing and Consumer Services, vol. 46, pp. 58–69, 2019.

[2] G. Wang, L. Zhou, J. Li, D. Ren, and J. Zhang, “An efficient method of content-targeted online video advertising,” Journal of Visual Communication and Image Representation, vol. 50, pp. 40–48, 2018.

[3] Y. Li, S. Wang, Y. Ma, Q. Pan, and E. Cambria, “Popularity prediction on vacation rental websites,” Neurocomputing, vol. 412, pp. 372–380, 2020.

[4] Z. Zhang, T. Chen, Z. Zhou, J. Li, and J. Luo, “How to become Instagram famous: post popularity prediction with dual-attention,” in Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, pp. 2383–2392, IEEE, December 2018.

[5] K. R. Purba, D. Asirvatham, and R. K. Murugesan, “Classification of Instagram fake users using supervised machine learning algorithms,” International Journal of Electrical and Computer Engineering, vol. 10, no. 3, pp. 2763, 2020.

[6] K. Almgren, J. Lee, and M. Kim, “Predicting the future popularity of images on social networks,” in Proceedings of the 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, vol. 1,–6, Union NJ, USA, August 2016.

[7] K. Cakmak, I. Cikrickenoglu, O. Demiralp et al., “The Causal Determinants of Popularity in Instagram,” Technical Report, 2017.

[8] S. De, A. Maity, V. Goel, S. Shitole, and A. Bhattacharya, “Predicting the popularity of Instagram posts for a lifestyle magazine using deep learning,” in Proceedings of the 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), pp. 174–177, IEEE, Mumbai, April 2017.

[9] M. Mazloom, R. Rievteld, S. Rudinac, M. Worring, and W Van Dolen, “Multimodal popularity prediction of brand-related social media posts,” in Proceedings of the 24th ACM International Conference on Multimedia, pp. 197–201, New York, USA, October 2016.

[10] K. Seo, S. Dodson, N. M. Harandi, N. Roberson, S. Fels, and I. Roll, “Active learning with online video: the impact of learning context on engagement,” Computers & Education, vol. 165, Article ID 104132, 2021.

[11] K. Ding, K. Ma, and S. Wang, “Intrinsic image popularity assessment,” in Proceedings of the 27th ACM International Conference on Multimedia, pp. 1979–1987, Nice France, October 2019.

[12] X. Mao and F. Xu, “Research on the popularity prediction of multimedia network information based on fast K neighbor algorithm,” Journal of Electrical and Computer Engineering, vol. 2022, pp. 1–11, Article ID 8745492, 2022.

[13] A. Deza and D. Parikh, “Understanding image virality,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1818–1826, Boston, MA, USA, June 2015.

[14] J. Berger and K. L. Milkman, “What makes online content viral?” Journal of Marketing Research, vol. 49, no. 2, pp. 192–205, 2012.

[15] Y. Bae and H. Lee, “Sentiment analysis of twitter audiences: measuring the positive or negative influence of popular twitterers,” Journal of the American Society for Information Science and Technology, vol. 63, no. 12, pp. 2521–2535, 2012.

[16] W. Tafesse and B. P. Wood, “Followers’ engagement with Instagram influencers: the role of influencers’ content and engagement strategy,” Journal of Retailing and Consumer Services, vol. 58, Article ID 102303, 2021.

[17] S. Guo, R. Chen, H. Li, T. Zhang, and Y. Liu, “Identify severity bug report with distribution imbalance by CR-SMOTE and ELM,” International Journal of Software Engineering and Knowledge Engineering, vol. 29, no. 02, pp. 139–175, 2019.

[18] H. Li, X. Yang, Y. Li, L. Y. Hao, and T. L. Zhang, “Evolutionary extreme learning machine with sparse cost matrix for imbalanced learning,” ISA Transactions, vol. 100, pp. 198–209, 2020.

[19] H. Li, Y. Qu, S. Guo, G. Gao, R. Chen, and G. Chen, “Surprise Bug Report Prediction Utilizing Optimized Integration with Imbalanced Learning Strategy,” Complexity, vol. 2020, Article ID 8509821, 14 pages, 2020.

[20] S. Delgado, C. Higuera, J. Calle-Espinosa, F. Morán, and F. Montero, “A SOM prototype-based cluster analysis methodology,” Expert Systems with Applications, vol. 88, pp. 14–28, 2017.

[21] J. C. Besdel, Pattern Recognition with Fuzzy Objective Function Algorithms, Springer Science & Business Media, Berlin, Germany, 2013.

[22] D. Q. Zhang and S. C. Chen, “Clustering incomplete data using kernel-based fuzzy c-means algorithm,” Neural Processing Letters, vol. 18, no. 3, pp. 155–162, 2003.

[23] Y. Wang, K. Ma, L. Garcia-Hernandez et al., “A CLSTM-TMN for marketing intention detection,” Engineering Applications of Artificial Intelligence, vol. 91, Article ID 103595, 2020.

[24] M. E. Chennafi, H. Bedlauoi, A. Dahou, and M. A. A. Al-qaqessAl-qaqess, “Arabic aspect-based sentiment classification using Seq2Seq dialect normalization and transformers,” Knowledge, vol. 2, no. 3, pp. 388–401, 2022.

[25] W. Aldjanabi, A. Dahou, M. A. Al-qaqess, M. A. Elaziz, A. M. Helmi, and R. Damaševičius, “Arabic offensive and hate speech detection using a cross-corpora multi-task learning model,” in Informatics, vol. 8, no. 4, p. 69, MDPI, 2021, October.

[26] H. Fan, W. Du, A. Dahou et al., “Social media toxicity classification using deep learning: real-world application UK brexit,” Electronics, vol. 10, no. 11, p. 1332, 2021.

[27] T. Ghanbarpour, E. Sahabeh, and A. Gustafsson, “Consumer response to online behavioral advertising in a social media context: the role of perceived ad complicity,” Psychology and Marketing, vol. 39, no. 10, pp. 1853–1870, 2022.

[28] X. Zhu, H. Tao, Z. Wu, J. Cao, K. Kalish, and J. Kayne, Fraud Prevention in Online Digital Advertising, Springer International Publishing, Berlin, Germany, 2017.

[29] J. A. Choi and K. Lim, “Identifying machine learning techniques for classification of target advertising,” ICT Express, vol. 6, no. 3, pp. 175–180, 2020.

[30] D. Li, B. Hu, Q. Chen et al., “Attentive capsule network for click-through rate and conversion rate prediction in online advertising,” Knowledge-Based Systems, vol. 211, Article ID 103595, 2020.

[31] H. Liu, X. Zhu, K. Kalish, and J. Kayne, “Ultr-ctr: fast page grouping using url truncation for real-time click through rate estimation,” in Proceedings of the 2017 IEEE International Conference on Information Reuse and Integration (IRI), pp. 444–451, IEEE, San Diego, CA, USA, August 2017.

[32] C. Li, Y. Lu, Q. Mei, D. Wang, and S. Pandey, “Click-through prediction for advertising in twitter timeline,” in Proceedings of the 21th ACM SIGKDD International Conference on
Knowledge Discovery and Data Mining, pp. 1959–1968, Sydney NSW Australia, August 2015.

[33] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction, 2017.

[34] T. Hong, J. A. Choi, K. Lim, and P. Kim, “Enhancing personalized ads using interest category classification of SNS users based on deep neural networks,” Sensors, vol. 21, no. 1, p. 199, 2020.

[35] J. M. Yap, Library Promotion and User Engagement in Pandemic Times: The Case of The UoN Library, Kazakhstan, 2020.

[36] J. Luke and S. Suharji, “Data mining of automatically promotion tweet for products and services using Naive Bayes algorithm to increase twitter engagement followers atPT. Bobobobo,” Procedia Computer Science, vol. 59, pp. 254–261, 2015.

[37] M. Bonilla-Quijada, J. Tugores-Ques, and J. L. Olmo Arriaga, “Promotion of urban tourism: insights into user engagement on social media,” Information Technology & Tourism, vol. 23, no. 4, pp. 611–632, 2021.

[38] X. Zheng, C. M. Cheung, M. K. Lee, and L. Liang, Building Brand Loyalty through User Engagement in Online Brand Communities in Social Networking Sites, Information Technology & People, 2015.

[39] W. B. Kim, J. E. Marinas, and R. B. Vender, “Public engagement with dermatology contents on Facebook,” Journal of Cutaneous Medicine and Surgery, vol. 19, no. 3, pp. 304–308, 2015.

[40] R. Stefko, R. Back, and I. Fedorko, “Facebook content analysis of banks operating on Slovak market,” Polish Journal of Management Studies, vol. 10, no. 1, pp. 145–152, 2014.

[41] C. Gasparomi, Event Marketing on Social media: The Importance of User Engagement on Social media for Kaustinen Folk Music Festival, Tampere University of Applied Sciences, 2020.

[42] R. Shi, X. Xu, J. Li, and Y. Li, Prediction and Analysis of Train Arrival Delay Based on XGBoost and Bayesian Optimization, Applied Soft Computing, 2021.

[43] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer Series in Statistics, Berlin, Germany, 2001.

[44] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, Classification and Regression Trees, CRC Press, Florida, USA, 1984.

[45] L. A. Zadeh, “Fuzzy sets, Advances in Fuzzy Systems-Applications and Theory,” in Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A Zadeh, pp. 394–432, 1996.

[46] N. Gerami SereshGerami Seresht, R. Lourenzutti, and A. R. Fayek, “A fuzzy clustering algorithm for developing predictive models in construction applications,” Applied Soft Computing, vol. 96, Article ID 106679, 2020.

[47] S. M. Chen and Y. C. Chang, “Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques,” Information Sciences, vol. 180, no. 24, pp. 4772–4783, 2010.

[48] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: the fuzzy c-means clustering algorithm,” Computers & Geosciences, vol. 10, no. 2–3, pp. 191–203, 1984.

[49] H. Sun, S. Wang, and Q. Jiang, “FCM-based model selection algorithms for determining the number of clusters,” Pattern Recognition, vol. 37, no. 10, pp. 2027–2037, 2004.

[50] S. Khalil, C. Amrit, T. Koch, and E. Dugundji, “Forecasting public transport ridership: management of information systems using CNN and LSTM architectures,” Procedia Computer Science, vol. 184, pp. 283–290, 2021.

[51] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” Neural Computation, vol. 1, no. 2, pp. 270–280, 1989.

[52] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[53] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” Physica D: Nonlinear Phenomena, vol. 404, Article ID 132306, 2020.

[54] A. Graves, Supervised sequence labelling with recurrent neural networkSpringer Berlin Heidelberg, Berlin, Germany, 2012.

[55] Q. Lu, S. Li, T. Yang, and C. Xu, “An adaptive hybrid XdeepFM based deep interest network model for click-through rate prediction system,” Peerf Computer Science, vol. 7, 2021.

[56] S. M. Ebadi Jokandan, P. Bayat, and M. Farrokhbakhht Founman, “CS- and GA-based hybrid evolutionary sampling algorithm for large-scale social networks,” Social Network Analysis and Mining, vol. 11, no. 1, 1–15, 2021.

[57] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150, Portland Oregon, June 2011.

[58] R. K. Behera, M. Jena, S. K. Rath, and S. Misra, “Co-LSTM: convolutional LSTM model for sentiment analysis in social big data,” Information Processing & Management, vol. 58, no. 1, Article ID 102435, 2021.

[59] O. Frank, “Survey Sampling in Networks,” The Sage handbook of social network analysis, vol. 1, pp. 389–403, 2011, https://www.researchgate.net/publication/289401526_Survey_sampling_in_networks.