Information Sciences 609 (2022) 1506–1517

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Deep learning based hashtag recommendation system for multimedia data

Youcef Djenouri [a], Asma Belhadi [b], Gautam Srivastava [c,e], Jerry Chun-Wei Lin [d,*]

[a] Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[b] Department of Technology, Kristiania University College, Oslo, Norway
[c] Department of Math and Computer Science, Brandon University, Brandon, Canada
[d] Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway
[e] Research Centre for Interneural Computing, China Medical University, Taichung, Taiwan

ARTICLE INFO

Article history:
Received 8 June 2021
Received in revised form 17 July 2022
Accepted 23 July 2022
Available online 28 July 2022

Keywords:
Pattern recommendation
Tweet data
Multimedia and social network
Deep learning

ABSTRACT

This work aims to provide a novel hybrid architecture to suggest appropriate hashtags to a collection of orpheline tweets. The methodology starts with defining the collection of batches used in the convolutional neural network. This methodology is based on frequent pattern extraction methods. The hashtags of the tweets are then learned using the convolution neural network that was applied to the collection of batches of tweets. In addition, a pruning approach should ensure that the learning process proceeds properly by reducing the number of common patterns. Besides, the evolutionary algorithm is involved to extract the optimal parameters of the deep learning model used in the learning process. This is achieved by using a genetic algorithm that learns the hyper-parameters of the deep architecture. The effectiveness of our methodology has been demonstrated in a series of detailed experiments on a set of Twitter archives. From the results of the experiments, it is clear that the proposed method is superior to the baseline methods in terms of efficiency.

1. Introduction

A hashtag is a well-known semantic representation for social media sites like Twitter. Tweets may be able to be divided into two groups based on their origins on Twitter. First, orpheline tweets are those that do not include a hashtag. We may be able to also refer to tweets that contain hashtags as tagged tweets. Hashtags are metadata that are added to the main content to help people quickly identify a message that has a specific theme or substance [1]. The study of hashtags is at the heart of several complicated applications, including query expansion [2], query expansion sentiment analysis [3], and/or smart cities [4]. As a result, hashtag recommendation is critical in hashtag analysis. Hashtag recommendation seeks to find appropriate hashtags for orpheline tweets (tweets without hashtags) from a collection of training non-orpheline tweets. More formally, consider the training non-orpheline tweets collection $\mathscr{T} = \{\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_m\}$ with the hashtags $\mathscr{H} = \{\mathscr{H}_1, \mathscr{H}_2, \ldots, \mathscr{H}_n\}$, every non-orpheline tweet $\mathscr{T}_i$, is composed by hashtags in $\mathscr{H}$. Hashtag recommendation aims to annotate hashtags from the set of hashtags $\mathscr{H}$ by analyzing the set of tweets $\mathscr{T}$.

* Corresponding author.

https://doi.org/10.1016/j.ins.2022.07.132
0020-0255/

1.1. Motivations

There are two types of algorithms that have been proposed for suggesting hashtags: Algorithms that examine the links between tweets, and algorithms that mine for patterns [2,3,5]. Some methods use inference to recommend hashtags for a new tweet after using deep learning architectures to identify distinct patterns and behaviours from a set of tweets [6–10]. In comparison to deep learning solutions, pattern mining techniques demand a lot of computing time, according to our analysis of state-of-the-art hashtag recommendation algorithms. Pattern mining methods, on the other hand, are far more accurate than deep learning solutions. This is because pattern mining systems need a long time to investigate the various dependencies and correlations among the tweets collection. Unlike deep learning solutions, which simply require a simple propagation to indicate the hashtags of a new tweet, deep learning solutions require no propagation at all. However, an enormous problem is that deep learning requires a high number of hyper-parameters that need to be tuned to achieve high accuracy performance. In this paper, we propose a hybrid solution of pattern mining and deep learning that combines the advantages of pattern mining and deep learning. Our goal is to improve the accuracy of deep learning solutions, reduce the computation time required for pattern mining solutions, and improve the accuracy of hashtag recommendation results. In addition, to optimize the numerous hyper-parameters of the deep learning architecture, an evolutionary algorithm based on the genetic process is included for hashtag recommendation.

1.2. Contributions

This paper proposes a novel technique to hashtag recommendations. Before deep learning is used to discover the hashtags associated with the tweets, the approach analyzes the numerous correlations that exist between the tweets in the collection. The following are the primary contributions of the presented work:

1. We offer a novel method for recommending orpheline tweet hashtags. Pattern mining, and deep learning, are both explored in this method. Pattern mining is used to analyze the numerous relationships in the collection of tweets to identify the different batches of the deep neural network. This is done to determine the batches. Additionally, a convolutional neural network is used to learn the numerous hashtags included in the tweet collection.
2. To improve the suggested solution, we present a pruning technique. As a result, the pattern mining procedure incorporates a coverage technique to reduce irrelevant patterns.
3. By studying and adapting the many hyper-parameters of the deep learning model used in the learning process, we provide an evolutionary algorithm for hashtag recommendations based on the genetic process.
4. We conduct extensive research and testing on a variety of Twitter collections. The studies have shown that the proposed solution outperforms the baseline algorithms in terms of both runtime and accuracy.

2. Related work

Two different approaches can be used to solve the problem of hashtag suggestions [11–15]. Traditional solutions promote appropriate hashtags using traditional data mining, and machine learning methodologies, while solutions for hashtag recommendations based on superior technology use more advanced deep learning architectures to make predictions about relevant hashtags. We will showcase pertinent connected works to both categories in the following sections. Zhao et al. [16] developed a technique for personalized hashtag recommendations based on Latent Dirichlet Association (LDA) user profiling. This technique first determines the frequency of all hashtags used by the top-k comparable users and then recommends the hashtags that are most relevant to that user. To assess user profiles, and determine the set of tailored hashtags, Li et al. [17] recommended using the probabilistic latent factor model and content information. The users with micro-topic latent factors are estimated first. The best micro-topics are then obtained for each user by fitting the distribution of the previously derived models.

Gong et al. [18] disseminate a model distributed that generates new information. The algorithm may consider both textual and visual information when making recommendations for hashtags to use. To decipher hidden topics, they use a Gibbs model to sample the data. Kou et al. [19] introduced a hashtag recommendation system using multiple features in microblogs. The system considers hashtags of friends from different microblogs as candidates. HRMF is used to determine the candidate's score. Liu et al. [20] developed a Hashtag2Vec model that takes advantage of hierarchical relationships such as hashtag/hashtag, hashtag/tweet, tweet/word, and word/word. This contributes to a better understanding of the semantic context of tweets that have been tagged. Then, they use a content-based embedding system to facilitate the derivation of network embeddings. Belhadi et al. [3] developed a new pattern mining algorithm called PM-HRec (Pattern Mining for Hashtag Recommendation). It consists of two main stages. Concerning the temporal information of the tagged tweets, offline processing first converts the corpus of tweets into a transactional database (tweets with hashtags). The technique identifies the top k high-average-utility temporal patterns. Offline construction is also performed for irrelevant tags and tag ontology. Second, to extract the most relevant hashtags for a given tweet, the utility patterns, ontology, and irrelevant tagged tweets (tweets without hashtags) are input during online processing.

Looking at the algorithms developed so far, we notice that in the learning phase they tend to neglect correlations that may exist between tweets. This severely compromises the quality of the mechanism used to suggest hashtags. In this work, we

took inspiration from the PM-HRec technique [3] and investigated and explored the relevant connections between the tagged tweets. Instead of using the frequent patterns directly as in PM-HRec, we introduce a new learning model that uses new pattern mining models to find the stacks of learning models that should be used to solve the major challenges of the hashtag recommendation problem accurately and efficiently. This model is novel because it uses these new pattern mining models to find the stacks of learning models.

3. DPM-HR based framework

3.1. Principle

This section introduces the DPM-HR framework (Deep learning Pattern Mining for Hashtag Recommendation). The goal is to accurately categorize the set of tweets so that suitable hashtags may be able to be recommended. It finds a set of hashtags associated with tweets about orpheline using deep learning and pattern mining. The method begins by transforming the tweet videos into multiple images, as seen in Fig. 1. After that, each image is subjected to the SIFT feature extractor, resulting in the feature database. To extract the important patterns, the features database is turned into a transaction database. The set of common patterns is utilized to build batches that are used as deep learning input. Deep learning is applied to previously prepared batches of tweets with a set of hashtags as outputs. The goal is to learn the network's weights so that the set of relevant hashtags may be able to be properly predicted from orpheline tweets. The genetic algorithm is also injected during the training process to find the optimal hyper-parameters of the deep learning model. In the rest of this section, we will show you how to use each of these terms in the context of DPM-HR.

3.2. Pattern mining

We investigate pattern mining to study the correlation between the set of tweets to obtain the best features for creating the batches of the tweets. It pulls the most relevant data from as many sources as possible. The term "frequent pattern mining" refers to the process of extracting relevant patterns from the transaction database that satisfy the minimum support threshold (*minsup*). We use the classical pattern mining approach in this study to quickly discover the best features of the training tweets. The pruning method represents a significant departure from previous pattern mining algorithms. While existing algorithms identify all patterns that exceed the bounds of minimal support, our approach filters out irrelevant patterns based on additional criteria, such as the collection of subsets of relevant patterns that covers the largest number of tweets. SSFIM [21] is a new approach for detecting common patterns in a single pass. It is a non-sensitive algorithm for determining the minimum support threshold. The SSFIM outperformed state-of-the-art pattern mining algorithms, according to the findings of the experiments. Therefore, SSFIM is used in this work to develop a model for detecting common pat-

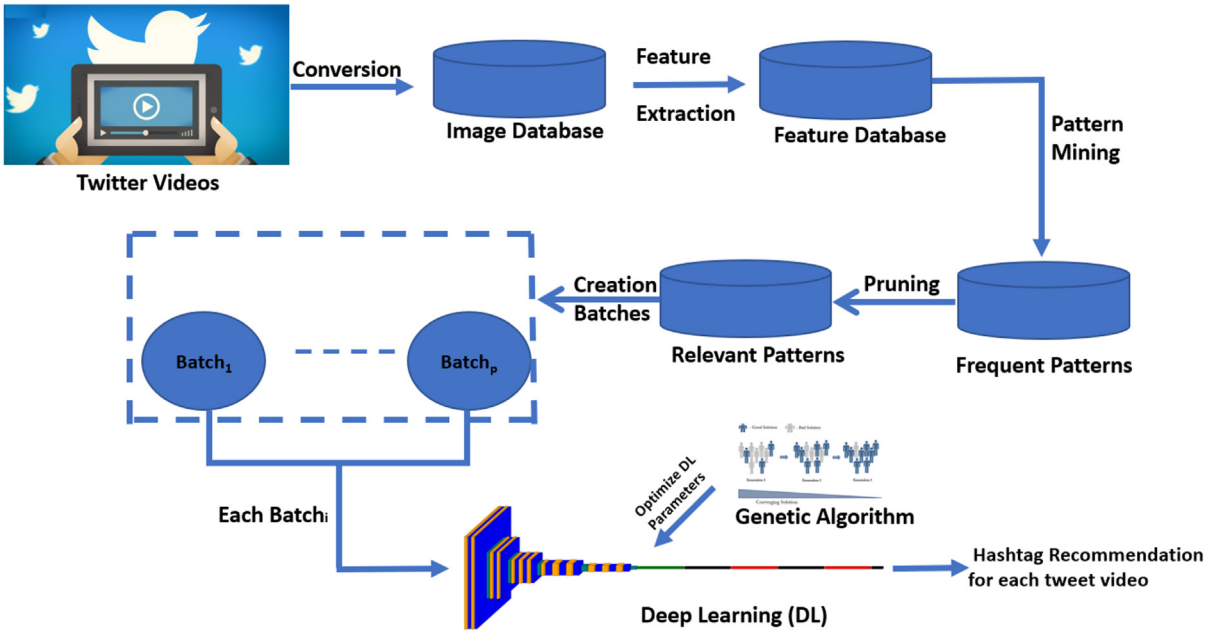

Fig. 1. DPM-HR Framework: each image collected from the tweeted video is subjected to the SIFT feature extractor, resulting in the feature database. Then, the pattern recognition process is introduced to build stacks. Deep learning with the genetic algorithm is applied to the stacks. The final output is the set of recommended hashtags for each tweeted video.

terns in a group of tweets. Tweet by tweet, the collection of tweets is scanned. When all possible combinations of patterns have been formed for each tweet, the hash table reflecting the frequency of all patterns is recursively updated. This method is repeated for each tweet and results in the identification of all patterns that exceed the minimum support threshold.

The drawback of generic pattern mining is that it uncovers a huge number of common patterns, resulting in slowness when dealing with big collections of tweets. Analyzing a large number of found patterns is time-consuming. To overcome this constraint, a new technique is proposed for filtering the mined frequent patterns in the mining process, resulting in the discovery of a small number of significant patterns that may be used to sketch the collection of training tweets. We utilize a novel idea called **Coverage** in the suggested pruning technique to maintain fewer and more relevant patterns based on the minimal description length principle [22] to cover the most number of tweets from a training tweets collection. It is possible to considerably minimize the number of frequent patterns. The developed model's identified patterns are not the same as the maximal [23] or closed [24] frequent patterns. Below are more thorough explanations of the potential solutions.

Definition 1. Let $S = \{S_1, S_2, \ldots, S_r\}$ be the set of the discovered frequent patterns in the mining progress. The *coverage pruning problem* is defined by maximizing $Pruning_{max}$ as:

$$\begin{array}{rcccc} Pruning_{max} & : & S & \rightarrow & \mathfrak{R} \\ & & S' & \mapsto & Pruning_{max}(S'). \end{array} \tag{1}$$

Definition 2. $Pruning_{max}$ is defined as a function that may be able to be used to obtain the maximum number of tweets from a given collection of training tweets. Let $\mathscr{T}(S_i)$ be the set of tweets that comprises a sample S_i. The goal of coverage pruning is to generate a collection of subsets $S' \subset S$ that optimizes the coverage value as:

$$\begin{array}{rcccc} Pruning_{max} & : & S & \rightarrow & \mathfrak{R}^{+} \\ & & S' & \mapsto & |\bigcup\limits_{S_i \in S'} \mathscr{T}(S_i)|. \end{array} \tag{2}$$

Definition 3. Finding the minimum collection of subsets $S^* \subset S$ is an optimal solution to the coverage pruning problem in a tweets collection that includes m tweets. Here, S^* covers all the tweets and is then defined as follows:

$$\begin{cases} Pruning_{max}(S^*) = m \\ \forall S' \subset S, \\ Pruning_{max}(S') = Pruning_{max}(S^*) \Rightarrow |S'| \geqslant |S^*|. \end{cases} \tag{3}$$

Given that a frequent collection of S-patterns may be able to be chosen from any of the 2^r possible S-collection of subsets, finding the best collection of subsets that satisfies the coverage pruning constraints is a *NP*-complete problem. Therefore, a complete search would be very time-consuming, when the cardinality of S is very large. To solve this problem, a greedy search may be able to be paired with an adjacent search to restrict the search space by providing an acceptable answer instead of an ideal answer globally.

We were motivated by the work of Hosseini et al. [25] who enumerated the search tree using the greedy method, performing local searches on each and every node that was produced. Initially, the set of frequent patterns S, the maximum number of iterations, and the number of tweets in the supplied collection of training tweets are considered, resulting in the set of patterns S^*. In the first step, common patterns are randomly selected from S. The answer is then stored in an S^* variable, which represents the optimal solution at the moment. Then an iterative procedure is performed to improve the current solution.

This process continues until either the number of tweets S^* covering the patterns is less than m or the number of iterations is less than the maximum number of iterations. To better enhance the discovered solutions, it is necessary to determine the *neighbors* of the solutions. All possible solutions that can be achieved by combining the current answer with one or more additional common patterns are generated here. If and only if it is better than the best solution $S*$ in the current phase, the pruning function sets the variable $S*$ to the value *best*. Otherwise, the variable remains unset. Note that the best solution among these solutions is denoted by the value *best*. It is worth noting that when two solutions, such as sol_1, and sol_2, are used, hold the condition as $Pruning_{max}(sol_1) \leqslant Pruning_{max}(sol_2)$, and $|sol_1| \leqslant |sol_2|$, then the solution proposed in sol_1 is judged to be better than the one proposed in sol_2. This is because there should be as few patterns as possible. First, a greedy model is provided to create a collection with the least number of common patterns that covers the maximum number of tweets. This is achieved by maximizing the number of tweets covered by the common patterns. Other pruning functions, it should be emphasized, may be able to be employed for other purposes. A series of relevant patterns is revealed at the end of this result. These patterns are used to create a set of batches, with each batch having a collection of subsets of the tweets from the training tweets collection that are covered by the provided pattern. Let us formally consider the set of relevant patterns S^* extracted by the patterns mining algorithm and the pruning strategy. Each pattern S_i^* in S^* is compared to each tweet $\mathscr{T}_j$ in $\mathscr{T}$. If S_i^* contains all hashtags from $\mathscr{T}_j$, then $\mathscr{T}_j$ is added to the B_i batch. This process is repeated for all tweets in $\mathscr{T}$. At the end of this step, a set of batches is created, where each batch represents a particular pattern in S^*.

3.3. Deep learning

The deep learning algorithm learns the collection of hashtags using the batches created in the previous stage. The list of hashtags was learned using a convolution neural network (CNN). Convolutional layers are used in CNN to extract features depending on the location of reference in images. On the input features, our network applies 32 filters with window sizes of 3, 5, and 7 in parallel. The feature maps produced by the three convolutional blocks are concatenated and fed sequentially into the hidden fully connected layer (FC), which consists of neurons with values of 1024and 256, respectively. The FC layer is followed by the softmax layer, which receives the output of the FC layer. Dropout is used in both FC layers to mitigate the effects of overfitting.

3.4. Genetic algorithm

Before attempting to create the network, you must first set the following parameters: the number of epochs, the learning rate, the activation functions of each layer, and the dropout rate. As a result, an intelligent technique for determining these parameters is required. We employ a genetic algorithm to identify the best parameters of the learning architecture to do so quickly. The developed genetic algorithm aims to intelligently explore the solution space of the possible configuration of the parameters of the deep learning architecture used in our proposal. This is used only once during the training process. When the hashtags of the new tweets are recommended, the model with the best parameters is used. The population set is initialized, with each individual represented by a set of values for each learning architecture parameter. After then, the crossover, mutation, and selection operators are used to generate more relevant individuals from the current population. This procedure is repeated for as many generations as possible.

3.5. Pseudo code

Algorithm 1 presents the pseudo-code of the DPM-HR algorithm. The process starts by transforming the tweets collection into the transaction database. We can for all tweets, for each hashtag in the given tweet, associate its value to 1, in its corresponding transaction. The single scan frequent itemset mining technique is used to extract meaningful patterns from a batch of transactions. To narrow the pattern space and only provide the most pertinent patterns, a coverage metric is also used. By calling the *CreatingBatches*() method, these patterns are used to create batches of tweets. The convolution, as well as max-pooling operators, are used to define the Deep Learning model. Then, the genetic algorithm is used to improve the hyperparameters of the deep learning model by training with batches of tweets. Finally, the trained model is used to suggest appropriate hashtags for the new tweet.

Algorithm 1 DPM-HR Algorithm

Input: $\mathscr{T} = \{\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_m\}$: the set of tweets.
$\mathscr{H} = \{\mathscr{H}_1, \mathscr{H}_2, \ldots, \mathscr{H}_n\}$: the set of hashtags.
minsup: minimum support threshold.
Output: $< H_{new} >$: the set of the relevant hashtags of the new tweet $\mathscr{T}_{new}$.
**********Pattern Mining**********
for i = 1 to m
  for j = 1 to n **do**
    if $\mathscr{H}_j \in \mathscr{T}_i$ **do**
      $D[i][j] \leftarrow 1$; **then**
    else
      $D[i][j] \leftarrow 0$;
    end if
  end for
end for
$\mathscr{P} \leftarrow SSFIM(D, \mathscr{H}, minsup)$;
$\mathscr{P} \leftarrow Pruning(D, \mathscr{H}, \mathscr{P})$;
**********Deep Learning**********
Batches $\leftarrow$ *CreatingBatches*($\mathscr{P}$);
model $\leftarrow$ VGG16();
Hyper_Param $\leftarrow$ GA(fit(model, Batches));
$\mathscr{H}_{new} \leftarrow predict(\mathscr{T}_{new}, model, Hyper\_Param)$;
return $\mathscr{H}_{new}$.

4. Performance evaluation

4.1. Experimental environment

Using benchmark hashtag recommendation collections, extensive tests were conducted to evaluate the performance of the proposed technique. The Keras deep learning package was used to implement all of the algorithms in Python 3.7. For pattern mining, we additionally use an itemset-mining library. To accurately assess recommended hashtags, computation time and accuracy are calculated. The run time is measured in seconds, and the accuracy is determined by *hit_ratemeasure*, which is defined as:

$$hit\_rate = \frac{\sum_{\mathscr{T}_i \in \mathscr{T}_{test}} Correct(\mathscr{T}_i)}{|\mathscr{T}_{test}|}, \tag{4}$$

where $Correct(\mathscr{T}_i)$ is set to 1 if and only if the set of the recommended hashtag of $\mathscr{T}_i$ contains the standard hashtags of $\mathscr{T}_i$. Otherwise, its value is 0.

In the experiments, several collections of tweets are employed (see Table 1 for more details). These databases range in size from tiny to large, sparse to dense. As a result, some tweet collections have a large number of tweets, others have a large number of hashtags, and yet others have a large number of tweets plus hashtags. PM-HRec [3] and GCN-PHR [9] were employed as baseline algorithms in these tests. The first approach employs pattern mining to recommend appropriate hashtags, whereas the second approach learns hashtags from a tweet collection using a graph convolution neural network.

4.2. DPM-HR analysis

This first experiment aims to analyze the performance of DPM-HR with and without using the pruning strategy, which is used to reduce the number of relevant patterns in the pattern mining process. In other words, the goal is to show the usefulness of the pruning strategy in the context of DPM-HR. Table 2 shows the accuracy of DPM-HR, determined by the hit_rate, with and without applying the pruning strategy of the pattern mining process. The results show a clear superiority of DPM-HR with the pruning strategy and this is independent of the corpus used during the training process. For example, when processing the Nelson Mandela corpus, the hit rate of DPM-HR with a pruning strategy is 82%, while the hit rate of DPM-HR without a pruning strategy is only 78%. These results are explained by the fact that the pruning strategy efficiently reduces the number of patterns, inferring only the relevant ones. This helps to create homogeneous batches that are used for the training process.

The second experiment aims to analyze the performance of DPM-HR with and without using the genetic algorithm used to find the best parameters of the deep learning architecture. In other words, the goal here is to show the usefulness of the genetic algorithm in the context of DPM-HR. Table 3 shows the accuracy of DPM-HR, determined by the hit_rate, with and without applying the genetic algorithm in the hyperparameter optimization step. The results show a clear superiority of DPM-HR with the genetic algorithm, regardless of the data collection used in the experiment. These results are explained by the fact that the genetic algorithm intelligently extends the parameter space of the deep learning architecture by providing crossover and mutation operators that allow both intensification and diversification of the search.

4.3. DPM-HR vs state-of-the-art solutions

The extensive experiments are then performed to show the performance between the designed DPM-HR and the two baseline approaches (e.g., PM-HRec and GCN-PHR) regarding the solutions of the hashtag recommendation under different tweet datasets. The reason to select those two approaches as the baseline models for comparison is that it is based on a pattern mining mechanism, and the GCCN-PHR is based on the convolution neural network; the designed DPM-HR combines the advantages of both the two algorithms. The run time of DPM-HR is compared to that of basic hashtag recommendation algorithms in Fig. 2. The *X*-axis indicates the percentage of total tweets used in each test, while the *Y*-axis indicates the dura-

Table 1
Tweets Collection Description.

Corpus	# Hashtags	# Tweets
Wikepedia1	13,156	81,270
Football	90,660	3,000,000
Nelson Mandela	50,425	2,813,461
Wikepedia3	32,280	168,199
TREC2015	66,384	250,306
Sewol ferry	723	239,117
Wikepedia2	19,124	86,929
TREC2011	106,682	333,491

Table 2
Analysis of the accuracy of DPM-HR with and without pruning strategy and with different collection of tweets.

Corpus	DPM-HR with pruning strategy	DPM-HR without pruning strategy
Wikepedia1	84	82
Football	85	83
Nelson Mandela	82	78
Wikepedia3	91	89
TREC2015	85	84
Sewol ferry	84	81
Wikepedia2	87	86
TREC2011	83	82

Table 3
Analysis of the accuracy of DPM-HR with and without genetic algorithm and with different collection of tweets.

Corpus	DPM-HR with pruning strategy	DPM-HR without pruning strategy
Wikepedia1	84	80
Football	85	84
Nelson Mandela	82	79
Wikepedia3	91	87
TREC2015	85	83
Sewol ferry	84	77
Wikepedia2	87	83
TREC2011	83	80

tion of the test in seconds. The results show that DPM-HR beats PM-HRec while it is comparable to GCN-PHR. When the number of tweets is changed from 20% to 100%, the proposed method runs faster than the two baseline techniques, indicating a clear advantage over PM-HRec. For example, if 100% of the tweets in the soccer collection are considered, PM-HRec takes more than 180 s to execute, but DPM-HR takes much less than 80 s to execute. In addition, DPM-HR performs better than GCN-PHR in five collections of tweets, but GCN-PHR performs better than DPM-HR in three other collections.

The accuracy of DPM-HR is compared to that of baseline hashtag recommendation systems using different tweet sets in Fig. 3. The *X*-axis shows the percentage of total tweets used for each test, and the *Y*-axis shows the calculated hit rate for each test. The results show that DPM-HR outperforms the GCN-PHR algorithm and compares very well with PM-HRec. When increasing the number of tweets from 20% to 100%, the proposed method achieves a higher hit rate than the two baseline techniques, showing a significant advantage over GCN-PHR. For example, with 100% of tweets in the *Sewol ferry* collection, the hit rate of GCN-PHR is only 75%, while the hit rate of DPM-HR is up to 85%. Moreover, DPM-HR outperforms PM-HRec for every tweet collection tested. These results are based on a variety of factors, including the following:

1. The utilization of batches of tweets produced from the tweet collections' most relevant patterns.
2. The use of a genetic algorithm to determine the optimal deep learning architecture hyper-parameters.

4.4. Performance on big tweets collection

Fig. 4 illustrates the run time in seconds, with the hit rate indicating the quality of the solution based on a $40,000,000$ tweet corpus. Note that the genetic algorithm is used only once during the training process. When the hashtags of the new tweets are recommended, the model with the best parameters is used. From these results, we may be able to infer that solutions based on pattern mining require more computation time than solutions based on deep learning. Pattern mining methods, on the other hand, are far more accurate than deep learning solutions. Furthermore, in terms of run time and accuracy, our solution surpasses both alternatives. These results may be able to be explained by the fact that while pattern mining systems spend a lot of time analyzing the many dependencies and correlations within the tweet collection, deep learning solutions only need a simple propagation to identify the hashtags associated with a new tweet. However, the effective integration of pattern mining and deep learning, and the various methods represented by the pruning strategy and hyperparameter optimization, all contribute to our approach's ability to improve accuracy while reducing the computation time required to solve the hashtag recommendation problem.

4.5. Validation process

A statistical test called the *Z*-test was performed on the data using the tweet collections described above to formally validate the superiority of the entire suggested solution (including pattern mining, and deep learning) over the baseline solutions. The *Z*-test is simulated as follows:

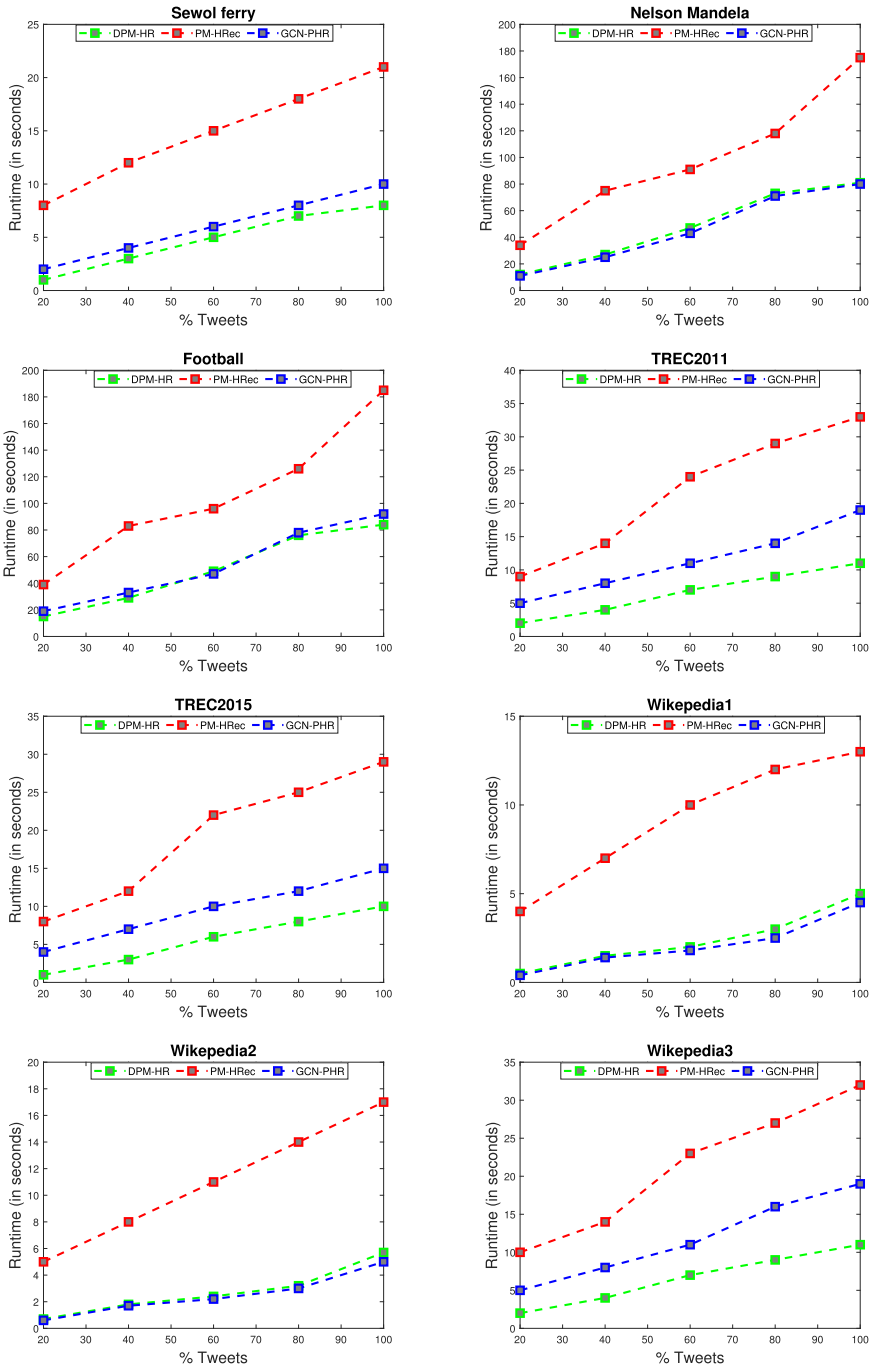

Fig. 2. Runtime performance.

1. The values of run time, and accuracy, are treated as regular variables.
2. Each Twitter collection is separated into five divisions, each and every containing 20% of the total number of tweets. Each, and every split, is regarded as a separate observation. As a result, 40 various observations are created.
3. The result of using the hashtag recommendation approach on each partition is a sample.

In this study, we will develop four different estimators, numbered E_1 to E_4. The runtime performance of the first two estimators is the responsibility of the first two estimators, while the accuracy performance of the second two estimators is the responsibility of the second two estimators. The following is a comprehensive explanation of the definitions of the four estimators:

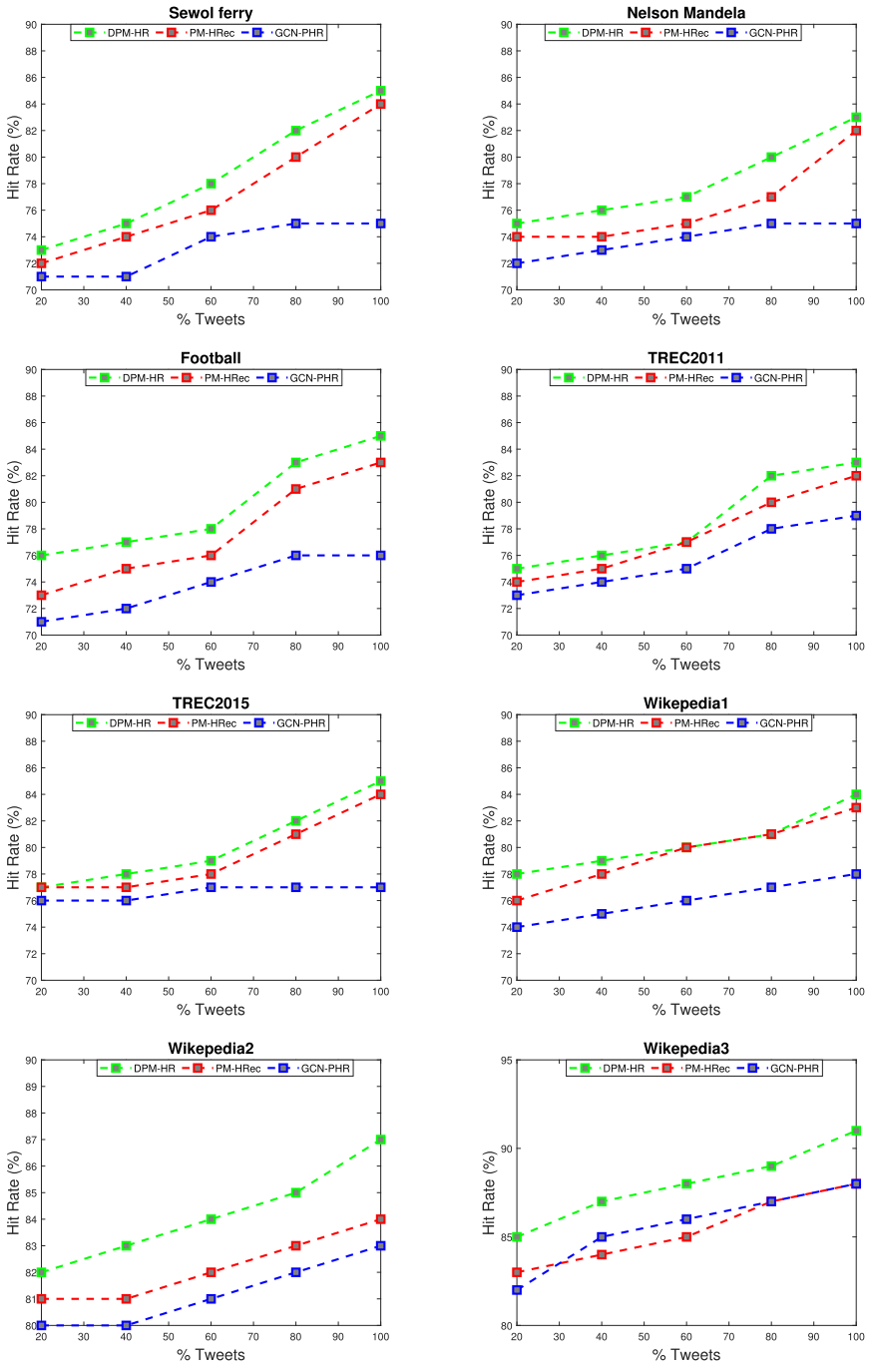

Fig. 3. Accuracy performance.

$$\begin{cases} E_1 = CPU(DPM) - Fmeasure(PM), \\ E_2 = E_1 - CPU(GCN), \\ E_3 = Accuracy(DPM) - Accuracy(PM), \\ E_4 = E_3 - MAP(GCN), \end{cases} \tag{5}$$

where $CPU(A)$ is the average of the run time values of the algorithm A in the 40 observations; $Accuracy(A)$ is the average of the accuracy values of the algorithm A in the 40 observations, and A. The algorithm belongs to the set {DPM-HR (DPM for short), PM-HRec (PM for short), GCN-PHR (GCN for short)}.

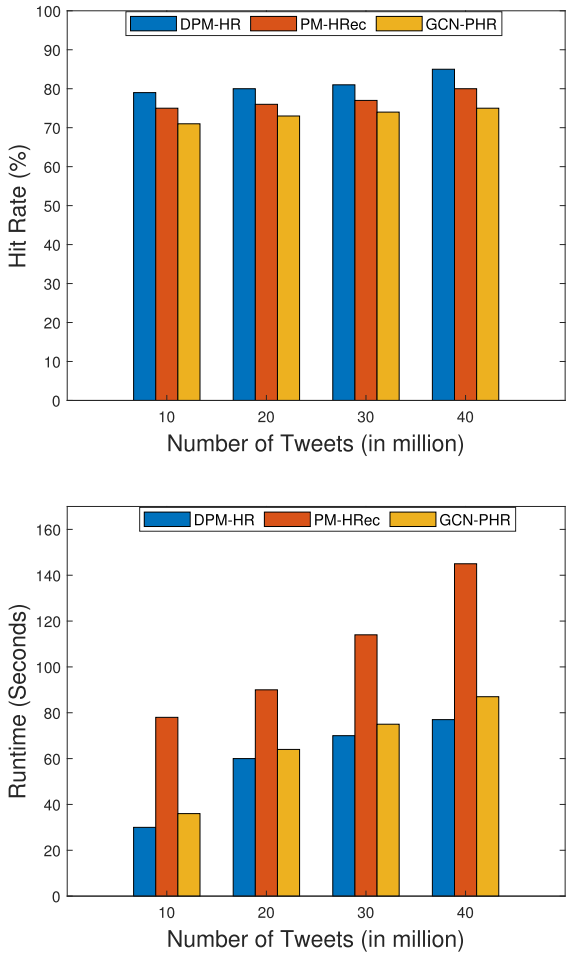

Fig. 4. Performance analysis on big tweets collection.

First, the normality of the three algorithms is checked using the Shapiro–Wilk test, which is included in the XLSTAT program. The following definitions apply to the initial hypothesis H_0 and the alternative hypothesis H_α : H_0 means that the algorithms follow a normal distribution; H_α means that the methods deviate from the normal distribution. The significance threshold used (α) was set at 2%. The results of the Shapiro–Wilk test show that H_0 cannot be rejected. This shows that the non-normality is not significant, i.e., the algorithms follow the normal distribution. The Z-test is then conducted to compare the approaches when the *alpha* is set as 2% in the experiments. According to XLSTAT, E_1 and E_3 can thus produce good values than the other estimators, which showed that the developed DPM-HR has achieved good performance and results than the other two models regarding the runtime and accuracy performance.

5. Discussions and future perspectives

Here, we discuss and explain the primary findings of the designed model in this paper, followed by the future prospects of the developed approach. The primary funding source for the combination of pattern mining and deep learning is as follows:

1. Correlation analysis of the collection of tweets is beneficial to the entire proposal process. This allows us to generate homogeneous batches that are beneficial to the overall learning process.
2. The homogeneous batches enable us to precisely learn the weights of the different layers of the convolution neural network. In this way, weight propagation is performed efficiently. This method avoids overfitting and improves accuracy.
3. The improvement of this paper permits the entire process to be progressed on both sides, pattern mining and deep learning. The pruning technique, which involves removing unnecessary patterns based on the coverage measure, enables us to precisely locate the batches required in the deep learning step. Furthermore, the hashtag recommendation problem's hyper-parameters optimization aims to accurately and rapidly achieve the ideal condition of the network, improving both the calculated accuracy and run time.

Motivated by the success of our approach to different tweets collection, several directions may be investigated in the future:

1. **Improving the pattern mining step.** Pattern mining is often used to study the correlation among the tweets collection. Additional techniques may be able to be used for identifying the batches of the convolution neural network. In the future, it might be interesting to combine many alternative pattern mining approaches with the one described here. For instance, it could include high utility pattern mining [26], hidden sensitive patterns [27], closed pattern mining [28–30], or sequential pattern mining [31,32] can be further investigated and discussed.
2. **Improving the recommendation step.** The recommendation step may be able to be improved which in turn will increase the overall performance of the methodology using known tools such as supercomputers frameworks based on cluster computing. If and only if we may be able to create jobs that are independent of each batch of tweets, then overall there are techniques that may be able to improve the perforation of the recommendation step. In this context, the voting mechanism [33,34] is needed to select the best outputs retrieved from the different jobs.
3. **Explainable AI (XAI).** It is also called interpretable AI or explainable machine learning (XML) and is a subset of artificial intelligence where the output of the solution is understandable to humans. It contrasts with the "black box" concept in machine learning, where even the creators of the AI cannot explain why it made a particular choice. Thus, since it is not easy to explore XAI to understand the process of updating the weights with the genetic algorithm, it is possible to improve the DPM-HR with the XAI concept shortly.
4. **Case studies.** Further investigation of various case studies outside of the ones presented in this paper will be vital to the enhancements of the methodicalness of this paper. Looking at domain-specific issues in big data analysis, and discovering the findings of this paper to become feasible in other domains will show the true strength of the results garnered here. Specifically, when analyzing vehicular technology use cases, runtime complexity becomes critical components of any framework. Furthermore, in any shares/stocks environment, the overall run time will be critical to maintaining usability during volatility in the markets.

6. Conclusion

This research presents a new hybrid architecture for recommending appropriate hashtags for orpheline tweets. Pattern mining and deep learning are both used in this method. It begins by selecting the batch size for the convolution neural network using frequent pattern mining algorithms. The hashtags of the tweets are then learned using the convolution neural network that was applied to the collection of batches of tweets. A pruning approach for performing the learning process accurately by removing irrelevant frequent patterns is also proposed. In addition, the evolutionary method is utilized to extract the best parameters for the deep learning model that is used in the learning process. This is accomplished by employing a genetic algorithm to learn the deep architecture's hyper-parameters. The effectiveness of our methodology has been demonstrated through a series of detailed experiments on a set of Twitter archives. The results of the experiments show that the proposed method is superior to the baseline methods in terms of both runtime and accuracy.

CRediT authorship contribution statement

Youcef Djenouri: Conceptualization, Writing – original draft, Methodology. **Asma Belhadi:** Validation, Formal analysis. **Gautam Srivastava:** Investigation, Writing – review & editing. **Jerry Chun-Wei Lin:** Investigation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A. Belhadi, Y. Djenouri, J.C.-W. Lin, C. Zhang, A. Cano, Exploring pattern mining algorithms for hashtag retrieval problem, IEEE Access 8 (2020) 10569–10583.
[2] N. Kumar, E. Baskaran, A. Konjengbam, M. Singh, Hashtag recommendation for short social media texts using word-embeddings and external knowledge, Knowl. Inf. Syst. (2020) 1–24.
[3] A. Belhadi, Y. Djenouri, J.C.-W. Lin, A. Cano, A data-driven approach for twitter hashtag recommendation, IEEE Access 8 (2020) 79182–79191.
[4] Y. Liu, D. Liu, Y. Chen, Research on sentiment tendency and evolution of public opinions in social networks of smart city, Complexity (2020).
[5] G. Du, J. Sun, X. Huang, J. Yu, Mining user interests for personalized tweet recommendation on map-reduce framework, in: Proceedings of the International Conference on Enterprise Information Systems, 2017, pp. 201–208.
[6] D. Cao, L. Miao, H. Rong, Z. Qin, L. Nie, Hashtag our stories: Hashtag recommendation for micro-videos via harnessing multiple modalities, Knowl.-Based Syst. 106114 (2020).
[7] K. Ota, M.S. Dao, V. Mezaris, F.G.D. Natale, Deep learning for mobile multimedia: A survey, ACM Trans. Multimedia Comput. Commun. Appl. 13 (3s) (2017) 1–22.
[8] M. Li, T. Gan, M. Liu, Z. Cheng, J. Yin, L. Nie, Long-tail hashtag recommendation for micro-videos with graph convolutional network, in: ACM International Conference on Information and Knowledge Management, 2019, pp. 509–518.

[9] Y. Wei, Z. Cheng, X. Yu, Z. Zhao, L. Zhu, L. Nie, Personalized hashtag recommendation for micro-videos, ACM International Conference on Multimedia (2019) 1446–1454.
[10] Q. Yang, G. Wu, Y. Li, R. Li, X. Gu, H. Deng, J. Wu, Amnn: Attention-based multimodal neural network model for hashtag recommendation, IEEE Trans. Comput. Soc. Syst. (2020).
[11] B. Shi, G. Poghosyan, G. Ifrim, N. Hurley, Hashtagger+: Efficient high-coverage social tagging of streaming news, IEEE Trans. Knowl. Data Eng. 30 (1) (2018) 43–58.
[12] R. Makki, E. Carvalho, A.J. Soto, S. Brooks, M.C.F.D. Oliveira, E. Milios, R. Minghim, ATR-Vis: Visual and interactive information retrieval for parliamentary discussions in Twitter, ACM Trans. Knowl. Discovery Data 12 (1) (2018) 3.
[13] S. Sedhai, A. Sun, Hashtag recommendation for hyperlinked tweets, in: International ACM SIGIR Conference on Research & Development in Information Retrieval, 2014, pp. 831–834.
[14] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, R. Van de Walle, Using topic models for twitter hashtag recommendation, in: International Conference on World Wide Web, 2013, pp. 593–596.
[15] S. Zhang, H. Cheng, Exploiting context graph attention for poi recommendation in location-based social networks, in: International Conference on Database Systems for Advanced Applications, 2018, pp. 83–99.
[16] F. Zhao, Y. Zhu, H. Jin, L.T. Yang, A personalized hashtag recommendation approach using lda-based topic model in microblog environment, Future Gener. Comput. Syst. 65 (2016) 196–206.
[17] Y. Li, J. Jiang, T. Liu, M. Qiu, X. Sun, Personalized microtopic recommendation on microblogs, ACM Trans. Intell. Syst. Technol. 8 (6) (2017) 77.
[18] Y. Gong, Q. Zhang, X. Huang, Hashtag recommendation for multimodal microblog posts, Neurocomputing 272 (2018) 170–177.
[19] F.-F. Kou, J.-P. Du, C.-X. Yang, Y.-S. Shi, W.-Q. Cui, M.-Y. Liang, Y. Geng, Hashtag recommendation based on multi-features of microblogs, J. Comput. Sci. Technol. 33 (4) (2018) 711–726.
[20] J. Liu, Z. He, Y. Huang, Hashtag2Vec: Learning Hashtag Representation with Relational Hierarchical Embedding Model, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2018, pp. 3456–3462.
[21] Y. Djenouri, M. Comuzzi, D. Djenouri, Ss-fim: Single scan for frequent itemsets mining in transactional databases, Pacific-Asia Conference on Knowledge Discovery and Data Mining (2017) 644–654.
[22] G.Y.-Y. Chan, P. Xu, Z. Dai, L. Ren, V i b r: Visualizing bipartite relations at scale with the minimum description length principle, IEEE Trans. Visual Comput. Graphics 25 (1) (2018) 321–330.
[23] K. Gouda, M.J. Zaki, Efficiently mining maximal frequent itemsets, IEEE International Conference on Data Mining (2001) 163–170.
[24] Y. Li, J. Xu, Y.-H. Yuan, L. Chen, A new closed frequent itemset mining algorithm based on gpu and improved vertical structure, Concurrency and Computation: Practice and Experience 29 (6) (2017) e3904.
[25] S. Hosseini, S. Kalam, K. Barker, J.E. Ramirez-Marquez, Scheduling multi-component maintenance with a greedy heuristic local search algorithm, Soft. Comput. 24 (1) (2020) 351–366.
[26] B. Zhang, J.C.-W. Lin, Y. Shao, P. Fournier-Viger, Y. Djenouri, Maintenance of discovered high average-utility itemsets in dynamic databases, Appl. Sci. 8 (5) (2018) 769.
[27] J.C.-W. Lin, Y. Zhang, B. Zhang, P. Fournier-Viger, Y. Djenouri, Hiding sensitive itemsets with multiple objective optimization, Soft. Comput. 23 (23) (2019) 12779–12797.
[28] J.C.W. Lin, Y. Djenouri, G. Srivastava, U. Yun, P. Fournier-Vigerg, A predictive ga-based model for closed high-utility itemset mining, Appl. Soft Comput. 108 (2021) 107422.
[29] J.C.-W. Lin, Y. Djenouri, G. Srivastava, P. Fourier-Viger, Efficient evolutionary computation model of closed high-utility itemset mining, Appl. Intell. (2022).
[30] J.C.-W. Lin, Y. Djenouri, G. Srivastava, Efficient closed high-utility pattern fusion model in large-scale databases, Inf. Fusion 76 (2021) 122–132.
[31] J.C.-W. Lin, Y. Li, P. Fournier-Viger, Y. Djenouri, L.S.-L. Wang, Mining high-utility sequential patterns from big datasets, IEEE International Conference on Big Data (2019) 2674–2680.
[32] J.C.W. Lin, Y. Djenouri, G. Srivastava, Y. Li, P.S. Yu, Scalable mining of high-utility sequential patterns with three-tier mapreduce model, ACM Trans. Knowl. Discovery Data 16 (3) (2021) 1–26.
[33] H. Pan, E. Hou, N. Ansari, M-note: A multi-part ballot based e-voting system with clash attack protection, in: IEEE International Conference on Communications, 2015, pp. 7433–7437.
[34] H. Pan, E. Hou, N. Ansari, Re-note: An e-voting scheme based on ring signature and clash attack protection, IEEE Global Communications Conference (2013) 867–871.