

Article

SC-Political ResNet: Hashtag Recommendation from Tweets Using Hybrid Optimization-Based Deep Residual Network

Santosh Kumar Banbhrani * , Bo Xu , Haifeng Liu  and Hongfei Lin 

School of Computer Science and Technology, Dalian University of Technology, Ganjingzi District, Dalian 116024, China; xubo@dlut.edu.cn (B.X.); liuhaifeng@mail.dlut.edu.cn (H.L.); hflin@dlut.edu.cn (H.L.)
* Correspondence: banbhrani@gmail.com

Abstract: Hashtags are considered important in various real-world applications, including tweet mining, query expansion, and sentiment analysis. Hence, recommending hashtags from tagged tweets has been considered significant by the research community. However, while many hashtag recommendation methods have been developed, finding the features from dictionary and thematic words has not yet been effectively achieved. Therefore, we developed an effective method to perform hashtag recommendations, using the proposed Sine Cosine Political Optimization-based Deep Residual Network (SC-Political ResNet) classifier. The developed SCPO is designed by integrating the Sine Cosine Algorithm (SCA) with the Political Optimizer (PO) algorithm. Employing the parametric features from both, optimization can enable the acquisition of the global best solution, by training the weights of classifier. The hybrid features acquired from the keyword set can effectively find the information of words associated with dictionary, thematic, and more relevant keywords. Extensive experiments are conducted on the Apple Twitter Sentiment and Twitter datasets. Our empirical results demonstrate that the proposed model can significantly outperform state-of-the-art methods in hashtag recommendation tasks.

Keywords: deep learning; hashtag recommendation; long short-term memory; natural language processing; sine cosine algorithm; text classification; tweet modeling; word embedding



Citation: Banbhrani, S.K.; Xu, B.; Liu, H.; Lin, H. SC-Political ResNet: Hashtag Recommendation from Tweets Using Hybrid Optimization-Based Deep Residual Network. *Information* **2021**, *12*, 389. <https://doi.org/10.3390/info12100389>

Academic Editors: Ida Mele and Luis Martínez López

Received: 23 July 2021
Accepted: 9 September 2021
Published: 22 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The micro-blogging platform Twitter has become one of the best-known social networks on the internet. Twitter offers a platform wherein users can create a set of follower connections, in order to share their views, and subscribe to subjects posted by their followers. Twitter is considered among the first social networking sites to use the hashtag concept [1]. Due to the proliferation of micro-blogging services, there is an ongoing expansion of short-text over the Internet. Due to the production of huge micro-posts, there exists a requirement for effectual categorization and data searching. Twitter is one of the highly rated micro-blogging sites that permits users to exploit hashtags to classify day-to-day posts. However, some tweets do not comprise tags, thus obstructing the search quality [2]. In twitter, users are liberally permitted to allocate hashtags to their tweets, which are nothing but a string whose prefix is the hash symbol, and are used to catalog their posts and highlight certain content. The hashtag is considered a community-driven principle that adds extra context to tweets. This method assists in searching, rapidly propagates the topic from millions of users, and allows users to link into specific discussions [3]. Since its introduction, twitter has become a popular standard for commencing electronic communications. Twitter utilizes post tweets that reflect an enormous variety of topics, including news, personal ideas, political affairs, events, technologies, and celebrities. Furthermore, Twitter permits its users to pursue other users, in order to track their attention [4].

Twitter possesses in-depth reach, due to the utilization of mobile applications such as smartphones. Tweets are communal in nature, and are explicitly available to every person

when posted using Twitter [2]. The culture of tagging is extensive; thereby, the hashtag recommendation model has acquired the interest of researchers. Recent literary works have been devised to recommend specific hashtags or infer topics hidden in the tweets. Even though such systems help motivate and assist users in acquiring tagging habits, it is not adequate for the information seeker who desires to discover promising hashtags. The recommendation of well-known hashtags imitates timely topics, but may involve deeply utilized universal hashtags, in which the suggestions are not personalized [5]. Hashtag recommendation addresses the suggestion that one or more hashtags should be used to tag a post made on the social platform. Hashtag users, who tend to allocate posts to a certain social network, rely on the message's key facets [6]. Hashtags are the major unit of Twitter, operating as a tagging method for grouping messages related to similar topics. The dynamic aspects of hashtags have led to the proposal of various research issues in recent years, involving topics such as semantic hashtag classification and hashtag recommendation-based discovery of events. As users generate hashtags, there is a possibility of excessive hashtags being created in a similar topic. In addition, it is a complex and lengthy process for the user to search associated hashtags. In this circumstance, the recommendation of pertinent hashtags to a user, based on their attention and preference, is complex [7].

Hashtag recommendation for Twitter has acquired huge interest amongst researchers in the natural language processing (NLP) field. Literary works have focused on the content similarity of tweets [8] and modeled topics using Latent Dirichlet Allocation (LDA) [9]. Various techniques [10] have been devised based on deep learning [11], which are considered promising methods. However, the majority of techniques have been devised for hashtag recommendation using micro-blogging platforms. These are mostly devised based on a syntactic criterion or conventional tagging methodologies. They range from probabilistic models to the utilization of classification to perform collaborative filtering. Thus, these techniques present a common shabbiness, wherein they utilize sparse lexical features, such as bag-of-words (BoW), to indicate tweets, while ignoring the semantic behind tweets with an equivalent illustration of multiple NLP tasks. Popular techniques which have been adapted for hashtag recommendation are Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM), which have good ability for learning sequence representations [12–15]. The classical methods of hashtag recommendation range from collaborative filtering to classification and probabilistic models, including topic models and Naïve Bayes. In addition, the majority of techniques rely on sparse lexical features that involve BoW models and delicately devise patterns. However, feature engineering is labor-intensive, as sparse and discrete features encode syntactic data from words. Meanwhile, neural techniques have been shown to have the ability to learn effectual representations and have presented improved performance in different NLP tasks [16]. The LSTM and modified RNN have been extensively employed in this respect, due to their ability to capture long-term dependencies [17].

This research was designed by employing a Deep Residual Network for recommending hashtags from twitter data. The proposed SCPO consists of the integration of the SCA and the PO algorithm. The SCA creates multiple initial random candidate solutions and requires them to fluctuate outwards or towards the best solution, using a mathematical model based on sine and cosine functions. This algorithm can explore different regions of a search space, avoid local optima, converge towards the global optimum, and exploit promising regions of a search space effectively during optimization. It can be highly effective in solving real problems with constrained and unknown search spaces. The PO consists of the mathematical mapping of all of the major phases of politics, including constituency allocation, party switching, election campaigns, inter-party elections, and parliamentary affairs. This algorithm assigns each solution a dual role, by logically dividing the population into political parties and constituencies, facilitating each candidate to update its position concerning the party leader and the constituency winner. The PO algorithm has excellent convergence speed, with good exploration capability in early iterations. This algorithm is invariant to function shifting and performs consistently in very high-dimensional

search spaces. We integrate the PO algorithm with the SCA, in order to attain the global best solution.

The input data considered are twitter data, called tweets. Input tweets are processed more effectively, in order to remove the stop words. The process of stop word removal from the input data increases the quality of data for further processing. After removing stop words, the keyword set is computed by finding dictionary words, thematic words, and more relevant words. The dictionary words are the unique keywords retrieved from all the tweets in the training data. Thematic words are extracted based on the co-occurring frequency words, whereas more relevant words repeatedly occur among the tweets. After forming the keyword set, the features associated with the keyword sets are extracted by considering the dictionary Term frequency (TF), document Inverse document frequency (IDF), thematic words, and the more relevant word features. For each tweet, the document TF, document IDF, thematic words, and the more relevant features are extracted, and their respective variance measures are computed more effectively. The input to the Deep LSTM consists of the features and their variance measures, such that the Deep LSTM can compute the hashing score by considering all the features to generate the feature vector. The Deep Residual network takes the feature vector as input. It processes the features more effectively, in order to recommend the hashtag. The SCPO algorithm carries out the training of the deep learning classifier.

The major contributions of this research are as follows:

- We model an effective Hashtag Recommendation system using a Deep Residual Network Classifier, trained by the SCPO algorithm, which is derived by integrating the SCA and the PO algorithm;
- The incorporation of features in the optimization algorithm pushes the solution towards the global optimum rather than local optima;
- We conduct extensive experiments on different datasets, in which the proposed method is shown to outperform state-of-the-art methods in the Hashtag Recommendation task.

The remainder of this paper is organized as follows: Section 2 provides a review of different Hashtag Recommendation methods. Section 3 briefly introduces the architecture of the proposed framework. The system implementation and evaluation are described in Section 4, and the results and discussions are given in Section 5. Finally, Section 6 concludes the overall work and provides recommendations for future research.

2. Related Work

Some traditional hashtag recommendation methods are reviewed in this section. Nada Ben-Lhachemi and El Habib Nfaoui [18] introduced a DBSCAN clustering technique for partitioning tweets into clusters consisting of semantically and syntactically relevant tweets. This method has the potential to recommend pertinent hashtags syntactically and semantically from a given tweet; however, it failed to involve deep architectures on various semantic knowledge bases, in order to enhance the accuracy. Yang Li et al. [17] introduced a Topical Co-Attention Network (TCAN) to guide content and topic attention. This method was shown to be useful in determining structures from large documents; however, the framework failed to involve temporal information. Nada Ben-Lhachemi et al. [15] introduced an LSTM network for encoding the tweet vector concerning the representation. The skip-gram model was employed to generate the tweet embeddings, considering the vector-based representation. It was capable of recommending suitable hashtags, but failed to utilize semantic knowledge bases to improve the accuracy. Asma Belhadi et al. [19] introduced the PM-HRec model to address the issues of hashtag recommendation with two stages; namely, offline and online processing. Offline processing was used to transform the tweet corpus to a transnational database based on temporal information, whereas online processing was used to extract related hashtags. This method works effectively on huge data, but failed to include parallel GPUs to handle large data. Areej Alsini et al. [20] introduced a community-based hashtag recommendation model by applying a tweet similarity

process. This method makes it easy to understand the different factors influencing hashtag recommendation, but it involves a lengthy process.

Table 1. Literature review.

Authors	Methods	Advantages	Disadvantages
Ben-Lhachemi, N. et al. [18]	DBSCAN clustering technique.	Has the potential to recommend pertinent hashtags syntactically and semantically from a given tweet.	Fails to involve deep architectures on various semantic knowledge bases to enhance accuracy.
Li, Y. et al. [17]	Topical Co-Attention Network (TCAN).	Useful for determining structures from large documents.	Fails to involve temporal information.
Ben-Lhachemi, N. and Boumhidi, J. [15]	Long Short-Term Memory Recurrent Neural Network.	Capable of recommending suitable hashtags.	Fails to utilize semantic knowledge bases to improve accuracy.
Belhadi, A. et al. [19]	Pattern Mining for Hashtag Recommendation (PM-HRec).	Works effectively on huge data.	Fails to include parallel methods (e.g., GPUs) to handle large data.
Alsini, A. et al. [20]	Community-based hashtag recommendation model.	Easy to understand different factors that influence hashtag recommendation.	Lengthy process.
Yang, Q. et al. [21]	Attention-based multimodal neural network model (AMNN).	Able to attain improved performance with multimodal micro-blogs.	Fails to adapt external knowledge, such as comments and user information, for recommendation.
Ma, R. et al. [22]	Co-attention memory network.	Can manage a huge number of hashtags.	The generation of high-quality candidate sets is complex.
Cao, D. et al. [23]	Neural network-based LOGO.	Assists in capturing sequential and considerate features at the same time.	Fails to include recommendation on micro-videos.
Can Li et al. [24]	Tag recommendation method with Deep learning and Collaborative filtering (TagDC)	Helps to locate similar software.	Fails to consider unpopular tags.
Sahraoui et al. [25]	Hybrid filtering technique in a social network for data mining.	Predicts the user's interest in a social network.	Fails to evaluate the performance in the signed Network.

Qi Yang et al. [21] developed an AMNN to learn a representation of multimodal micro-blogging and to recommend related hashtags. The features from both images and text were extracted using a hybrid neural network model. However, it improved performance with multimodal micro-blogs but failed to adapt to external knowledge, such as comments and user information, for the purpose of recommendation. Renfeng Ma et al. [22] developed a Co-attention memory network for the recommendation of hashtags, in which the users were allowed to introduce some new hashtags dynamically. This method can manage a huge number of hashtags; however, the generation of high-quality candidate sets is complex. Da Cao et al. [23] developed a Neural Network-based LOGO model for recommending hashtags by considering multiple modalities. It assists in capturing sequential and considerate features simultaneously, but fails to include a recommendation on micro-videos. Can Li et al. [24] developed a tag recommendation technique—the Tag recommendation method with Deep learning and Collaborative filtering (TagDC)—which was developed with two different modules: initially, word learning is employed using an enhanced CNN; then, collaborative filtering is employed for tag recommendation on soft-

ware sites. The performance was evaluated based on the F-Measure, recall, and precision. However, they failed to consider unpopular tags. Sahraoui et al. [25] developed a hybrid filtering technique for mining the data in a social network. In this, Big Five personality traits were devised for the mining; then, the user's interest is represented in the graph and predicted using the meta path discovery method. Its efficiency was evaluated using the precision, recall, and F-Measure. However, they failed to evaluate its performance in signed networks. A community-based hashtag recommendation model [20] was presented for performing recommendations with tweets. However, it involves a complex and lengthy process for searching related hashtags. Moreover, the classical techniques of hashtag recommendation utilize probabilistic models, such as topic models and Naïve Bayes. In addition, the majority of techniques rely on sparse lexical features involving the Bag-of-Words (BoW) model and devised structures. However, the associated feature engineering is challenging, and these methods cannot encode semantic and syntactic information [17]. Table 1 depicts the literature review in detail.

3. Proposed Framework

The overall architecture of the SCPO-based Deep Residual Network for Hashtag Recommendation is illustrated in Figure 1, comprising several components. The details of each component are presented in the following.

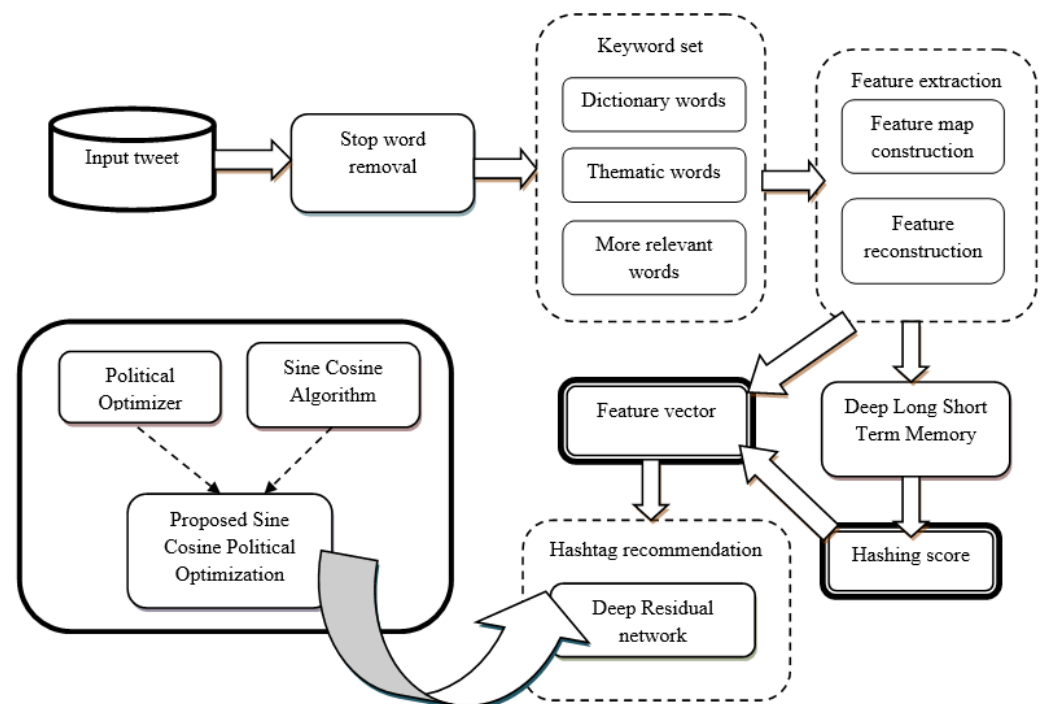


Figure 1. An illustration of the proposed SCPO-based Deep Residual Network for Hashtag Recommendation.

A user uses a hashtag by inserting a hash symbol (#) in front of a word, in order to access the data more easily. Due to the growth of information being shared among social services, such as Flickr and Instagram, a number of micro-blogs focused on certain the hashtags are available. Existing hashtag recommendation methods have been designed solely considering textual data. Hence, in this research, we focus on designing a model using an SCPO-based Deep Residual Network for recommending hashtags. Here, the input tweet is passed through a stop word removal phase, in order to eliminate the stop words associated with the input tweet data. The keyword set is computed based on dictionary words, thematic words, and more relevant words. The features extracted from the keyword

set and hashing score computed from the Deep LSTM are passed as input to the Deep Residual Network, which performs the recommendation process.

3.1. Removal of Stop Words from Input Tweet

Informal languages are commonly used in tweets, which are restricted to 140 characters. However, this restriction forces the user to be imaginative, in terms of shortening words, writing abbreviations, and using symbols and emoticons. Moreover, the tweet may contain special components, such as URLs, media, user mentions, and hashtags. Let us consider a dataset comprised of a number of tweets, represented as

$$\alpha = \{T_1, T_1, \dots, T_j, \dots, T_n\}, \quad (1)$$

where α denotes the database, T_j indicates the tweet located at j^{th} index of the dataset, and n represents the total number of tweets.

The words associated with tweet T_j are expressed as

$$T_j = \{W_1, W_2, \dots, W_k, \dots, W_L\}, \quad (2)$$

where W_L denotes the total number of words in the tweet.

A stop word is the word that appears often and does not contain any meaning. Stop word removal is the process of removing these words, including ‘the’, ‘a’, ‘is’, ‘an’, and so on. As the stop word does not have any sentiment, it is very significant to remove these words from the tweet. The stop word removal process increases the performance of hashtag recommendation by reducing the dimensionality rate. After removing the stop words from tweet T_j , the resulting output tweet is represented as T_{sr} .

3.2. Keyword Set: Dictionary, Thematic and More Relevant Words

In the keyword set process, the extraction of dictionary words, thematic words, and more relevant words is carried out on the tweet.

Dictionary words: The words that have a unique keyword set from all the reviews of training data are termed as dictionary words. A tweet contains a number of words, whereas the dataset contains a number of tweets. Here, the dictionary words associated with the entire dataset α are selected by individually processing each tweet T_j , and the selected dictionary words are included into the dictionary word set, denoted by DW .

Thematic words: These are words which are closely associated with the theme of the text. Thematic words can be easily incorporated in the summarizing statement of text, by expressing the meaning of the text. The words having co-occurring frequencies in a set of words are selected from each tweet T_j of the entire dataset α , in order to find the thematic words. The dataset may contain a few words with co-occurring frequency; among those the top N words with co-occurring frequency set are selected as thematic words. The set of thematic words selected from the dataset is termed the thematic word set, denoted by TW .

More relevant words: Each tweet may contain few relevant words while, when processing the entire dataset, there may exist several relevant words while individually processing each tweet T_j . By selecting the more relevant words from each tweet, the more relevant word set can be formed, which is denoted by MR .

The keyword set comprises three subsets—namely, DW , TW , and MR —such that the keyword set can be written as $T_j = \{DW, TW, MR\}$.

3.3. Feature Extraction Based on Keyword Set

After finding the keyword set, the features associated with the keyword set are selected by constructing the feature map. This section explains the construction of feature maps, feature reconstruction, and the computation of hashing scores.

3.3.1. Feature Map Construction

The feature map is constructed for the keyword set by considering each tweet word in the dataset. For an individual word in a tweet, the feature map is constructed based on dictionary keywords, thematic keywords, and more relevant keywords. However, the dictionary feature is constructed using the *TF* and *IDF* for each word W_K of the tweet. Figure 2 represents the schematic view of feature map construction. *TF* is the number of times each word W_K is present in the tweet, while W_K indicates W^{th} word of the tweet. The *IDF* is computed based on the ratio of the total number of tweets to the number of tweets that contain the word W_K , measured as

$$IDF = \log \frac{T_n}{DF(W_K)}. \tag{3}$$

Let us consider the number of words in a tweet to be four; hence, the document *TF* computed for the four words of the tweet is represented as X^1 , document *IDF* measured for the same words of the tweet is represented as X^2 , and thematic and more relevant keywords measured for the four words of the tweet are denoted by X^3 and X^4 , respectively.

	W_1	W_2	W_3	W_4
Document TF				
Document IDF				
Thematic				
More relevant				

Figure 2. Schematic view of feature map construction.

3.3.2. Feature Reconstruction

After constructing the feature map, the features obtained from the keyword set using the dictionary *TF*, dictionary *IDF*, thematic keyword, and the non-relevant keywords are represented as f_1, f_3, f_5 , and f_7 , respectively, indicating the keyword set’s mean value, where

$$f_1 = \frac{1}{L} \sum_{v=1}^L X_v^1, \tag{4}$$

$$f_2 = \frac{1}{L} \sum_{v=1}^L (X_v^1 - f_1)^2, \tag{5}$$

where f_1 denotes the features acquired using dictionary *TF* and the variance measure of respective f_1 is represented as f_2 .

$$f_3 = \frac{1}{L} \sum_{v=1}^L X_v^2, \tag{6}$$

$$f_4 = \frac{1}{L} \sum_{v=1}^L (X_v^2 - f_3)^2. \tag{7}$$

The features acquired from the keyword set based on the dictionary *IDF* are denoted as f_3 , and their respective variance is indicated as f_4 .

$$f_5 = \frac{1}{L} \sum_{v=1}^L X_v^3, \tag{8}$$

$$f_6 = \frac{1}{L} \sum_{v=1}^L (X_v^3 - f_5)^2. \tag{9}$$

The feature f_5 is acquired from the keyword set, based on thematic words, and the variance measure of the respective feature f_5 is represented as f_6 .

$$f_7 = \frac{1}{L} \sum_{v=1}^L X_v^4, \tag{10}$$

$$f_8 = \frac{1}{L} \sum_{v=1}^L (X_v^4 - f_7)^2. \tag{11}$$

The features acquired based on more relevant words are represented as f_7 , and their corresponding variance measure is represented as f_8 . Here, L indicates the total number of words in the tweet, and features $f_2, f_4, f_6,$ and f_8 denote the variances of the corresponding features $f_1, f_3, f_5,$ and $f_7,$ respectively. The features acquired from the keyword set have the dimension of $[V \times 8]$.

3.3.3. Hashing Score Prediction by Deep LSTM

Deep LSTM [26] is employed to determine the hashing score, considering the keyword set features as input. A deep neural Network constructed by using a number of layers within the LSTM framework is called a Deep LSTM. It takes advantages of the benefits of both deep neural Networks and LSTM Networks to obtain the hashing score. It consists of internal state cells, which act as memory cells. The output of the Deep LSTM is the hashing score, which is computed based on the cell states. The working process of this Network solely depends on the memory cells.

The input node receives the input feature f , where $f = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$, at the input layer.

$$Q_F = \tanh(f * K_{OF} + M_{F-1}K_{QM} + Z_Q), \tag{12}$$

where \tanh denotes the hyperbolic tangent function, Q_F denotes the input node at time F , f denotes the input feature, K_{Of} indicates the weights among the input layer and input nodes of the memory cell, M_{F-1} represents the input of the hidden state, K_{QM} denotes weight matrix for the hidden state, and Z_Q indicates the bias of input node.

The input gate, Y_F , is equivalent to that of the input node, and it receives the input feature f . The input gate blocks information flow between the nodes. The input gate is represented as

$$Y_F = X(f * K_{OF} + M_{F-1}K_{QM} + Z_Y), \tag{13}$$

where Y_F denotes the input gate, Z_Y denotes the bias of the input gate, and X specifies the sigmoid activation function.

The internal state, λ_F , is a node with a self-loop recurrent edge consisting of a linear activation function and a unit weight, such that the internal state can be represented as

$$\lambda_F = Y_F \odot Q_F + \lambda_{F-1}, \tag{14}$$

where λ_F denotes the internal state at time F .

The forget gate is the sub-unit used for re-initiating the internal state of the memory cell, which is computed as

$$\beta_F = X(f * K_{Mf} + M_{F-1}K_{\beta M} + Z_\beta), \tag{15}$$

where β_F denotes the forget gate, $K_{\beta M}$ denotes weights among the forget gate and the hidden states, and K_{Mf} represents the weights among the forget gate and the input layer.

Finally, the output gate, γ_F , is expressed as

$$\gamma_F = X(f * K_{\gamma f} + M_{F-1}K_{\gamma M} + Z_\gamma), \tag{16}$$

where $K_{\gamma f}$ denotes the weights among the output gate and the input layer, $K_{\gamma M}$ represents the weights among the output gate and the hidden states, and Z_γ denotes the bias of the output gate. The output of memory cell is calculated as

$$M_F = \tanh(\lambda_F) \odot_{\gamma_F} \tag{17}$$

where $\lambda_F = Q_F \odot_f + \lambda_{F-1} \odot \beta_F$ and \odot denotes the XNOR operator. The score computed using the Deep LSTM is termed the hashing score, and is represented as f_9 with the dimension of $[V \times 1]$. It is used to perform the hashtag recommendation, by passing it as an input to the Deep Residual Network, in addition to the features of the keyword set. Figure 3 shows the structure of the Deep LSTM classifier.

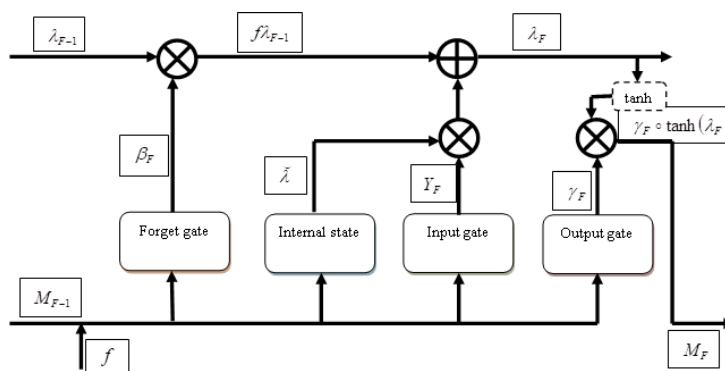


Figure 3. Structure of Deep LSTM classifier.

3.3.4. Computation of Feature Vector

The feature vector is computed by considering the features of the keyword set, along with the hashing score. The feature vector used to perform the process of hashtag recommendation is represented as

$$\mu = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9\}, \tag{18}$$

where μ denotes the feature vector, with the dimension of $[V \times 9]$.

3.4. Hashtag Recommendation Using Proposed SC-Political ResNet

To make the tweet searching process easy, twitter users use hashtags for classifying their tweets. In this paper, the process of hashtag recommendation is carried out using a Deep Residual Network, in such a way that the training process of deep learning classifier is carried out using the SCPO algorithm.

3.4.1. Architecture of Deep Residual Network

The process of hashtag recommendation is carried out by a Deep Residual Network, while the training process of the classifier is done using the proposed SCPO algorithm. The structure of a Deep Residual Network [27] is composed of different layers; namely, residual blocks, Convolutional (conv) layers, average pooling layers, and linear classifiers. Figure 4 represents the structure of the Deep Residual Network.

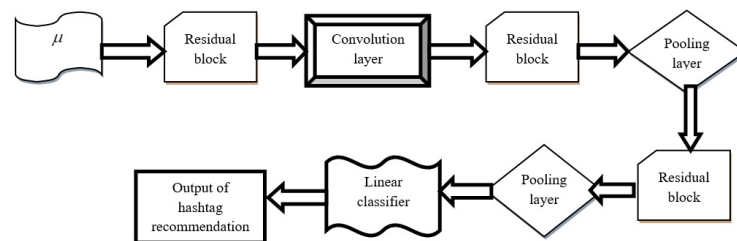


Figure 4. Architecture of Deep Residual Network.

Convolutional layer: The two-dimensional conv layer (conv2d) is used to significantly reduce the number of free parameters in the training procedure and to ensure performance improvement by utilizing the advantages of weight sharing and the local receptive field. This layer processes the input data using a series of filters based on local connections. The series of filters used in this layer is termed the convolution kernel. The computation process of convolution layer involves the mathematical operation used for sliding the individual filter over input matrix, calculating the dot product in the kernel. The calculation process for the convolutional layer can be expressed as

$$Conv2d(B) = \sum_{z=0}^{b-1} \sum_{h=0}^{b-1} A_{z,c} \bullet \mu_{(u+z),(v+c)}, \tag{19}$$

$$Conv1d(B) = \sum_{l=0}^{X_{in}-1} A_l * \mu, \tag{20}$$

where μ denotes the feature vector, u and v are utilized to observe the coordinates in the 2D input, A indicates a $(b * b)$ kernel matrix (which is learned in the training process), z and c indicates the position indices of the kernel matrix, A_l indicates kernel size for the l^{th} input, and $*$ denotes the cross-correlation operator.

Pooling layer: This layer is present between the conv layers, and is generally used to minimize the spatial dimension of the feature vector and to control overfitting.

$$z_{out} = \frac{z_{in} - l_z}{a} + 1, \tag{21}$$

$$c_{out} = \frac{c_{in} - l_c}{a} + 1, \tag{22}$$

where z_{in} denotes the width of the input two-dimensional matrix, c_{in} denotes the height of the two-dimensional matrix, and z_{out} and c_{out} represent the width and height of the output two-dimensional matrix. Moreover, l_z denotes the width of the kernel, and l_c represents the height of the kernel.

Activation function: This function is employed to learn the nonlinear and complicated features, thus enhancing the nonlinearity of features. Here, the Rectified Linear Unit (ReLU) is used for the activation function, which is expressed as

$$B(\mu) = \begin{cases} 0; \mu < 0 \\ \mu; \mu \geq 0 \end{cases} \tag{23}$$

where μ indicates the input feature vector.

Batch normalization: The training set is divided into different small sets called mini-batches. This can effectively train the model to achieve a better trade-off between convergence and computational complexity. However, as the mini-batch model suffers from the internal co-variate shift, the stability and training speed may be reduced. To overcome this issue, batch normalization has been designed to minimize the internal co-variate shift, using input layer normalization by adjusting and scaling the activation functions. Hence,

the reliability and training speed can be improved by alleviating gradient explosion and overfitting issues, along with a better learning rate.

Residual blocks: A shortcut connection is made from the input to the output and, through this connection, the inputs are directly attached to output only if the input and output have the same dimension. For various dimensions, the dimension matching factor is employed for matching input, as well as the output dimensions, which is expressed as

$$C = g(\mu) + \mu, \tag{24}$$

$$C = g(\mu) + w_f \mu, \tag{25}$$

where μ denotes the input to the residual blocks, C indicates the output, g specifies the mapping relationship between the input and output, and w_u indicates the dimension matching factor.

Linear classifier: Hashtag recognition is carried out by a linear classifier. Here, the linear classifier contains a soft-max function and a fully connected layer. The FC , with dot product, is calculated as

$$C = w_{U*V} \mu_{V*1} + D_{U*1}, \tag{26}$$

$$Softmax(\mu_u) = \frac{e^{\mu_u}}{\sum_{l=1}^{X_0} e^{\mu_l}}; u = 1, 2, \dots, X_0, \tag{27}$$

where W denotes the weight matrix of size $[U \times V]$, μ_{V*1} indicates the input feature vector of size $[V \times 1]$, and D denotes the bias. The softmax function is employed to normalize the input vector to a probability vector corresponding to each class, such that class with highest probability is selected as the output result.

3.4.2. Training Process of Deep Residual Network Using SCPO

The Deep Residual Network training process is carried out using the proposed SCPO, which was derived through the integration of the SCA [28] and PO [29] algorithms. The SCA generates a number of candidate solutions randomly and fluctuates them towards an optimal solution using mathematical functions consisting of sine and cosine functions. Adaptive parameters are used to emphasize the exploitation and exploration of the search space. It considers test cases that move the solution towards global optima and offer better convergence. Meanwhile, the PO algorithm considers two perspectives, where each solution optimizes the goodwill to win the election, whereas each party tries to maximize their number of seats in parliament, in order to form a government. Here, the party members are considered as the candidate solution and the position of the candidate in the solution space is termed as their goodwill. However, the goodwill of a political member is defined based on performance-related factors, mimicking the components or variables of the candidate solution. The PO involves five phases; namely, party formation, election campaign, party-switching, inter-party election, and parliamentary phases. The algorithmic procedure involved in the proposed SCPO is elaborated in the following:

Initialization: Let us initialize a population E with q of political parties, party members, and number of constituencies. The total number of iterations is represented as h_{max} , while j_{max} denotes the upper limit of party switching. The political parties are initialized as

$$E = \{E_1, E_2, \dots, E_k, \dots, E_q\}. \tag{28}$$

Each party E_k contains q members or candidates in the solution space, and is represented as

$$E_k = \{R_k^1, R_k^2, \dots, R_k^q\}. \tag{29}$$

Each r^{th} member, R_k^r , is assumed to be a potential solution, and each solution is a p -dimensional vector, given as

$$R_k^r = \{R_{k,1}^r, R_{k,2}^r, \dots, R_{k,p}^r\}, \tag{30}$$

where p denotes the number of input variables and $R_{k,i}^r$ denotes the i^{th} dimension of R_k^r .

Compute fitness measure: The best fitness value is evaluated using the fitness measure, such that training the weights using the optimal value of fitness results in higher performance. The fitness measure is represented as

$$F = \frac{1}{N} \sum_{t=1}^N [C_t - Q_t]^2, \tag{31}$$

where F denotes the fitness measure, C is the classifier output, Q denotes the target value, and N denotes the total number of samples.

Compute party leader R^* : The potential solution can act as the election candidate in this phase. The q constituencies are listed as $H = \{H_1, H_2, \dots, H_q\}$, and r^{th} member of individual party from r^{th} constituency H_r is given as $H_r = \{R_1^r, R_2^r, \dots, R_q^r\}$. The party member is the fittest member, selected after the inter-party election. The party leader is selected using the following equation:

$$R_k^* = R_k^x, \tag{32}$$

$$x = \underset{1 \leq r \leq q}{\operatorname{argmin}} S(R_k^r); \forall k = \{1, 2, \dots, q\}, \tag{33}$$

where R_k^* represents the leader of the k^{th} party. The set of all party leaders is expressed as

$$R^* = \{R_1^*, R_2^*, \dots, R_q^*\}. \tag{34}$$

Compute constituency winners H^* : In the reallocation of roles in the constituency, the winner is represented as

$$H_r^* = R_x^*, \tag{35}$$

$$x = \underset{1 \leq k \leq q}{\operatorname{argmin}} S(R_k^r). \tag{36}$$

Update solution: The standard equation of PO is represented as

$$R_k^{h+1} = s^* + d(s^* - R_k^h); \text{ if } R_k^{h-1} \leq R_k^h \leq s^* \text{ or } R_k^{h-1} \geq R_k^h \geq s^*, \tag{37}$$

and the standard equation of SCA is given as

$$R_k^{h+1} = \begin{cases} R_k^h + d_1 \sin(d_2) \times |d_3 G_k^h - R_k^h| & ; d_4 < 0.5 \\ R_k^h + d_1 \cos(d_2) \times |d_3 G_k^h - R_k^h| & ; d_4 \geq 0.5 \end{cases}. \tag{38}$$

The equation that satisfies the condition $d_4 \geq 0.5$ is selected for training the weights.

$$R_k^{h+1} = R_k^h + d_1 \sin(d_2) \times |d_3 G_k^h - R_k^h|. \tag{39}$$

Let us assume that $G_k^h > R_k^h$. Thus, $|\cdot|$ will be removed, and

$$R_k^{h+1} = R_k^h + d_1 \sin(d_2) \times (d_3 G_k^h - R_k^h), \tag{40}$$

$$R_k^{h+1} = R_k^h + d_1 d_3 \sin(d_2) G_k^h - d_1 \sin(d_2) R_k^h, \tag{41}$$

$$R_k^{h+1} = R_k^h [1 - d_1 \sin(d_2)] + d_1 d_3 \sin(d_2) G_k^h, \tag{42}$$

$$R_k^h [1 - d_1 \sin(d_2)] = R_k^{h+1} - d_1 d_3 \sin(d_2) G_k^h, \tag{43}$$

$$R_k^h = \frac{R_k^{h+1} - d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)}. \tag{44}$$

By substituting Equation (44) into Equation (37), we have

$$R_k^{h+1} = s^* + d \left(s^* - \left(\frac{R_k^{h+1} - d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)} \right) \right), \tag{45}$$

$$R_k^{h+1} + \frac{d R_k^{h+1}}{1 - d_1 \sin(d_2)} = s^* + d \left(s^* + \frac{d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)} \right), \tag{46}$$

$$\frac{R_k^{h+1} (1 - d_1 \sin(d_2)) + d R_k^{h+1}}{1 - d_1 \sin(d_2)} = s^* + d \left(s^* + \frac{d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)} \right), \tag{47}$$

$$R_k^{h+1} \left[\frac{(1 - d_1 \sin(d_2)) + d}{1 - d_1 \sin(d_2)} \right] = s^* + d \left(s^* + \frac{d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)} \right), \tag{48}$$

$$R_k^{h+1} = \frac{1 - d_1 \sin(d_2)}{(1 - d_1 \sin(d_2)) + d} \left[s^* + d \left(s^* + \frac{d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)} \right) \right]. \tag{49}$$

Hence, the update solution of proposed SCPO algorithm can be expressed as

$$R_k^{h+1} = \begin{cases} \frac{1 - d_1 \sin(d_2)}{(1 - d_1 \sin(d_2)) + d} \left[s^* + d \left(s^* + \frac{d_1 d_3 \sin(d_2) G_k^h}{1 - d_1 \sin(d_2)} \right) \right]; d_4 < 0.5 \\ \frac{1 - d_1 \cos(d_2)}{(1 - d_1 \cos(d_2)) + d} \left[s^* + d \left(s^* + \frac{d_1 d_3 \cos(d_2) G_k^h}{1 - d_1 \cos(d_2)} \right) \right]; d_4 \geq 0.5 \end{cases}, \tag{50}$$

where R_k^h denotes the position of the current solution in k^{th} dimension at the h^{th} iteration; $r_1, r_2,$ and r_3 are random numbers; G_k indicates the position of the destination point in the k^{th} dimension; and $|\cdot|$ specifies the absolute value.

Termination: The above steps are repeated until the global best solution is obtained.

Algorithm 1 provides the pseudo-code for the proposed SCPO-based Deep Residual Network.

Algorithm 1: Pseudo-code of proposed SCPO-based Deep Residual network.

```

1      Input:  $R_k^h$ ,
2      Output:  $R_k^{h+1}$ 
3      Initialize  $E$ 
4      Compute fitness measure
5      Compute  $R^*$ 
6      Compute  $H^*$ 
7       $h = 1$ 
8       $E(h - 1) = E$ 
9       $S(E(h - 1)) = S(E)$ 
10      $J = J_{\max}$ 
11     while  $h \leq h_{\max}$ 
12     Do
13      $E_{\text{temp}} = E$ 
14      $S(E_{\text{temp}}) = S(E)$  |
15     for each  $E_k \in E$ 
16     Do
17     for each  $R_k^r \in E_k$ 
18     Do
19      $R_k^r =$  Election campaign ( $R_k^r, R_k^r(h - 1), R^*, H^*$ )
20     End
21     End
22     Party switching ( $E, J$ )
23     Compute  $R^*$  and  $H^*$  at election phase
24     Parliamentary affairs  $H^*, E$ 
25      $E(h - 1) = E_{\text{temp}}$ 
26      $S(E(h - 1)) = S(E_{\text{temp}})$ 
27      $J = J - J_{\max}/h_{\max}$ 
28      $h = h + 1$ 
29     End

```

4. System Implementation and Evaluation

In this section, we first present the used datasets and then describe the experimental setup and baseline benchmarks. Finally, the evaluation metrics are detailed.

4.1. Description of Datasets

In order to evaluate our system, we used the Apple Twitter Sentiment Dataset <https://data.world/crowdfunder/apple-twitter-sentiment/workspace/file?filename=Apple-Twitter-Sentiment-DFE.csv> (accessed on 15 April 2021) and Twitter dataset <https://www.kaggle.com/kavita5/twitter-dataset-avengersendgame?select=tweets.csv> (accessed on 15 April 2021). These datasets are explained in detail below.

- Apple Twitter Sentiment Dataset: This dataset contains 12 columns and 3886 rows, including a number of fields; namely, id, sentiment, query, text, and so on. The size of the dataset is 798.47 KB. However, the text contains hashtags with positive, negative, or neutral tweet sentiments.
- Twitter Dataset: This dataset contains 10,000 records acquired from the twitter of the trending domain “AvengersEndgame”. These data can be used for sentiment analysis. It contains 17 columns with the 8 string-type variables, 4 Boolean-type variables, 3 Integer-type variables, and two other types of variables.

4.2. Experimental Setup

The method we proposed was implemented in Python Programming Language, with the Neural Network library of Tensorflow/Pytorch. Our Networks were trained on an

NVIDIA GTX 1080 in a 64-bit computer with Intel(R) Core(TM) i7-6700 CPU@3.4 GHz, 16 GB RAM, and the Ubuntu 16.04 operating system.

4.3. Evaluation Metrics

The performance of the developed scheme was analyzed by considering the metrics of Precision, Recall, and F1-Score.

Precision: The precision was used to quantify the number of hashtags that are actually classified as belonging to the positive class. It is computed as

$$PR = \frac{A_p}{A_p + B_p}, \quad (51)$$

where A_p denotes the true positive rate, B_p denotes the false positive rate, and PR is the precision.

Recall: Recall is the measure that detects the number of positive predictions achieved by considering all the positive results. It is calculated as

$$RC = \frac{A_p}{A_p + B_n}, \quad (52)$$

where B_n denotes the false negative rate and RC denotes the recall.

F1-Score: The F1-score offers a single score that is a balance between precision and recall. It is computed as

$$FS = 2 * \frac{PR * RC}{PR + RC}, \quad (53)$$

where FS denotes F1-score.

4.4. Baseline Methods

In order to evaluate the effectiveness of the proposed framework, our method was compared with several existing algorithms, including:

- LSTM-RNN Network [15]: For this, the hashtag recommendation was designed by encoding the tweet vector using the LSTM-RNN technique;
- Pattern Mining for Hashtag Recommendation (PMHRec) [19]: PMHRec was designed using the top k high-average utility patterns for temporal transformation tweets, from which the hashtag recommendation was devised;
- Attention-based multimodal neural network model (AMNN) [21]: The AMNN was designed to extract the features from text and images, after which correlations are captured for hashtag recommendation;
- Deep LSTM [26]: Deep-LSTM was designed to evaluate the daily pan evaporation;
- Emhash [30]: For this method, the hashtag recommendation was developed using a neural network-based BERT embedding;
- Community-based hashtag recommendation [20]: For this, the hashtag recommendation was designed using the communities extracted from social Networks.

5. Results and Discussion

The performance results of our proposed model are presented in this section. The results are compared with previously introduced methods, which were tested on the same datasets.

5.1. Results Based on Apple Twitter Sentiment Dataset

This section describes the analysis on the Apple twitter sentiment dataset. Figure 5 shows the analysis of the proposed model when using features f_1 – f_6 . Figure 5a shows the precision measure. With 70% of the training data, the precision measured by the proposed SCPO-based Deep Residual network was 0.88, making it 14.27%, 13.01%, 7.85%, 7.48%, 5.31%, and 4.30% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and

Community-based hashtag recommendation, respectively. The analysis of the recall measure is presented in Figure 5b. With 80% of the training data, the recall measured by the proposed SCPO-based Deep Residual network was 0.91, making it 8.64%, 7.98%, 5.98%, 5.36%, 3.81%, and 3.02% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the F1-score measure of the proposed method in presented in Figure 5c. With 70% of the training data, the F1-score measured by the proposed SCPO-based Deep Residual network was 0.87, making it 16.01%, 14.62%, 8.89%, 8.47%, 6.63%, and 5.36% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively.

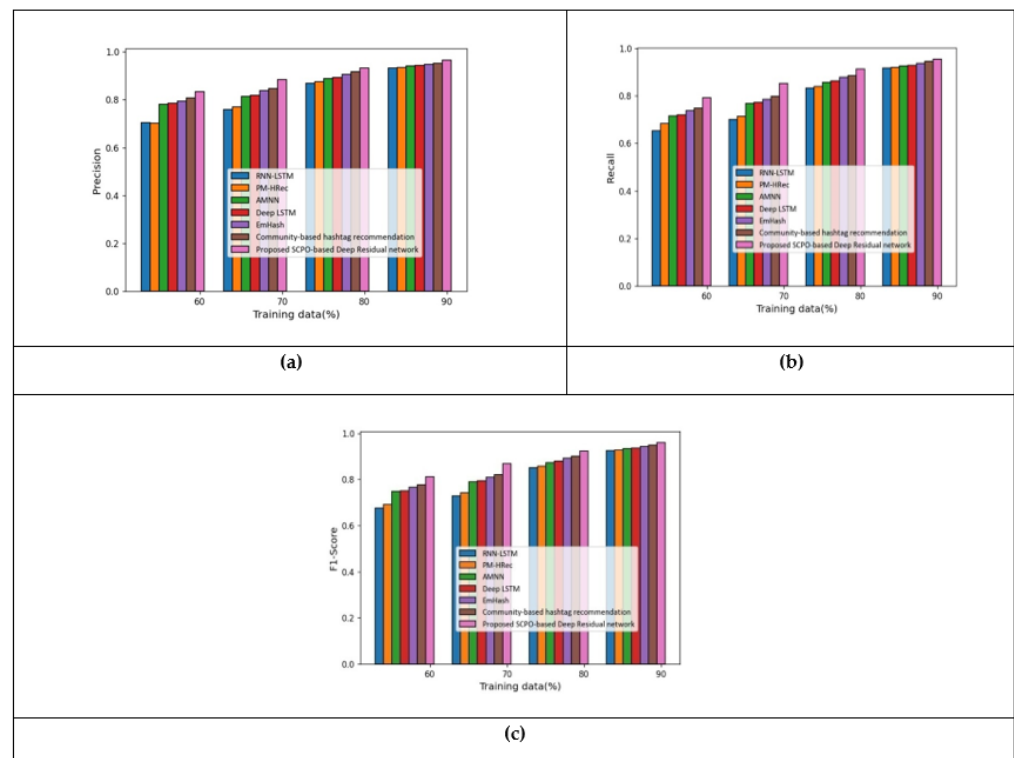


Figure 5. Analysis using features f_1 – f_6 based on Apple twitter sentiment dataset: (a) Precision; (b) Recall; and (c) F1-score.

The analysis of the developed approach, considering the feature f , is illustrated in Figure 6. Figure 6a portrays the precision metric. With 70% of the training data, the precision measured by the proposed SCPO-based Deep Residual network was 0.90, making it 13.56%, 11.16%, 8.82%, 7.15%, 5.66%, and 4.13% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the recall measure is presented in Figure 6b. With 80% of the training data, the recall measured by the proposed SCPO-based Deep Residual network was 0.92, making it 9.89%, 9.23%, 7.44%, 6.83%, 5.42%, and 3.37% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the F1-score measure is presented in Figure 6c. With 70% of the training data, the F1-score measured by the proposed SCPO-based Deep Residual network was 0.88, making it 16.31%, 13.82%, 8.73%, 6.36%, 5.00%, and 3.61% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively.

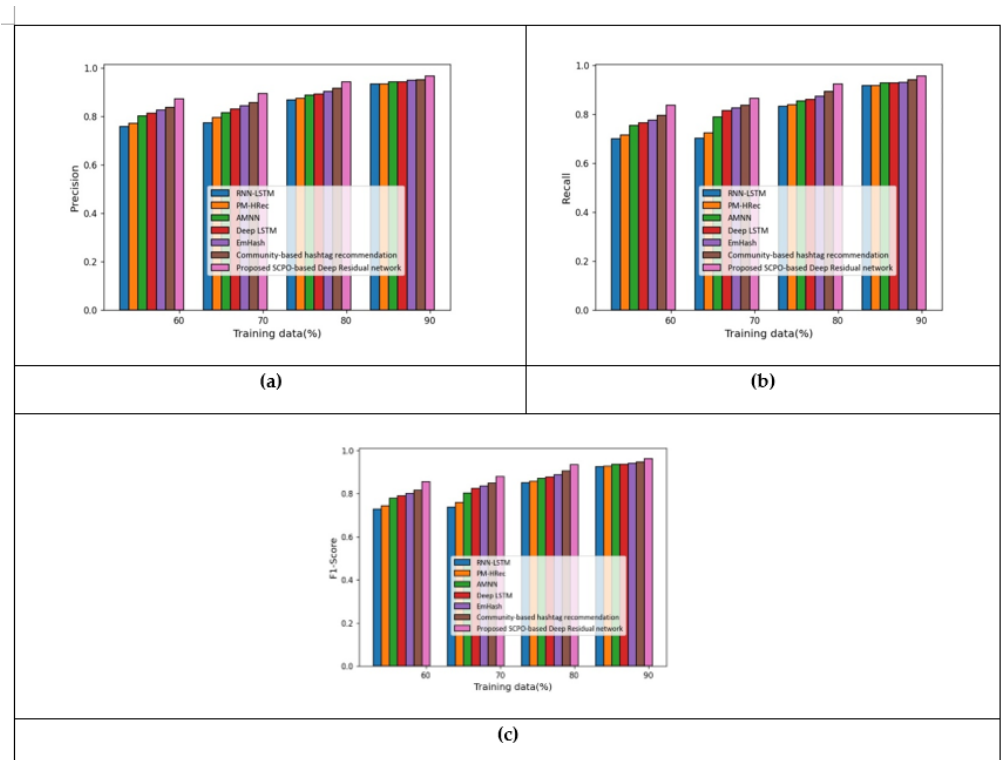


Figure 6. Analysis with f features based on the Apple twitter sentiment dataset: (a) Precision; (b) Recall; and (c) F1-score.

5.2. Results Based on Twitter Dataset

The comparative analysis carried out considering the Twitter dataset is detailed in this section. Figure 7 presents the analysis when considering the features f_1 – f_6 . Figure 7a presents the analysis with respect to the precision measure. With 80% of the training data, the precision measured by the proposed SCPO-based Deep Residual network was 0.87, making it 5.14%, 4.65%, 2.81%, 2.60%, 1.79%, and 0.76% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the recall measure is presented in Figure 7b. With 60% of the training data, the recall measured by the proposed SCPO-based Deep Residual network was 0.78, making it 7.94%, 6.54%, 5.24%, 4.74%, 3.80%, and 2.26% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the F1-score measure is presented in Figure 7c. With 70% of the training data, the F1-score measured by the proposed SCPO-based Deep Residual network was 0.85, making it 9.05%, 8.23%, 5.66%, 5.51%, 4.10%, and 2.80% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively.

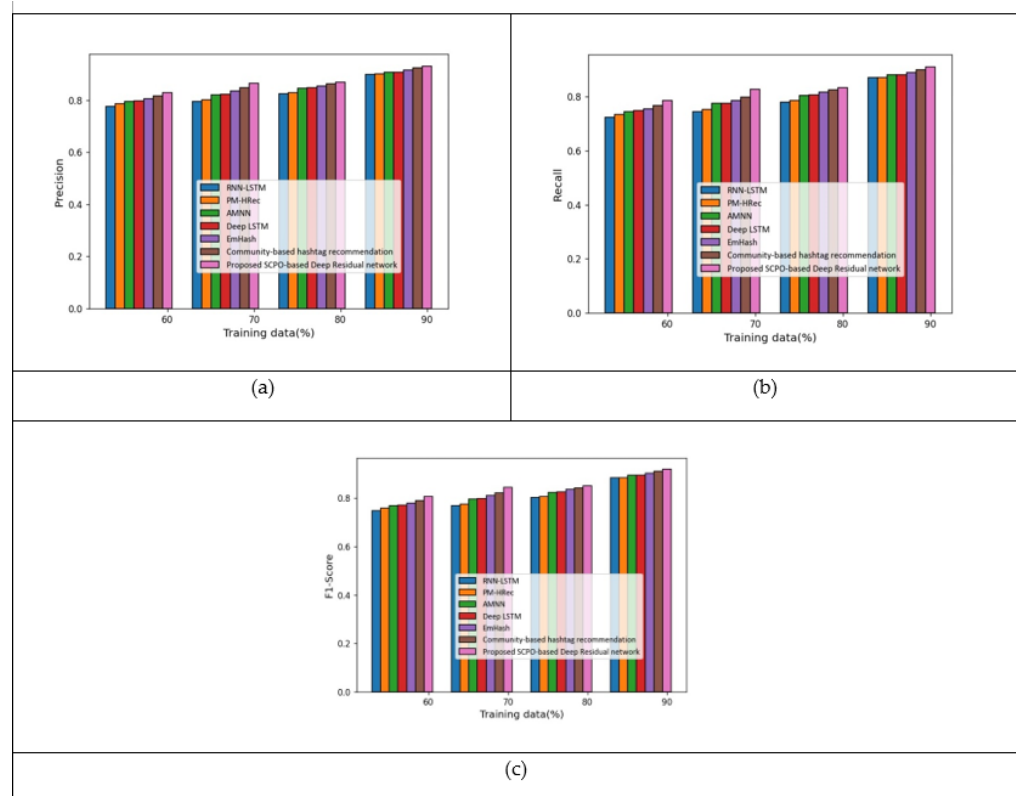


Figure 7. Analysis with features f_1 – f_6 based on Twitter dataset: (a) Precision; (b) Recall; and (c) F1-score.

Figure 8 indicates the comparative analysis of the proposed scheme using the feature f . Figure 8a portrays the analysis based on the precision measure. With 80% of the training data, the precision measured by the proposed SCPO-based Deep Residual network was 0.87, making it 5.56%, 5.18%, 3.29%, 3.05%, 1.94%, and 1.23% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the recall measure is presented in Figure 8b. With 60% of the training data, the recall measured by the proposed SCPO-based Deep Residual network was 0.87, making it 8.92%, 7.68%, 6.21%, 6.12%, 5.04%, and 2.78% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The analysis of the F1-score measure is presented in Figure 8c. With 70% of the training data, the F1-score measured by the proposed SCPO-based Deep Residual network was 0.86, making it 10.09%, 9.04%, 7.08%, 7.04%, 5.81%, and 4.59% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively.

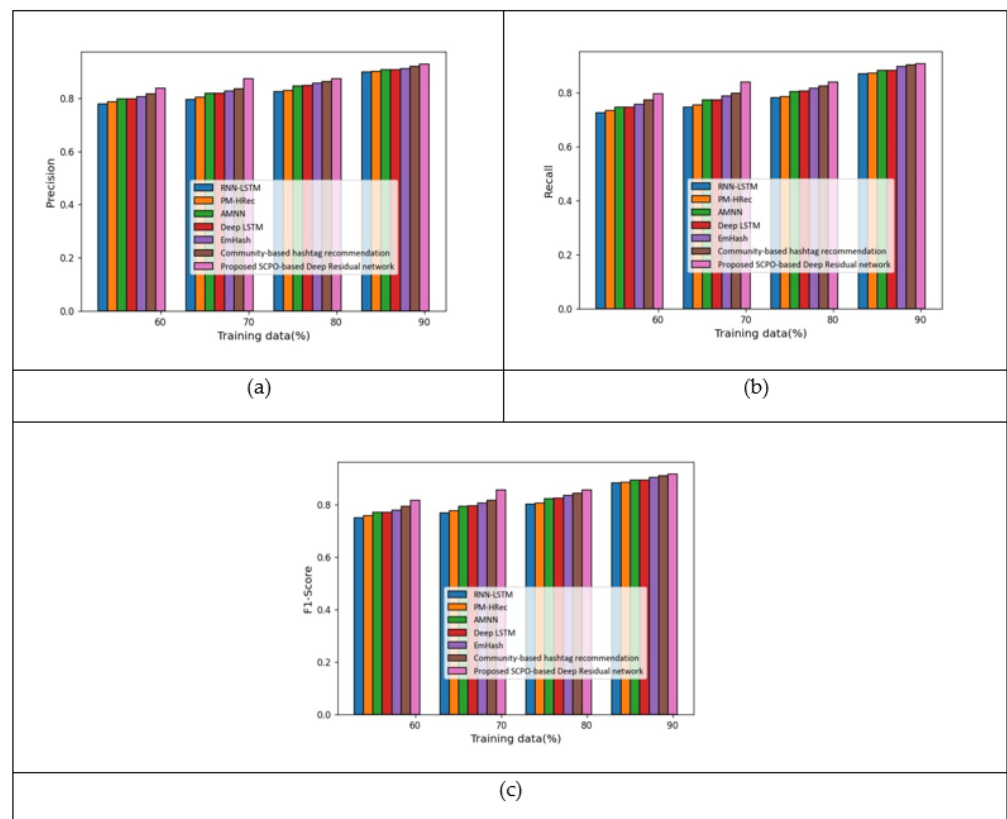


Figure 8. Analysis with *f* features based on Twitter dataset: (a) Precision; (b) Recall; and (c) F1-score.

5.3. Comparative Discussion

Table 2 presents a comparison of the results of the tested methods, showing the values obtained when considering all of the features of the dataset and 90% of the training data. For the Apple twitter sentiment dataset, the precision achieved by the proposed SCPO-based Deep Residual Network was 0.958, making it 3.51%, 3.31%, 2.58%, 2.48%, 1.86%, and 1.55% superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. The recall achieved by the proposed SCPO-based Deep Residual Network was 0.958, making it 4.28%, 4.07%, 3.03%, 2.92%, 2.71%, and 1.57%, superior to RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. Likewise, the F1-score obtained by the proposed SCPO-based Deep Residual Network was 0.963, making it 3.84%, 3.74%, 2.80%, 2.70%, 2.28%, and 1.56% better than RNN-LSTM, PM-HRec, AMNN, Deep LSTM, Emhash, and Community-based hashtag recommendation, respectively. Similarly, the proposed SCPO-based Deep Residual Network had the maximum precision, recall, and F1-score values on the Twitter dataset, compared to the considered state-of-the-art methods.

Table 2. Comparison of existing methods and the proposed SCPO-DRN using Apple twitter sentiment dataset and Twitter dataset, in terms of Precision, Recall, and F1-Score.

Metrics/Method		RNN-LSTM	PM-HRec	AMNN	Deep LSTM	Emhash	Comm. Based	Proposed SCPO-DRN
Apple	Precision	0.934	0.936	0.943	0.944	0.950	0.953	0.968
	Recall	0.917	0.919	0.929	0.930	0.932	0.943	0.958
	F1-score	0.926	0.927	0.936	0.937	0.941	0.948	0.963
Twitter	Precision	0.899	0.900	0.908	0.909	0.913	0.921	0.929
	Recall	0.870	0.871	0.882	0.882	0.898	0.904	0.907
	F1-score	0.884	0.886	0.895	0.895	0.905	0.912	0.918

The proposed method outperformed all of the other comparative methods, in terms of precision, recall, and F1-score. This performance enhancement was achieved through the use of population diversity in the proposed algorithm, which improves the search space, thus achieving better performance enhancement. Furthermore, it is helpful to carry out the optimization of complex problems in parallel, in order to enhance the efficiency and accuracy. The proposed algorithm also has a fast convergence rate and obtained the best solution by avoiding local optima. Overall, the proposed method had improved performance, compared to the other conventional techniques.

6. Conclusions and Future Work

In this research, we developed a Hashtag Recommendation model by employing the SCPO-based Deep Residual Network method, in which a deep learning classifier—that is, the Deep Residual Network—is trained by the proposed SCPO algorithm. The proposed method performs the hashtag recommendation process by considering the features of the keyword set. The input tweet is subjected to stop word removal, effectively removing the stop words associated with the tweet. The keyword set is then computed by finding the dictionary words, thematic words, and more relevant words in the tweet data. The features of the keyword sets are passed to the Deep LSTM classifier, in order to find the hashing score. In addition to the features of the keyword set, the hashing score is also employed as an input to the Deep Residual Network which recommends the hashtags. Empirical results demonstrated that the proposed model can outperform state-of-the-art models on the Apple Twitter Sentiment dataset and Twitter dataset. In the future, the dimensionality of features can be further reduced by employing a feature selection model. Moreover, the training process may be further enhanced by testing the application of other optimization algorithms.

Author Contributions: S.K.B. designed and wrote the paper; H.L. (Hongfei Lin) supervised the work; S.K.B. performed the experiments with an advise from B.X. and H.L. (Haifeng Liu) organized and proofread the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China No. 61572102.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

AMNN	Attention-based Multi-model Neural Network
BoW	Bag of Words
CNN	Convolutional Neural Network
DNN	Deep Neural Networks
LSTM	Long Short-Term Memory
NB	Naive Bayes
NLP	Natural Language Processing
PO	Political Optimizer
RNN	Recurrent Neural Network
SCA	Sine Cosine Algorithm

References

1. Kowald, D.; Pujari, S.C.; Lex, E. Temporal effects on hashtag reuse in twitter: A cognitive-inspired hashtag recommendation approach. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1401–1410.
2. Godin, F.; Slavkovikj, V.; De Neve, W.; Schrauwen, B.; Van de Walle, R. Using topic models for twitter hashtag recommendation. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013; pp. 593–596.
3. Alvari, H. Twitter hashtag recommendation using matrix factorization. *arXiv* **2017**, arXiv:1705.10453.
4. Kabakus, A.T.; Kara, R. “TwitterSpamDetector”: A Spam Detection Framework for Twitter. *Int. J. Knowl. Syst. Sci. (IJKSS)* **2019**, *10*, 1–14. [[CrossRef](#)]
5. Otsuka, E.; Wallace, S.A.; Chiu, D. A hashtag recommendation system for twitter data streams. *Comput. Soc. Netw.* **2016**, *3*, 1–26. [[CrossRef](#)] [[PubMed](#)]
6. Dey, K.; Shrivastava, R.; Kaushik, S.; Subramaniam, L.V. Emtagger: A word embedding based novel method for hashtag recommendation on twitter. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 1025–1032.
7. Cui, A.; Zhang, M.; Liu, Y.; Ma, S.; Zhang, K. Discover breaking events with popular hashtags in twitter. In Proceedings of the 21st ACM international conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 1794–1798.
8. Li, T.; Wu, Y.; Zhang, Y. Twitter hash tag prediction algorithm. In Proceedings on the International Conference on Internet Computing (ICOMP), Las Vegas, NV, USA, 18–21 July 2011.
9. Krestel, R.; Fankhauser, P.; Nejdl, W. Latent dirichlet allocation for tag recommendation. In Proceedings of the Third ACM conference on Recommender Systems, New York, NY, USA, 23–25 October 2009; pp. 61–68.
10. Khabiri, E.; Caverlee, J.; Kamath, K.Y. Predicting semantic annotations on the real-time web. In Proceedings of the 23rd ACM Conference on Hypertext and Social Media, Milwaukee, WI, USA, 25–28 June 2012; pp. 219–228.
11. Weston, J.; Chopra, S.; Adams, K. # tagspace: Semantic embeddings from hashtags. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1822–1827.
12. Cristin, R.; Cyril Raj, V.; Marimuthu, R. Face image forgery detection by weight optimized neural network model. *Multimed. Res.* **2019**, *2*, 19–27.
13. Gangappa, M.; Mai, C.; Sammulal, P. Enhanced Crow Search Optimization Algorithm and Hybrid NN-CNN Classifiers for Classification of Land Cover Images. *Multimed. Res.* **2019**, *2*, 12–22.
14. Vidyadhari, C.; Sandhya, N.; Premchand, P. A semantic word processing using enhanced cat swarm optimization algorithm for automatic text clustering. *Multimed. Res.* **2019**, *2*, 23–32.
15. Ben-Lhachemi, N.; Boumhidi, J. Hashtag Recommender System Based on LSTM Neural Recurrent Network. In Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 28–30 October 2019; pp. 1–6.
16. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [[CrossRef](#)] [[PubMed](#)]
17. Li, Y.; Liu, T.; Hu, J.; Jiang, J. Topical co-attention networks for hashtag recommendation on microblogs. *Neurocomputing* **2019**, *331*, 356–365. [[CrossRef](#)]
18. Ben-Lhachemi, N. Using tweets embeddings for hashtag recommendation in Twitter. *Procedia Comput. Sci.* **2018**, *127*, 7–15. [[CrossRef](#)]
19. Belhadi, A.; Djenouri, Y.; Lin, J.C.W.; Cano, A. A data-driven approach for Twitter hashtag recommendation. *IEEE Access* **2020**, *8*, 79182–79191. [[CrossRef](#)]
20. Alsini, A.; Datta, A.; Huynh, D.Q. On utilizing communities detected from social networks in hashtag recommendation. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 971–982. [[CrossRef](#)]
21. Yang, Q.; Wu, G.; Li, Y.; Li, R.; Gu, X.; Deng, H.; Wu, J. AMNN: Attention-Based Multimodal Neural Network Model for Hashtag Recommendation. *IEEE Trans. Comput. Soc. Syst.* **2020**, *7*, 768–779. [[CrossRef](#)]
22. Ma, R.; Qiu, X.; Zhang, Q.; Hu, X.; Jiang, Y.G.; Huang, X. Co-attention Memory Network for Multimodal Microblog’s Hashtag Recommendation. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 388–400. [[CrossRef](#)]
23. Cao, D.; Miao, L.; Rong, H.; Qin, Z.; Nie, L. Hashtag our stories: Hashtag recommendation for micro-videos via harnessing multiple modalities. *Knowl. Based Syst.* **2020**, *203*, 106114. [[CrossRef](#)]
24. Li, C.; Xu, L.; Yan, M.; Lei, Y. TagDC: A tag recommendation method for software information sites with a combination of deep learning and collaborative filtering. *J. Syst. Softw.* **2020**, *170*, 110783. [[CrossRef](#)]
25. Dhelim, S.; Aung, N.; Ning, H. Mining user interest based on personality-aware hybrid filtering in social networks. *Knowl.-Based Syst.* **2020**, *206*, 106227. [[CrossRef](#)]
26. Majhi, B.; Naidu, D.; Mishra, A.P.; Satapathy, S.C. Improved prediction of daily pan evaporation using Deep-LSTM model. *Neural Comput. Appl.* **2020**, *32*, 7823–7838. [[CrossRef](#)]
27. Chen, Z.; Chen, Y.; Wu, L.; Cheng, S.; Lin, P. Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions. *Energy Convers. Manag.* **2019**, *198*, 111793. [[CrossRef](#)]
28. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]

-
29. Askari, Q.; Younas, I.; Saeed, M. Political optimizer: A novel socio-inspired meta-heuristic for global optimization. *Knowl. Based Syst.* **2020**, *195*, 105709. [[CrossRef](#)]
 30. Kaviani, M.; Rahmani, H. Emhash: Hashtag recommendation using neural network based on bert embedding. In Proceedings of the 2020 6th International Conference on Web Research (ICWR), Tehran, Iran, 22–23 April 2020; pp. 113–118.