IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2020, accepted February 29, 2020, date of publication March 6, 2020, date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2978505

HYPNER: A Hybrid Approach for Personalized News Recommendation

ASGHAR DARVISHY[1], HAMIDAH IBRAHIM[2], FATIMAH SIDI[2], AND AIDA MUSTAPHA[3]

[1]Department of Software Engineering, South Tehran Branch, Islamic Azad University, Tehran 15847, Iran
[2]Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Seri Kembangan 43400, Malaysia
[3]Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Malaysia

Corresponding author: Hamidah Ibrahim (hamidah.ibrahim@upm.edu.my)

This work was supported in part by the U.S. Department of Commerce under Grant BS123456, and in part by the Universiti Putra Malaysia through the PUTRA Grant Scheme under Grant GP-IPS/2013/9397400.

ABSTRACT A personalised news recommendation system extracts news set from multiple press releases and presents the recommended news to the user. In an effort to build a better recommender system with high accuracy, this paper proposes a personalised news recommendation framework named Hybrid Personalised NEws Recommendation (HYPNER). HYPNER combines both collaborative filtering-based and content-based filtering methods. The proposed framework aims at improving the accuracy of news recommendation by resolving the issues of scalability due to large news corpus, enriching the user's profile, representing the exact properties and characteristics of news items, and recommending diverse set of news items. Validation experiments showed that HYPNER achieved 81.56% improvement in $F1$-score and 5.33% in diversity as compared to an existing recommender system, SCENE.

INDEX TERMS Collaborative filtering, content-based filtering, news recommendation, personalised news recommendation.

I. INTRODUCTION

One of the most popular and general social media platforms is online news release that makes easy news accessing, sharing, and commenting. An efficient media for publishing news articles on the World Wide Web requires that it does not have traditional print-based publishing limitations. The number of news articles published per hour grows exponentially; so, multiple news sources around the world and the variety of news categories make it difficult for the user to find preferred news to read. Users prefer a filtered view of relevant news articles, allowing them to focus on news items that contain rich contextual information tailored to their behaviours and interests. Rich Site Summary (RSS) news feed by the news agencies only recommend a fixed number of news items to users without considering their preferences and interests in news reading.

A personalised news recommendation system extracts news set from multiple news press releases and processes such a high number of news articles before releasing the news set to the users. A typical personalised news recommendation system is made up of the following components: (i) user profile based on their historical reading behaviour, (ii) news metadata that summarises the news items according to terms and weights, (iii) news and user clusters that group similar users as well as similar news items into clusters, (iv) news selection that selects news from a big collection of news items based on their similarities, and (v) news representation which presents news items to the user.

A news item is specific in nature and is different from other items to recommend [17], [18], [32]. A news item may belong to more than one news category. Apart from that, a news item has a short lifetime and may expire in a small duration of time. Recency is the most commonly used property to determine a news lifetime based on the timespan of the first time the news is published. Another property is popularity, which shows the number of times a news item is read by the users throughout its lifetime. It is highly possible that a hot news item is read millions of times in few short minutes while an uninteresting news item is read less than a hundred times throughout its lifetime. However, recency and popularity of news articles change dramatically over time, which differentiates news items from products, books, and movies, where rendering the traditional recommendation methods ineffective [18].

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Marozzo.

Scalability is one of the issues in news recommendation that requires effective algorithms to deal with large news corpus. One of the common strategies used for solving scalability is clustering. However, the existing clustering algorithms do not take into consideration the news nature in clustering the news items. On the other hand, the existing personalised news recommendation systems do not make an attempt to filter the number of news items to recommend based on the reading rate behaviour of a user. Moreover, early researches only consider explicit profile, short-term profile, and long-term profile of a user but none of them use all of the above user profiles in a single solution. News selection is another issue that requires new solution to effectively select news items to recommend.

The main aim of this research work is to propose a personalised news recommendation framework that would improve the accuracy of news recommendation by resolving the issues mentioned above. The proposed framework incorporates both the collaborative filtering (CF)-based and content-based filtering methods along the following contributions:

(i) A new clustering algorithm named *Ordered Clustering* (OC) that is able to cluster news items and users based on the nature of news and user's reading behaviour.
(ii) A user profiling model that is made of user's explicit profile, long-term profile, and short-term profile. Short-term and long-term profiles are gathered from the user's reading behaviour.
(iii) A news metadata model that incorporates two new properties in user modelling, namely: *ReadingRate* and *HotnessRate*. While to enrich the news metadata a new property is defined called *Hotness*.
(iv) A news selection model that is based on the submodularity model in order to achieve diversity in news recommendation.

The remainder of this paper is structured as follows. Section II presents the attempts made by previous works in achieving accurate news recommendation. Section III gives the detailed explanation of the proposed news recommendation framework, named HYPNER. Section IV presents the experiments conducted to validate the proposed framework. Finally, Section V concludes the paper and gives some suggestions for future works.

II. RELATED WORKS

In recent years, there has been much focus on the design and development of personalised news recommendation systems that monitor and learn users' reading behaviours and generate news set based on these behaviours. Common news recommendation systems are often based on collaborative filtering (CF), content-based filtering (CB) or in some cases, hybrid methods. The CF-based news recommendation systems generate personalised recommendation for users based on their behaviours in news reading. In this method, similar users are clustered in a group based on their similarities in news access behavioural patterns. Such behaviours are expressed in the form of binary votes or numerical ratings on each news item. Nonetheless, CF algorithms have difficulty in generating reliable recommendation when data are sparse, and they cannot recommend news items that have no rating from the users, which often known as cold-start recommendation [16]. Google News [6], GroupLens [29], and DRN [35] are examples of CF-based method.

TABLE 1. Summary of news recommandation systems.

News Recommendation System	Technique	Methods		User Profiling				Properties			
		CF	*CB*	*I*	*Ex*	*S*	*L*	*P*	*Rc*	*H*	*Hr*
Google News [6]	Min-hash and PLSI	√	-	√	√	-	√	√	-	-	-
Google News [23]	Bayesian framework	√	√	√	-	-	√	√	-	-	-
SCENE [18]	Min-hash, LSH, and hierarchical clustering	√	√	√	-	√	√	√	√	-	-
PENETRATE [34]	User and news group clustering by hierarchical clustering	√	√	√	-	-	√	√	√	-	-
Yahoo! News [20]	Contextual bandit problem	√	√	√	-	-	√	√	-	-	-
CROWN [33]	Contextual information	-	√	√	√	-	√	-	√	-	-
PRemiSE [22]	Inference algorithm	√	√	√	-	-	√	-	-	-	-
CCNS [12]	*K*-means algorithm and dynamic weighted hybrid method	√	√	√	-	-	√	√	-	-	-
HYPNER	Ordered Clustering	√	√	√	√	√	√	-	-	√	√

Note: *I*: Implicit Profile, *Ex*: Explicit Profile, *S*: Short-term Profile, *L*: Long-term Profile, *P*: Popularity, *Rc*: Recency, *H*: Hotness; *Hr*: HotnessRate

On the other hand, content-based news recommendation system recommends news items based on content similarities between the news items and user's profile. It considers a given user's reading behaviour and analyses the content of the newly-published news before presenting it to the users. This type of system computes similarity between newly-published news items and the user's content-based profile and rates them. The news items with high rates are then recommended to the users. However, content-based methods cannot recommend accurately to a new user with low access in news reading [24], hence the rise of hybrid recommendation systems that combine two or more recommendation techniques in order to gain better performance [3]. Examples of content-based method include News Dude [2], Newsjunkie [8], CROWN [33], and the works by [7], [11], [15], [28], [32]. Meanwhile, representative examples of hybrid recommendation systems include [5], [13], [21], [22], [25], [32] (other examples can be found in Table 1).

Aside from the cold-start and data sparseness problems, scalability is one of the major issues in news recommendation that requires elegant algorithms to effectively deal with large news corpus [18], [34]. Several strategies can be used to address the scalability issue such as the MinHash algorithm [18] and clustering. The most commonly used clustering algorithms in news recommendation systems are

hierarchical clustering [18], [34] and k-means [12], [20]. Nevertheless, these clustering algorithms do not take into consideration the nature of news when clustering them into groups [18]. Consequently, a news item will only belong to a single cluster while in reality a news item can be categorised in more than one news category. This is also in line with the fact that users' interests are also not limited to only one news category. To address this gap, a clustering algorithm for news recommender system should be able to cluster news items without limiting their membership to a single cluster. It is also important to ensure that the clustering algorithm will not add any additional complexity computationally.

Early recommender systems used popularity [6], [12], [14], [20], [23] or recency [31], [33], or both [7], [18], [34] as properties to demonstrate the interestingness of the news items. Popularity represents the number of times a news item is read by the users throughout its lifetime. Recency, on the other hand, determines how long a news item has been published. However, both properties are insufficient because old news with high popularity might no longer be an interesting news items to read. On the other hand, newly-published news may have low popularity but may be an interesting item to read. In other words, a newly-published news item does not always indicate that it is more pleasing to read than those that were published earlier.

Several news recommendation frameworks have been proposed in an attempt to increase the recommendation accuracy, overcome the large volume of data, and recommend diverse of news items [12], [18], [20], [22], [23], [34]. Such frameworks required a rich model of user profiling in order to capture their news reading behaviours, which is an important property to determine the user preference in reading recent news vs. popular news. On the other hand, even the personalised news recommendation systems such as Google News [6], [23], SCENE [18], YahooNews [20], and PENETRATE [34] do not make an attempt to filter the number of news items to recommend. These systems recommend the same number of news items to the users, i.e. they are unable to recommend the appropriate number of news items to each user based on the individual user behaviour in news reading. Although, early researches consider explicit profile [6], [31], [33], short-term profile [18], [32], and long-term profile [6], [7], [12], [14], [18]–[20], [22], [23], [31]–[35], none of the works used the above different types of user profiles in a single solution.

One notable work on news recommendation system is SCENE [18] which focused on the issue of news selection. SCENE employed sub-modularity modelling and experimented how news set can be matched to the users' interests as much as possible while maintaining highest diversity of news. This is achieved by constructing a rich news metadata and user profiles that subsequently affect news selection, hence the accuracy of news recommendation [18]. Overall, news selection requires new strategy in utilising rich user profile and news metadata to assist news recommendation system in achieving accurate and diverse recommendation of news items.

Some of the notable research works from the literature in the area of news recommendation system are listed in Table 1.

III. HYBRID PERSONALISED NEWS RECOMMENDATION (HYPNER)

This paper proposes a new framework for news recommendation system named Hybrid Personalised NEws Recommendation (HYPNER). HYPNER is a hybrid recommendation framework, which combines Collaborative Filtering (CF)-based technique and Content-based technique. It consists of three components, which are *User and News Clustering, News Selection*, and *Personalised News Recommendation*. In the first component, *User and News Clustering*, news metadata is generated from the newly-published news articles via OpenCalais [30]. In order to support this component a new clustering algorithm called Ordered Clustering (OC) is proposed. The second component, *News Selection*, compares a given user's behaviour to the other similar users and matches the user's profile with the news metadata, to select the recommendable news set. Finally, the third component, *Personalised News Recommendation*, prioritises and ranks the pruned news articles to recommend the final news set to the user. Fig. 1 illustrates the framework along with the components and procedures, which is further elaborated in the following subsections. Table 2 summarises the symbols and notations used throughout the paper.

A. COLLABORATIVE FILTERING IN HYPNER

Collaborative filtering in HYPNER covers the two main components; *User and News Clustering* and *News Selection*. Under the *User and News Clustering* component, the procedures related to the construction of long-term user profile and *Clustering Users based on the Long-term User Profile* are utilised. Meanwhile, under the *News Selection* component, the *Select News from Similar Users* procedure and the *Weight News based on User Similarity in Clusters* procedure are utilised. In general, the CF-based method in HYPNER generates a news set that will serve as the input to the third component, the *Personalised News Recommendation*.

The main idea of the CF-based method is to select news items accurately based on similar users' reading behaviours. Here, the definitions and notations that are related to the CF-based method are provided.

Definition 1 (Set of Users): U is a set of users in the dataset D, i.e. $U = \{u_1, u_2, \ldots u_n\}$, where u_i is the *ith* user.

Definition 2 (Set of News Items): N is a set of news items in the dataset D, i.e. $N = \{n_1, n_2, \ldots, n_m\}$, where n_j is the *jth* news.

Definition 3 (News Rating (NR) Matrix): NR matrix is a $n \times m$ matrix with binary values. The entry NR_{ij} is either 1 or 0 where 1 indicates that the user u_i has read the news item n_j and 0 otherwise. The term rated is sometimes used to indicate the read action.

Fig. 2 illustrates a sample of news rating matrix that is gathered from a real dataset [1]. Here, $U = \{u_1, u_2, \ldots, u_{20}\}$ while $N = \{n_1, n_2, \ldots, n_{20}\}$, and NR is a 20×20 matrix.

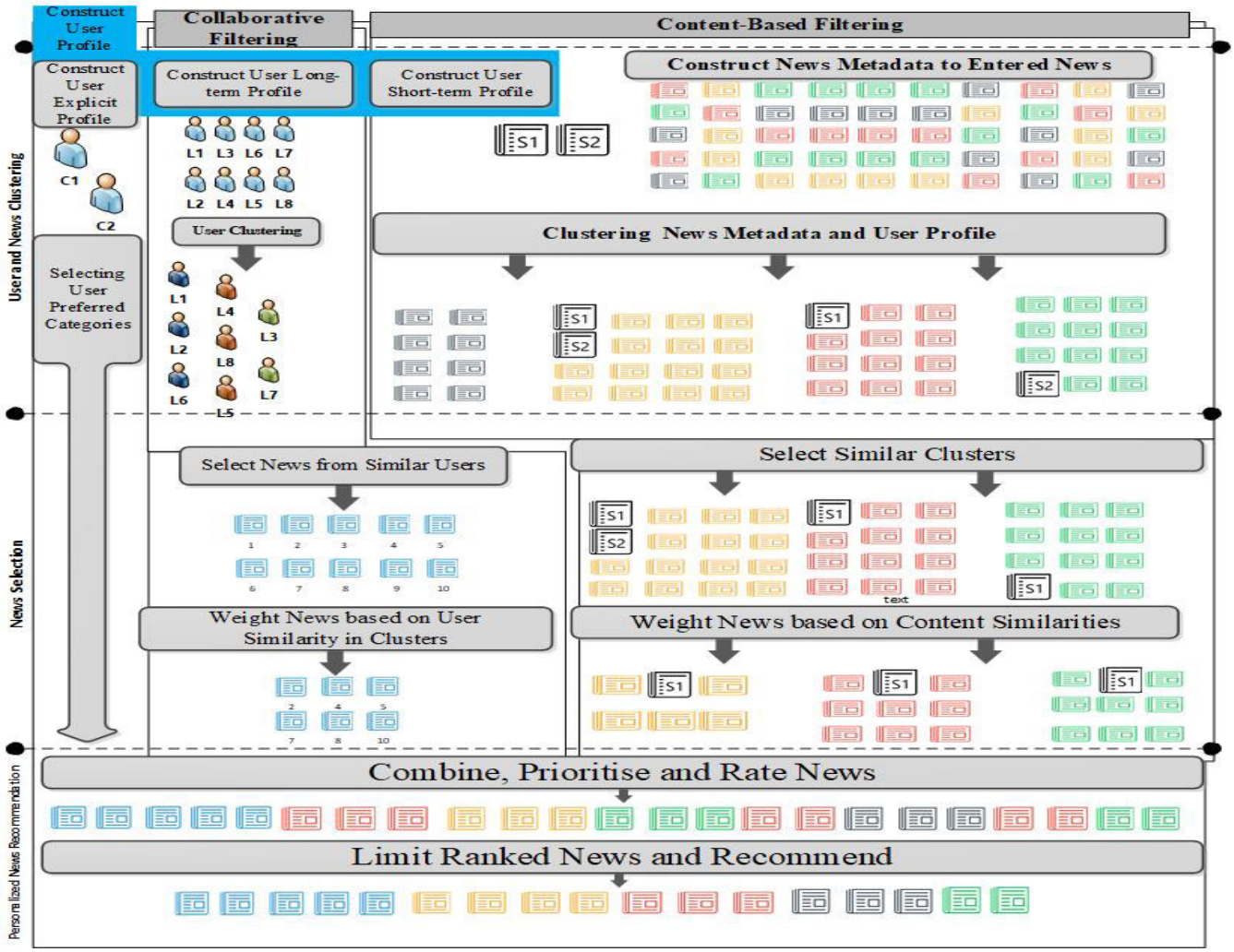

FIGURE 1. The proposed framework.

Note that the number of users and the number of news items are not necessarily equal.

Definition 4 (Binary Jaccard Similarity [10]): The Binary Jaccard Similarity measures similarity between two bit strings as shown in (1).

$$Sim(X, Y) = \frac{\sum_{i=1}^{l} (X_i \wedge Y_i)}{\sum_{i=1}^{l} (X_i \vee Y_i)} \tag{1}$$

where X and Y are two bitwise strings and l is the length of the bit strings.

Definition 5 (Similarity Matrix (SM) [6]): SM is a $n \times n$ matrix where SM_{ij} denotes the similarity ratio between user u_i and user u_j that is computed based on (1). The similarity ratio is a real value from 0 to 1 with $SM_{ij} = SM_{ji}$ and $SM_{ii} = 1$.

The users' reading behaviours are captured based on their click behaviours which can be represented as a bit string. For instance, based on Fig. 2 the bit string of user u_1 is (10010110…1) where 1 denotes that the news item has been read by the user u_1, i.e. the user has clicked on the news item

News / User	n_1	n_2	n_3	n_4	n_5	n_6	n_7	n_8	…	n_{20}
u_1	1	0	0	1	0	1	1	0	…	1
u_2	1	1	1	0	0	0	1	0	…	1
u_3	1	1	1	1	1	1	0	0	…	1
u_4	0	0	1	1	0	1	0	1	…	1
u_5	0	1	0	0	1	0	0	0	…	1
u_6	0	1	0	0	1	0	1	0	…	1
u_7	1	1	1	1	1	0	0	1	…	0
u_8	1	1	1	1	1	0	0	0	…	0
…	…	…	…	…	…	…	…	…	…	…
u_{20}	1	0	1	0	0	1	0	0	…	1

FIGURE 2. Example of news rating matrix.

while 0 otherwise. By utilising the Binary Jaccard Similarity formula, the peer-to-peer similarities between users can be computed. These similarity ratios are represented in a Similarity Matrix.

Fig. 3 illustrates the users' peer-to-peer similarities based on the click behaviours of 20 users, where in this example SM

TABLE 2. List of symbols/notations.

Symbols/Notations	Remarks
CF	Collaborative Filtering
CB	Content-based Filtering
$U = \{u_1, u_2, \ldots u_n\}$	A set of users (Definition 1)
$N = \{n_1, n_2, \ldots, n_m\}$	A set of news items (Definition 2)
NR	News Rating Matrix (Definition 3)
Sim(*X*, *Y*)	Binary Jaccard Similarity (Definition 4)
SM	Similarity Matrix (Definition 5)
L = <*Us*, *R*, *Hr*>	*L*: user profiling in the CF-based method; *Us*: a set of users with the similarity ratios to a given user; *R*: *ReadingRate*; *Hr*: *HotnessRate*
OC	Ordered Clustering
$C = \{c_1, c_2, \ldots c_k\}$	A set of clusters
J(*X*, *Y*)	Jaccard Similarity (Definition 6)
$SemiSim_{Jaccard}(X, Y)$	Jaccard Semi-Similarity (Definition 6)
$Sim_{cosine}(X, Y)$	Cosine Similarity (Definition 7)
$SemiSim_{Cosine}(X, Y)$	Cosine Semi-Similarity (Definition 7)
SemiSim(*X*, *Y*)	Semi-Similarity
N = <*T*, *E*, *P*, *Rc*, *H*>	*N*: news metadata; *T*: a set of named entities and their relevance tags that are extracted from the news topic; *E*: a set of named entities and their relevance tags that are extracted from the news content; *P*: *Popularity*; *Rc*: *Recency*; *H*: *Hotness*
S = <*T*, *E*>	*S*: short-term user profile; *T*: a set of named entities and their relevance tags that are extracted from the news topic; *E*: a set of named entities and their relevance tags that are extracted from the news content
S_{New}	User's new short-term profile
S_{Prev}	User's previous short-term profile
S_{recent}	User's recent activities in news reading
Ex	Explicit user profile
$News_{CF}$	CF-based news set
$News_{CB}$	Content-based news set
$News_{HYPNER}$	HYPNER news set

User	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	...	u_{20}
u_1	1	0.3125	0.3125	0.3125	0.4	0.5	0.3157	0.2778	...	0.5
u_2	0.3125	1	0.25	0.25	0.25	0.3571	0.3333	0.2941	...	0.4285
u_3	0.3125	0.25	1	0.5385	0.3333	0.3333	0.4117	0.4667	...	0.4285
u_4	0.3125	0.25	0.5385	1	0.25	0.25	0.3529	0.2222	...	0.3333
u_5	0.4	0.25	0.3333	0.25	1	0.6667	0.4375	0.4667	...	0.2666
u_6	0.5	0.3571	0.3333	0.25	0.6667	1	0.3333	0.2941	...	0.3333
u_7	0.3157	0.3333	0.4117	0.3529	0.4375	0.3333	1	0.7333	...	0.2631
u_8	0.2778	0.2941	0.4667	0.2222	0.4667	0.2941	0.7333	1	...	0.3125
...	...	...	...	...	...	...	...	...	...	...
u_{20}	0.5	0.4285	0.4285	0.3333	0.2667	0.3333	0.26316	0.3125	...	1

FIGURE 3. User similarity matrix.

u_x	u_y	u_z	...	u_j	...	u_l

FIGURE 4. An array list of cluster $c_i C_i$ consisting of users u_x to u_l.

is a 20 × 20 matrix. SM_{ij} denotes the similarity ratio between u_i and u_j. For instance, the similarity ratio between u_1 and u_2 is represented by $SM_{12} = 0.3125$. The higher the similarity ratio value, the higher is the similarities between the users. For instance, user u_1 reading behaviour is more similar to user u_6 ($SM_{16} = 0.5$) as compared to the reading behaviour between user u_1 and user u_8 ($SM_{18} = 0.2778$). This matrix is an upper triangular matrix.

1) CONSTRUCT LONG-TERM USER PROFILE

In this procedure, a long-term user profile is constructed for each user. A long-term user profile captures the properties related to a user which includes the user click behaviours as the main historical data. Each user click behaviour is presented in a news rating matrix, NR, and based on the NR a similarity matrix SM is computed. By examining the news published time, the user's news reading time, and the number of news items read by the user, new properties, namely: *HotnessRate* and *ReadingRate*, are established. To accommodate the above, a new model of the user profiling in the CF-based method is proposed, which is parameterised as a three-dimensional tuple: $L = \langle Us, R, Hr \rangle$ where:

1. *Us* represents a set of users and the similarity ratios to a given user u_i which are computed by utilising the Binary Jaccard Similarity (see Definition 4), i.e. $Us = \{\langle u_1, s_1 \rangle, \langle u_2, s_2 \rangle, \ldots, \langle u_j, s_j \rangle, \ldots, \langle u_n, s_n \rangle\}$ where u_j is the *jth* user and s_j is the similarity ratio between the given user u_i and user u_j, $1 \leq j \leq n$ and $i \neq j$.
2. *R* is the *ReadingRate* in HYPNER. Because the number of news articles that a user reads daily is different from the other users, this behaviour should be considered in the news selection process. *ReadingRate* is defined as a property to determine the average number of news items a user read in a day, e.g. user *a* reads *i* news on average per day and user *b* reads *j* news on average per day. HYPNER limits the number of recommended news items for a given user, u_i, to the coefficient of *ReadingRate* as shown in (2):
$$R = \frac{1}{n} \sum_{l=1}^{n} R_l \tag{2}$$
where *R* is the *ReadingRate*, *n* is the number of days that data acquisition is performed on the user u_i, and R_l is the number of news that the user u_i has read on the *lth* day.
3. *HotnessRate* (*Hr*) is the average value of *Hotness* of a news article which a user likes to read. For example, some users can follow sports news on weekends and they need to access all related news articles that were published in the previous week. This means the user likes to read news with lower degree of *Hotness*. However, some users prefer to read news articles with higher degree of *Hotness*. The *HotnessRate* for a given user, u_i, is defined as (3):
$$Hr = \frac{1}{m} \sum_{j=1}^{m} H_{nj} \tag{3}$$
where *Hr* is the *HotnessRate*, *m* is the number of news articles that a user u_i has read in the duration of data acquisition, and H_{nj} is the *Hotness* of the *jth* news that the user u_i has read.

2) USER CLUSTERING

Due to the fact that there is a large scale of news data, clustering needs to be performed. Generally, in the CF-based method, memory-based clustering is employed to cluster the users. As mentioned in Section II based on the news nature and the user's behavior, a multiple of membership clustering algorithm is needed to cluster the news items and users. In this paper, a new clustering algorithm named Ordered Clustering

(OC) is proposed and utilised to cluster the users and news items in the CF-based method of HYPNER. However, in this procedure, OC is employed to cluster the users. Here, the similarity matrix is the main input to the OC algorithm.

The OC algorithm, an unsupervised linear clustering algorithm, selects the highest similarity ratio in the Similarity Matrix and adds these users into a cluster. This process is repeated with the next highest similarity ratio. This means the users in a cluster are ordered in descending order based on the similarity ratios between the users. Consequently, given a user u_j of cluster c_i, the left-hand side neighbours of u_j is said to be more similar than the right-hand side neighbours of u_j. For example, refer to Fig. 4 which shows a cluster c_i with l members. To the given user u_j of cluster c_i, the left-hand side neighbours $u_x, u_y, \ldots, u_{j-1}$ are more similar compared to the u_j's right-hand side neighbours $u_{j+1}, u_{j+2}, \ldots, u_l$ as the similarity ratio values on the left-hand side of u_j are greater than those on the right-hand side.

The similarities between the users can be written as: $Sim(u_x, u_y) \geq Sim(u_y, u_z) \geq \ldots \geq Sim(u_{j-1}, u_j) \geq Sim(u_j, u_{j+1}) \geq \ldots \geq Sim(u_{l-2}, u_{l-1}) \geq Sim(u_{l-1}, u_l)$. To illustrate this, consider the users u_1, u_5, u_6, u_{11}, u_{12}, u_{14}, u_{16}, and u_{20} that are clustered in a cluster say c_3, with the following similarity ratios: $Sim(u_5, u_6) = 0.6667$, $Sim(u_6, u_{11}) = 0.6667$, $Sim(u_{11}, u_{14}) = 0.5835$, $Sim(u_{14}, u_{16}) = 0.5385$, $Sim(u_{16}, u_1) = 0.5$, $Sim(u_1, u_{12}) = 0.5$, and $Sim(u_{12}, u_{20}) = 0.3333$. Based on these values the users in c_3 are ordered to form $c_3 = \{u_5, u_6, u_{11}, u_{14}, u_{16}, u_1, u_{12}, u_{20}\}$.

Fig. 5 illustrates the proposed OC algorithm. Step 4 repeats and ensures that eventually all users of U are considered and belong to at least a cluster. In Step 6, the highest similarity ratio in SM is identified. This value indicated by SM_{ij} is assigned to a variable called Max (Step 7). In Step 8, the entry SM_{ij} of SM is set to 0 to avoid it from being chosen again in the next iteration. In Step 9, the existing clusters are checked and if user u_i is a member of an existing cluster say c_l then user u_j is added into the same cluster c_l of user u_i (Step 13). However, if u_i is not found in the cluster c_l but u_j is found to be a member of cluster c_l then u_i is added into the cluster c_l (Step 20). Yet, if both users u_i and u_j do not belong to any clusters, then a new cluster c_k is created and both u_i and u_j are added into the cluster c_k (Step 28). The algorithm is terminated when all users are members of at least one cluster, i.e. $U = \emptyset$.

3) SELECT NEWS FROM SIMILAR USERS

The CF-based method of HYPNER exploits the long-term user profile to predict the user's interest. The user's reading behaviour is modelled as a semi-supervised learning problem, whereby a newly-published news item is predicted either to be read by the user or otherwise. This procedure selects the clusters containing a given user u_j. Based on u_j's similar users in the selected clusters, the news set is formed from these similar users' reading behaviours. However, in our work only the users on the left-hand side of user u_j are selected and considered as similar users to user u_j.

Input: Similarity Matrix SM, Set of Users $U = \{u_1, u_2, \ldots, u_{20}\}$
Output: A set of Clusters $C = \{c_1, c_2, \ldots, c_k\}$

1. **BEGIN**
2. $k = 0$
3. $Found = F$
4. **WHILE** $U = \emptyset$ **DO**
5. **BEGIN**
6. Find the maximum value in SM
7. $Max = SM_{ij}$
8. $SM_{ij} = 0$
9. **FOR** each cluster c_l in C **AND** $Found \neq T$ **DO**
10. **BEGIN**
11. **IF** u_i is a member of the cluster c_l **THEN**
12. **BEGIN**
13. Insert u_j into c_l
14. $U = U - u_j$
15. $Found = T$
16. **END**
17. **ELSE**
18. **IF** u_j is a member of the cluster c_l **THEN**
19. **BEGIN**
20. Insert u_i into c_l
21. $U = U - u_i$
22. $Found = T$
23. **END**
24. **ELSE**
25. **BEGIN**
26. $k = k + 1$
27. Create a new cluster c_k
28. Insert u_i and u_j into c_k
29. $U = U - u_i$
30. $U = U - u_j$
31. $Found = T$
32. **END**
33. **END**
34. $Found = F$
35. **END**
36. **END**

FIGURE 5. The ordered clustering algorithm.

$$CM = \begin{bmatrix} 00110000000010100100 \\ 00010011000000101100 \\ 10001100001101010001 \\ 00000000010001010000 \\ 01000000100000000000 \\ 10001000001000000011 \end{bmatrix}$$

FIGURE 6. Cluster matrix.

4) WEIGHT NEWS BASED ON SIMILARITY IN CLUSTERS

In this procedure, based on the similarity ratios between a given user and his/her neighbouring users, the selected news items produced by the previous procedure are weighted. The weights are used to sort the news items in descending order. Based on the ReadingRate of a user, the top weighted news items are selected.

News items are completely different from other web objects like movies and music, in terms of popularity and recency. *Hotness* is an important property in news ranking of HYPNER. *Hotness* determines the interestingness of a news item, i.e. the degree of recency and popularity of a news item. When a news article is published online, there will be users that click it to read. A news item with higher *Hotness* value will have higher priority in news selection.

TABLE 3. The clustering results based on OC.

Cluster No	Members
c_1	$\{u_3, u_{13}, u_4, u_{15}, u_{18}\}$
c_2	$\{u_7, u_8, u_{15}, u_{18}, u_{17}, u_4\}$
c_3	$\{u_5, u_6, u_{11}, u_{14}, u_{16}, u_1, u_{12}, u_{20}\}$
c_4	$\{u_{10}, u_{16}, u_{14}\}$
c_5	$\{u_2, u_9\}$
c_6	$\{u_1, u_{19}, u_6, u_{12}, u_{20}\}$

TABLE 4. The clusters consisting of u_4.

Cluster No	Members
c_1	$\{u_3, u_{13}, u_4, u_{15}, u_{18}\}$
c_2	$\{u_7, u_8, u_{15}, u_{18}, u_{17}, u_4\}$

Besides, the difference of user *HotnessRate* and news *Hotness* is another property used in news selection. The *HotnessRate* of a given user u_i as Hr_{ui} is differentiated to the *Hotness* of news n_j as H_{nj} of each selected news item as given in (4).

$$\alpha = \left| Hr_{ui} - H_{nj} \right| \tag{4}$$

where Hr_{ui} is the *HotnessRate* of the *ith* user and H_{nj} is the *Hotness* of the *jth* news item. A news item with lower α is more similar to the user's behaviour and if $\alpha \rightarrow 0$ it means the news item n_j is similar to the user u_i's reading behaviour and this news item can be selected to recommend.

5) EXAMPLE

This example clarifies the processes of the CF-based method of HYPNER. In this example, we assume that there are 20 users, 20 news items, and user u_4 is the active user, i.e. $U = \{u_1, u_2, \ldots, u_{20}\}$ and $N = \{n_1, n_2, \ldots, n_{20}\}$. We also assume that the following have been constructed:

1. News Rating (*NR*) matrix as given in Figure 2.
2. Similarity Matrix (*SM*) as given in Figure 3.
3. The *HotnessRate* of the user u_4 is 12.6.

With the above assumptions, the followings are constructed:

1. Section A – 1 – Construct the long-term user profile with the structure $L = \langle Us, R, Hr \rangle$ where the entries of *Us* are derived from the *SM* given in Assumption 2 above.
2. Section A – 2 – Cluster the users using the OC algorithm. The results of this step are given in Table 3. From Table 3 it is obvious that 6 clusters have been formed, i.e. c_1, c_2, c_3, c_4, c_5, and c_6. Also, notice that multiple memberships can be seen from the results where a user is a member of more than one cluster. See for instance users u_1, u_4, u_{15}, u_{18}, u_{12}, u_6, u_{16}, and u_{20}. The results of clustering the users are represented in a Cluster Martix (*CM*) as shown in Fig. 6.
3. Section A – 3 – Given the active user u_4 and by referring to the *CM* matrix, the clusters in which u_4 is a member, are identified. In this case, clusters c_1 and c_2 are selected as shown in Table 4. Based on these clusters, the users that appeared on the left-hand side of u_4 are selected. The users on the left-hand side of user u_4 are more similar than those on the right-hand side of u_4. Thus, users u_3 and u_{13} are selected from cluster c_1 while users u_7, u_8, u_{15}, u_{18}, and u_{17} are selected from cluster c_2. Referring to the *NR* matrix, the news items that these users have rated are identified, as shown below:
$L_3 = \{n_1, n_2, n_3, n_4, n_5, n_6, n_{12}, n_{17}, n_{19}, n_{20}\}$
$L_{13} = \{n_1, n_2, n_3, n_4, n_5, n_6, n_8, n_{12}, n_{17}, n_{19}, n_{20}\}$
$L_7 = \{n_1, n_2, n_3, n_4, n_5, n_8, n_{10}, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{17}, n_{18}\}$
$L_8 = \{n_1, n_2, n_3, n_4, n_5, n_{10}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{17}\}$
$L_{15} = \{n_1, n_3, n_4, n_6, n_8, n_{11}, n_{12}, n_{15}, n_{16}, n_{17}\}$
$L_{18} = \{n_3, n_5, n_6, n_8, n_{10}, n_{12}, n_{14}, n_{15}, n_{16}, n_{17}\}$
$L_{17} = \{n_2, n_3, n_5, n_8, n_9, n_{10}, n_{11}, n_{12}, n_{15}, n_{17}, n_{20}\}$
where L_i is the set of news items read by the user u_i.
4. Section A – 4 – The similarity ratios between user u_4 and the other selected similar users in both clusters c_1 and c_2 are considered to weight all the selected news items. This is shown as follows:
$L = Sim(u_4, u_3)\{n_1, n_2, n_3, n_4, n_5, n_6, n_{12}, n_{17}, n_{19}, n_{20}\} + Sim(u_4, u_{13})\{n_1, n_2, n_3, n_4, n_5, n_6, n_8, n_{12}, n_{17}, n_{19}, n_{20}\} + Sim(u_4, u_7)\{n_1, n_2, n_3, n_4, n_5, n_8, n_{10}, n_{11}, n_{12}, n_{14}, n_{15}, n_{16}, n_{17}, n_{18}\} + Sim(u_4, u_8)\{n_1, n_2, n_3, n_4, n_5, n_{10}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{17}\} + Sim(u_4, u_{15})\{n_1, n_3, n_4, n_6, n_8, n_{11}, n_{12}, n_{15}, n_{16}, n_{17}\} + Sim(u_4, u_{18})\{n_3, n_5, n_6, n_8, n_{10}, n_{12}, n_{14}, n_{15}, n_{16}, n_{17}\} + Sim(u_4, u_{17})\ \{n_2, n_3, n_5, n_8, n_9, n_{10}, n_{11}, n_{12}, n_{15}, n_{17}, n_{20}\}$
where L is the selected news set based on similar user behaviors in the CF-based method. The $Sim(u_i, u_j)$ value is as presented in the Similarity Matrix in Fig. 3. For instance, the news item n_1 appeared in L_3, L_7, L_8, L_{13}, and L_{15}. The weight of n_1 is computed as follows: $Sim(u_4, u_3) + Sim(u_4, u_7) + Sim(u_4, u_8) + Sim(u_4, u_{13}) + Sim(u_4, u_{15}) = 0.5385 + 0.3529 + 0.2222 + 0.6154 + 0.5385 = 2.2675$. Thus, the weight of n_1 is 2.2675. Similar calculation is performed to weight the other news items. Based on (4), the difference between the *HotnessRate* of u_4, i.e. 12.6, and the *Hotness* of a selected news item, n_i, results in the α_i value as shown in Table 5. The final news set derived by the CF-based method of HYPNER based on this example is as shown below. The news items are ordered in ascending order of α_i values. These news items are passed to the *Personalised News Recommendation* component.

$$News_{CF} = \{n_3, n_4, n_1, n_6, n_8, n_{15}, n_{12}, n_2, n_{20}, n_{11}, n_{19}, n_{17}, n_{16}, n_{10}, n_{14}, n_9, n_{13}, n_{18}, n_5\}$$

B. CONTENT-BASED FILTERING IN HYPNER

This section presents our proposed Content-based filtering technique in HYPNER. It covers the two main components of HYPNER, namely: *User and News Clustering* and *News Selection*. Under the *User and News Clustering* component, the procedures utilised include *Construct Short-term User Profile*, *Construct News Metadata to Entered News*, and

TABLE 5. The selected news items and their weights.

News Items	Weight	News Hotness	HotnessRate of u_4	α
n_3	3.0008	13.33		0.73
n_4	2.2675	13.33		0.73
n_1	2.2675	11.67		0.93
n_6	1.7632	11.67		0.93
n_8	2.2401	15.00		2.40
n_{15}	1.8469	10.00		2.60
n_{12}	3.0008	16.67		4.07
n_2	2.1290	8.33		4.27
n_{20}	1.5539	8.33		4.27
n_{11}	1.2914	6.67	12.6	5.93
n_{19}	1.1539	6.67		5.93
n_{17}	3.0008	20.00		7.40
n_{16}	1.4469	5.00		7.60
n_{10}	1.3084	3.33		9.27
n_{14}	0.9084	1.67		10.93
n_9	0.4000	1.33		11.27
n_{13}	0.2222	1.00		11.60
n_{18}	0.3529	0.83		11.77
n_5	2.4623	25.00		12.40

Clustering News Metadata and User Profile while under the *News Selection* component the procedures involved are *Select Similar Clusters* and *Weight News based on Content Similarities*. The Content-based filtering technique in HYPNER generates a news set, which is the input to the *Personalised News Recommendation* of HYPNER.

In this subsection, the definitions that are related to the Content-based filtering technique in HYPNER are presented which are necessary to clarify the proposed procedures.

Definition 6 (Jaccard Semi-Similarity): Jaccard similarity coefficient [26] denotes similarity ratio between two finite sets as shown in (5).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{5}$$

where A and B are two finite sets on the dataset D. If $|A| \gg |B|$ or $A \to D$, then $J(A, B)$ moves to 0 as shown in (6).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \; and \; (A \to D \; or \; |A| \gg |B|) \; then \lim_{A \to D} \frac{A \cap B}{A \cup B} = 0 \tag{6}$$

Based on (6), Jaccard Semi-Similarity is defined as (7).

$$SemiSim_{Jaccard}(A, B) = \frac{|A \cap B|}{|B|} \; |A \cap B| \leq |B| \; then \; 0 \leq \frac{|A \cap B|}{|B|} \leq 1 \tag{7}$$

In our proposed models for user profiling and news metadata construction, the short-term user profile includes an unlimited number of named entities from the read news articles, however a news article has a limited number of named entities. On the other hand, to measure the similarity ratio between a given user u_i with a short-term profile S_i and a news item n_j with metadata N_j, the Jaccard Semi-Similarity can be employed when $|S_i| \gg |N_j|$. This similarity ratio is computed as (8).

$$SemiSim_{Jaccard}(S_i, N_j) = \frac{|S_i \cap N_j|}{|N_j|} \tag{8}$$

Definition 7 (Cosine Semi-Similarity): Based on Definition 6, the Cosine Similarity [4], [9] is as given in (9). The Cosine Semi-Similarity which is a modified version of Cosine Similarity is defined as (10).

$$Sim_{Cosine}(A, B) = \frac{A.B}{||A||.||B||} \tag{9}$$

$$SemiSim_{Cosine}(A, B) = \frac{A.B}{||B||} \tag{10}$$

where A and B are two finite sets on the dataset D. To measure the similarity ratio between a given user u_i with a short-term profile S_i and a news item n_j with metadata N_j, the Cosine Semi-Similarity is employed as (11).

$$SemiSim_{Cosine}(S_i, N_j) = \frac{S_i.N_j}{||N_j||} \tag{11}$$

The Jaccard Semi-Similarity and the Cosine Semi-Similarity are utilised to compute the similarity ratio between a short-term user profile and news metadata in the Content-based filtering method of HYPNER.

1) CONSTRUCT NEWS METADATA TO ENTERED NEWS

An entered news article includes important elements that are topic, unstructured body, and crawled time. The main tasks in news metadata construction are news summarisation and tokenization. The keywords of a news article are not predefined but are extracted from the text of the article using OpenCalais [30]. OpenCalais is a web service that automatically creates rich semantic metadata for the content. By running OpenCalais API, each news article and its topics are summarised into vector (named entity, relevance tag). To enrich news metadata the properties, namely: *Hotness*, *Popularity*, and *Recency* are defined which assist HYPNER in identifying news to recommend. *Hotness*, *Popularity*, and *Recency* are computed and updated during news access by HYPNER.

Generally, news metadata includes a summarised version of the news that can be extracted by the Term Frequency–Inverse Document Frequency (TF-IDF) algorithm [10]. In TF-IDF, the main structure of news metadata includes terms and their weights in the news content. In our work, news metadata consists of the news topics, the named entities of news contents, and the relevance tags that are extracted by OpenCalais. The news metadata N is parameterised with a five-dimensional tuple, $N = \langle T, E, P, Rc, H \rangle$ where:

1. T denotes a set of named entities and their relevance tags that are extracted from the news topic, $T = \{\langle t_1, tr_1 \rangle, \langle t_2, tr_2 \rangle, \ldots, \langle t_m, tr_m \rangle\}$ where t_i represents the named entity and tr_i represents the relevance tag of t_i.
2. E represents a set of named entities and their relevance tags that are extracted from the news content,

TABLE 6. Tokenization of news topics by OpenCalais.

Topics	Relevance Tag (%)
Sports	93
Indonesia	92

TABLE 7. Tokenization of news content by OpenCalais.

Named Entity	Named Entity	Relevance Tag (%)
City	Jakarta, Indonesia	20
Continent	Asia	20
Country	Indonesia	80
	China	20
	Malaysia	20
Person	Fajar Alfian	20
	Filia Paramita	20
	Joko Widodo	20
	Kevin Sukamuljo	20
Sport Event	Asian Games	100
Sports Game	Badminton	100

$E = \{\langle e_1, er_1\rangle, \langle e_2, er_2\rangle, \ldots, \langle e_m, \mathrm{er_m}\rangle\}$ where e_i represents the named entity and er_i represents the relevance tag of e_i, which can answer the question related to when, where, how, and by whom an event happened.

3. P is the news popularity and it represents the number of times a news article is read by the users.
4. Rc is the news recency and it is a score that is computed based on (12):

$$Rc = NewsReadTime - NewsPublishedTime \quad (12)$$

where *NewsReadTime* is the date and time that the news article is clicked to read and *NewsPublishedTime* is the date and time that the news article is published on the web.

5. H is the Hotness of a news article. In other words, it represents the interestingness of the news article. Hotness is computed as (13):

$$H = P/Rc \quad (13)$$

As an example, refer to Fig. 7 which illustrates a news article that is retrieved from the Star news agency (URL: https://www.thestar.com.my/sport/badminton/2018/08/28/badminton-a-smash-hit-as-indonesia-excels-at-asian-games/). Table 6 (Table 7) illustrates the news topics (news contents, respectively) that are tokenized by OpenCalais based on the entered news article (the news content, respectively) of Fig. 7.

Tables 6 and 7 are shown in tuple format as follows:

$$\begin{aligned} N &= \langle T, E, P, Rc, H\rangle \\ T &= \{\langle\text{“}Sports\text{”}, 0.93\rangle, \langle\text{“}Indonesia\text{”}, 0.92\rangle \\ E &= \{\langle\text{“}Jakarta, Indonesia\text{”}, 0.2\rangle, \langle\text{“}Asia\text{”}, 0.2\rangle, \\ &\quad \langle\text{“}Indonesia\text{”}, 0.8\rangle, \langle\text{“}China\text{”}, 0.2\rangle, \ldots, \\ &\quad \langle\text{“}Badminton\text{”}, 1.0\rangle\} \end{aligned}$$

FIGURE 7. An example of an entered news article.

2) CONSTRUCT SHORT-TERM USER PROFILE

A short-term user profile includes the user's recent behaviour in news reading and it is used in the selection of news items. In our research work, the content-based profile is considered as the short-term user profile. A short-term user profile includes the named entities and the relevance tags of news articles that are read by the user. The short-term user profile is parameterised with a two-dimensional tuple $S = \langle T, E\rangle$, where:

1. T represents the named entities with their relevance tags that are extracted from the topics of an accessed news article, $T = \{\langle t_1, tr_1\rangle, \langle t_2, tr_2\rangle, \ldots, \langle t_m, tr_m\rangle\}$ where t_i represents the named entity and tr_i represents the relevance tag of t_i, and they are gathered from the user's accessed news topics.
2. E is a set of named entities and their relevance tags that are extracted from the read news content, $E = \{\langle e_1, er_1\rangle, \langle e_2, er_2\rangle, \ldots, \langle e_m, er_m\rangle\}$ where e_i represents the named entity and er_i represents the relevance tag of e_i, and they are gathered from the user's accessed news content.

In the short-term user profile, named entities illustrate the news specifications that include its time, place, and doer that the user normally follows. Content distribution is created from the user's historical news reading which is gathered from the news content.

User interests in news reading can be changed. This is achieved by placing less weight on the user's older profile, considering the user's recent behaviour with a higher weight, and updating the user's profile with his/her recent reading behaviour. Over duration of time the user's previous accessed news content is ignored and the short-term user profile is updated based on the user's recent accessed reading. Equation (14) shows how a short-term user profile is updated over time.

$$S_{New} = (1-\alpha)S_{Prev} + \alpha S_{recent} \quad (14)$$

where α is a ratio in [0..1] to control how the user's previous profile can be updated based on the user's recent behaviour. S_{New} is the user's new short-term profile and S_{Prev} represents

the user's previous short-term profile. S_{recent} is the user's recent activities in the recommendation system that should be updated to the user's short-term profile. Here, the user's previous short-term profile, S_{Prev}, is updated by multiplying each relevance tag value of a named entity with $(1 - \alpha)$. While the user's current news access behaviour is updated by multiplying each relevance tag value of a named entity with α. The user's new short-term profile, S_{New}, is obtained by integrating the above two lists.

In HYPNER, the duration of the short-term user profile is considered one week (a user accesses a news recommendation system at least one time per week) and α is $\frac{H}{168}$. Here, 168 denotes the total hours in a week. H is the time difference between the user's current and previous access in hours. Based on the considered value of α, the short-term user profile has one week lifetime and the user's behaviour with more than one week lifetime will be removed from the short-term user profile and updated with the user's recent news access behaviour.

For example, given a user u_i and $H = 196$, then $\alpha = 96/168 = 0.5714$. Assuming that the S_{Prev} of $u_i = \{\langle$"*Friendship*", 0.8$\rangle$, $\langle$"*Love*", 0.75$\rangle$, $\langle$"*Life*", 0.7$\rangle$, while the S_{recent} of $u_i = \{\langle$"*Nature*", 0.9$\rangle$, $\langle$"*Holiday*", 0.8$\rangle$, $\langle$"*Life*", 0.75$\rangle\}$, then the S_{New} of u_i is updated as follows:

$$\begin{aligned} S_{New} &= (1 - 0.5714)S_{Prev} + 0.5714S_{recent} \\ &= 0.4286S_{Prev} + 0.5714S_{recent} \\ &= \{\langle\text{"}Friendship\text{"}, 0.3429\rangle, \langle\text{"}Love\text{"}, 0.3215\rangle, \\ &\quad \langle\text{"}Life\text{"}, 0.3000\rangle\} + \{\langle\text{"}Nature\text{"}, 0.5143\rangle, \\ &\quad \langle\text{"}Holiday\text{"}, 0.4571\rangle, \langle\text{"}Life\text{"}, 0.4286\rangle\} \\ &= \{\langle\text{"}Life\text{"}, 0.7286\rangle, \langle\text{"}Nature\text{"}, 0.5143\rangle, \\ &\quad \langle\text{"}Holiday\text{"}, 0.4571\rangle, \langle\text{"}Friendship\text{"}, 0.3429\rangle, \\ &\quad \langle\text{"}Love\text{"}, 0.3215\rangle\} \end{aligned}$$

3) CLUSTERING NEWS METADATA AND USER PROFILE

In this procedure news metadata and short-term user profile are clustered. The Ordered Clustering is performed based on the similarity ratios between the news metadata and the short-term user profile. This method creates a chain of news and users with similar content and it helps to select additional similar news items.

The similarity ratio is computed based on the news topics and news contents. As mentioned before, the short-term user profile includes a two-dimensional tuple $S = \langle T, E\rangle$, while news metadata includes a five-dimensional tuple $N = \langle T, E, P, Rc, H\rangle$. Semi-Similarity between the short-term user profile S and the news metadata N is defined as shown in (15) [18]:

$$SemiSim(S, N) = \frac{\alpha SemiSim_{Co\sin e}(T_S, T_N) + \beta SemiSim_{Co\sin e}(E_S, E_N)}{\sqrt{\alpha^2 + \beta^2}} \tag{15}$$

where α and β are parameters to control how the corresponding news metadata and short-term user profile are trusted. $SemiSim_{Cosine}(T_S, T_N)$ and $SemiSim_{Cosine}(E_S, E_N)$ are computed using the Cosine Semi-Similarity. Here, T_S is the news topic of short-term user profile, T_N is the news topic of news metadata, E_S is the news content of short-term user profile, and E_N is the news content of news metadata.

4) SELECT SIMILAR CLUSTERS

In the Content-based filtering method of HYPNER, the sub-modularity model is employed by utilising Cosine Semi-Similarity to compute the similarity ratio between the news metadata and the short-term user profile instead of Cosine Similarity as utilised by [18]. The news articles are selected and the recommendation news set is prepared for the user based on the contextual data similarity between the short-term user profile and the news metadata.

Most of the news articles in a cluster have similar or even the same topic, with minor difference in the main content of the corresponding topics. For example, a given news article cluster is reporting about "*FIFA World Cup 2018*", one piece of news may focus on the results in each match, while another may describe the video assistant referee (VAR) in this world cup. Naturally, a user is not interested in all similar given topics. Based on this insight, news selection strategy is presented in sub-modularity model and a quality function f is defined to evaluate the current selected news set S over the entire news cluster N as given in (16) [18].

$$\begin{aligned} f(S) = &\frac{1}{|N\backslash S|.|S|} \sum_{n_1 \in N\backslash S} \sum_{n_2 \in S} Sim(n_1, n_2) \\ &+ \frac{1}{\binom{|S|}{2}} \sum_{\substack{n_1, n_2 \in S \\ n_1 \neq n_2}} -Sim(n_1, n_2) \\ &+ \frac{1}{|S|} \sum_{n_1 \in S} SemiSim_{Co\sin e}(u, n_1) \end{aligned} \tag{16}$$

where S represents the selected news set, N denotes the original news cluster, n_1 and n_2 denote the news items, u represents the given user, $Sim(.,.)$ represents the similarity between two news metadata, and $SemiSim(.,.)$ represents the similarity between the user profile and the news metadata that is computed by the presented semi-similarity functions.

Note that there are three components in (16). The first one aims to evaluate the quality of how representative the selected news set S is over the original news set, the second one provides a perspective on the diversity of the selected news articles, and the last component gives the evidence of how much the selected news set matches to the user's behaviour. To combine the recommended news items from different news sets into the final recommendation set, the news item with the highest *Hotness* is selected within each set, where the number of news items selected in one set is proportional to the interest weight of the user on the corresponding topic category.

TABLE 8. The clustering results.

Cluster No	Members
c_1	$\{n_{20}, n_{11}, S_2, S_1, n_7, n_3\}$
c_2	$\{n_{14}, n_{13}, S_1, n_{18}\}$
c_3	$\{n_{12}, n_{10}, n_9, S_1, n_6, n_5\}$
c_4	$\{n_4, n_{19}, n_{11}, S_2, n_3\}$
c_5	$\{n_8, S_2, n_{14}\}$
c_6	$\{S_3, n_{10}, n_{12}, n_{16}\}$

5) WEIGHT NEWS BASED ON CONTENT SIMILARITES

By running the procedure *Select Similar Clusters* in the previous subsection, a news set is obtained in each cluster. Taking into account the particular characteristics of the news article such as *Hotness*, news set needs to be adjusted in order to create a recommendable news set.

The news *Hotness* is maintained in the news metadata to adjust the selected news set with normalised *Hotness* in descending order as given in (17).

$$H = abs(\frac{H_{ni} - H_{\min}}{H_{\max} - H_{\min}}) \tag{17}$$

where H_{ni} is the *Hotness* value of a given news article, H_{max} and H_{min} are the maximum and minimum values, respectively of *Hotness* in a news pool. *Hotness* is restricted by the number of times users click on a news item (popularity) and the news item access time (recency); the greater the value of *Hotness* means higher interestingness and consequently higher ranking of news article. The recommended news set needs one more comparison: *Hotness* of each news item is compared to the given user's *HotnessRate*. All the news articles with higher or equal to *Hotness* of the given user's *HotnessRate* will be chosen for recommendation.

6) EXAMPLE

This example clarifies the processes of the Content-based method of HYPNER. In this example, there are three users, $U = \{u_1, u_2, u_3\}$, with their respective short-term profile, $S = \{S_1, S_2, S_3\}$. We also assume that there are 19 news items, $N = \{n_2, n_3, \ldots, n_{20}\}$. The active user for this example is user u_1. We also assume that the followings have been constructed:

1. News metadata to entered news as discussed in Section B–1.
2. Short-term user profile as discussed in Section B – 2.
3. Similarity ratio between two news items n_i and n_j is computed utilising $Sim(n_i, n_j)$ as given in (15).
4. Similarity ratio between a news item n_i and a user u_k with short-term profile S_k is computed utilising $Sim(S_k, n_i)$ as given in (15).
5. The *ReadingRate* of the user u_1 is 10.

With the above assumptions, the following are performed:

1. Section B – 3 – Cluster the news metadata and short-term user profile by utilising the Ordered Clustering. The results of the clustering are shown in Table 8. We omit the details here as the process of clustering the news metadata and short-term user profile is similar to the process of clustering the users as shown in Section III – A – 2.
   From Table 8, it is obvious that 6 clusters have been formed, namely: $c_1, c_2, \ldots, c_6$. Also notice that there are users (represented by the short-term user profile) as well as news items (represented by the news metadata) that belong to more than one cluster.
2. Section B – 4 – The clusters consisting of the user u_1 with short-term profile S_1 as member are then selected. Referring to the Table 8 these clusters, namely: c_1, c_2, and c_3 are selected as follows:

$$c_1 = \{n_{20}, n_{11}, S_2, S_1, n_7, n_3\}$$
$$c_2 = \{n_{14}, n_{13}, S_1, n_{18}\}$$
$$c_3 = \{n_{12}, n_{10}, n_9, S_1, n_6, n_5\}$$

   The left hand-side members of S_1 from each cluster are then chosen. Only news items are selected. This is as shown below:

$$g_1 = \{n_{20}, n_{11}\}$$
$$g_2 = \{n_{14}, n_{13}\}$$
$$g_3 = \{n_{12}, n_{10}, n_9\}$$

   where g_1, g_2, and g_3 are derived from the clusters c_1, c_2, and c_3, respectively.
   Sub-modularity is then performed on each group g_1 using (16). Here, we only show the working example for g_3 as the rest can be calculated in a similar way.
   $fs = \frac{1}{2\times3}(Sim(n_6, n_{12}) + Sim(n_6, n_{10}) + Sim(n_6, n_9) + Sim(n_5, n_{12}) + Sim(n_5, n_{10}) + Sim(n_5, n_9)) + (-\frac{1}{3}(Sim(n_{12}, n_{10}) + Sim(n_{12}, n_9) + Sim(n_{10}, n_9))) + \frac{1}{3}(Sim(S_1, n_{12}) + Sim(S_1, n_{10}) + Sim(S_1, n_9)) = \frac{1}{2\times3}(0.3325 + 0.4178 + 0.2791 + 0.2661 + 0.2587 + 0.4211) + (-\frac{1}{3}(0.8231 + 0.7892 + 0.4518)) + \frac{1}{3}(0.7503 + 0.4761 + 0.4695) = 0.3292 + 0.6880 + 0.5653 = 0.2065$
3. Section B – 5 News items are weighted based on their content similarities. It should be noted that c_1 consists of the user u_2 short-term profile, S_2. This means that S_1 and S_2 are similar. We explore other clusters consisting of S_2 to be considered in the news selection of user u_1. Thus, the following clusters are selected for further analysis:

$$c_4 = \{n_4, n_{19}, n_{11}, S_2, n_3\}$$
$$c_5 = \{n_8, S_2, n_{14}\}$$

   The left hand-side members of S_2 from each cluster are then chosen as follows:

$$g_4 = \{n_4, n_{19}, n_{11}\}$$
$$g_5 = \{n_8\}$$

   where g_4 and g_5 are derived from the clusters c_4 and c_5, respectively. Each news item that belongs to g_1 or g_2 or g_3 has a primary weight of 1.0. If a news item belongs to g_4 or g_5 the weight of the news item $= 1.0 + Sim(S_1, S_2)$, for instance the weight for $n_{11} = 1.0 + Sim(S_1, S_2) = 1.0 + 0.7638 = 1.7638$. Here, we only

TABLE 9. The selected news items and their weights in $News_{CB}$.

News Item	n_{11}	n_{20}	n_{14}	n_{13}	n_{12}	n_{10}	n_9	n_4	n_{19}	n_8
Weight	1.7638	1.0	1.0	1.0	1.0	1.0	1.0	0.7638	0.7638	0.7638

TABLE 10. An example of news categories and sub-categories in explicit user profile.

News Categories							
News		Sport		Culture		Lifestyle	
Subcategory	**Rate**	**Subcategory**	**Rate**	**Subcategory**	**Rate**	**Subcategory**	**Rate**
World News	1.0	Football	1.0	Books	1.0	Fashion	0.0
UK news	0.0	Rugby	0.2	Music	1.0	Food	0.1
Science	0.8	Cricket	0.0	TV&radio	0.0	Recipes	0.0
Cities	0.5	Tennis	0.5	Art&design	0.0	Health&fitness	1.0
Tech	1.0	Cycling	0.8	Film	0.5	Home&garden	0.5
Business	0.5	Golf	0.0	Games	0.8	Women	0.0

show the calculation for n_{11} as the rest can be calculated in a similar way. Table 9 shows the selected news items and their weights.

In the final step of this procedure, for each news item, the value of *Hotness* is calculated based on (13) and normalised based on (17). Similar to the process described in Section III – A, the *Hotness* value and the *HotnessRate* value are used to calculate the value of α. The final $News_{CB}$ is as shown below which is passed to the next component, *Personalised News Recommendation*. Note that the number of news items is based on the *ReadingRate* of the user u_1, i.e. $ReadingRate = 10$.

$$News_{CB} = \{n_{11}, n_{20}, n_{14}, n_{13}, n_{12}, n_{10}, n_9, n_4, n_{19}, n_8\}$$

C. PERSONALISED NEWS RECOMMENDATION OF HYPNER

In this section, the *Personalised News Recommendation* of HYPNER is presented. It covers the two components of HYPNER, namely: *User and News Clustering* and *Personalised News Recommendation*. Under *User and News Clustering* component, the *Construct Explicit User Profile* procedure is utilised while under the *Personalised News Recommendation* component, the procedures involved are *Combine, Prioritise and Rate News* and *Limit Ranked News and Recommend*.

1) CONSTRUCT EXPLICIT USER PROFILE

The explicit user profile includes user's preferred categories which are an important priority in selecting and rating news items. It means news articles should be selected and recommended based on user preferred categories. The explicit user profile enhances the accuracy of user modelling and as a result provides better personalisation in the recommendation process.

A given user u_i rates explicitly a number of categories from a set of news categories. The explicit user profile is a two-dimensional vector: $Ex = \{\langle ct_1, r_1\rangle, \langle ct_2, r_2\rangle, \ldots, \langle ct_k, r_k\rangle\}$ where ct_j is the category of news items and r_j is the rate given by user u_i towards the category ct_j. r_j is a real value between [0, 1] that a user can select to express his/her level of interest. The explicit user profile is updatable since user's behaviour in news reading may change over time.

Table 10 shows an example of an explicit user profile. This example depicts the rating values given by a user for each subcategory of a news category. Here, the news categories set = {*"News", "Sport", "Culture", "Lifestyle"*}, while the subcategories set of *Sport* = {*"Football", "Rugby", ..., "Golf"*}. From the example, the user has given high rate for *World News*, *Tech*, *Football*, *Books*, *Music*, and *Health&fitness*, while low rate for *UK news*, *Cricket*, *Golf*, *TV&radio*, *Art&design*, *Fashion*, *Women*, and *Recipes*. Low rate indicates that the user is not interested to read the news items under that subcategory. Based on the structure determined above, we can represent the explicit user profile as follows:

$$Ex = \{\langle \text{"}WorldNews\text{"}, 1.0\rangle, \langle \text{"}Tech\text{"}, 1.0\rangle, \langle \text{"}Football\text{"}, 1.0\rangle, \ldots, \langle \text{"}Food\text{"}, 0.1\rangle\}$$

In real application, a user interface is designed where users can explicitly select and rate their preferred news categories and sub-categories. A real value between [0, 1] (0 indicates not interested while 1 denotes high interest category) is normally used to specify the user's preferences. However, the dataset obtained from [1] does not contain the explicit user profile. Thus, we have simulated the users' explicit profiles based on their historical reading behavior as follows. First, the user's read news items are grouped based on the news category. Second, the number of news read by the user in each category is counted. For example, the number of news read by a given user is 125 which include 40 Football news, 60 World news, and 25 Science news. In the third step, these values are normalised to real values between [0, 1.0] by dividing the number of news read by the user in each category to the number of news read by the user of a category with the highest value (in this example is 60). *Ex* can be constructed as follow:

$$\begin{aligned}
&\text{Read News List} \\
&\quad = \{\langle \text{"}WorldNews\text{"}, 60\rangle, \\
&\qquad \langle \text{"}Football\text{"}, 40\rangle, \langle \text{"}Science\text{"}, 25\rangle\} \\
Ex &= \{\langle \text{"}WorldNews\text{"}, 60/60\rangle, \langle \text{"}Football\text{"}, 40/60\rangle, \\
&\qquad \langle \text{"}Science\text{"}, 25/60\rangle\} \\
&= \{\langle \text{"}WorldNews\text{"}, 1.0\rangle, \langle \text{"}Football\text{"}, 0.67\rangle, \\
&\qquad \langle \text{"}Science\text{"}, 0.42\rangle\}
\end{aligned}$$

2) COMBINE, PRIORITISE, AND RATE NEWS

The selected news sets from both of the CF-based and the Content-based methods of HYPNER are combined to generate the final news set to recommend. An approach is proposed to prioritise the combined news set to finalise the recommended news set. The proposed approach is performed as follows.

Firstly, the two sets of the selected news, namely: the CF-based news set ($News_{CF}$) and the Content-based news set ($News_{CB}$) as well as the explicit user profile are passed to this procedure as inputs. Secondly, the $News_{CF}$ and $News_{CB}$ are combined and prioritised to produce the final news set, $News_{HYPNER}$, as shown below:

$$News_{HYPNER} = \alpha News_{CF} + \beta News_{CB}$$

where α and β are parameters to control how we trust the corresponding CF-based and Content-based methods. In our work, α and β are normalised values of $F1$-score of both the CF-based and Content-based methods of HYPNER. Here, α is multiplied to the weight of each news item in the selected news set from the CF-based method of HYPNER, $News_{CF}$, β is multiplied to the weight of each news item in the selected news set from the Content-based method of HYPNER, $News_{CF}$, while the final news set, $News_{HYPNER}$, is obtained by integrating the above two lists. Finally, the news items in $News_{HYPNER}$ are prioritised based on the category rates specified in the explicit user profile, Ex. The result of this procedure, i.e. $News_{HYPNER}$, is the input to the next procedure.

3) LIMIT RANKED NEWS AND RECOMMEND

Recall that long-term user profile includes ReadingRate. ReadingRate determines an average number of news items which a user prefers to read per day. Each user has a different behaviour in news reading and the number of daily news reading varies based on the user's interest and behaviour in news reading. The number of recommended news articles is computed as a coefficient of ReadingRate. In this procedure, HYPNER limits the number of news items by the coefficient of ReadingRate.

4) EXAMPLE

This example clarifies the processes of the *Personalised News Recommendation* component of HYPNER. In this example, we assume that there are 20 news items, 20 users, and user u_1 is the active user, i.e. $U = \{u_1, u_2, \ldots, u_{20}\}$ and $N = \{n_1, n_2, \ldots, n_{20}\}$. We also assume that the following have been constructed:

1. News set constructed by the CF-based method of HYPNER, $News_{CF}$ (Section III – A). In this example we assume the $News_{CF}$ is as follows: $News_{CF} = \{n_3, n_{11}, n_7, n_{20}, n_{13}, n_{14}, n_{18}, n_5\}$.
2. News set constructed by the Content-based method of HYPNER, $News_{CB}$(Section III – B). Here, we assume the $News_{CB}$ is as follows: $News_{CB} = \{n_3, n_{12}, n_{17}, n_5, n_1, n_4, n_2, n_{15}, n_8, n_{20}\}$.
3. In this example, for simplicity the weights of the news items in $News_{CF}$ and $News_{CB}$ are considered as 1.0. In actual the weights are the values calculated as shown in Section III – A – 4 and Section III – B – 5.
4. The *ReadingRate* of user $u_1 = 12.3$.

With the above assumptions, the following are performed:

1. Section C – 1 – construct the explicit user profile of user u_1, as follows: $Ex = \{\langle$"*World News*", 1.0$\rangle$, $\langle$"*Tech*", 1.0$\rangle$, $\langle$"*Football*", 1.0$\rangle, \ldots, \langle$"*Food*", 0.1$\rangle\}$
2. Section C – 2 – *Combine, Prioritise and Rate News*
   e.g. Let $F1\text{-}score_{CF} = 0.2$ and $F1\text{-}score_{CB} = 0.5$ then $\alpha = \frac{0.2}{0.5+0.2} = 0.29$ and $\beta = \frac{0.5}{0.5+0.2} = 0.71$. From here, $News_{HYPNER}$ is as follows:
   $News_{HYPNER} = 0.29\ News_{CF} + 0.71\ News_{CB}$
   $= 0.29\{n_3, n_{11}, n_7, n_{20}, n_{13}, n_{14}, n_{18}, n_5\} + 0.71\{n_3, n_{12}, n_{17}, n_5, n_1, n_4, n_2, n_{15}, n_8, n_{20}\}$
   $= \{1.0n_3,\ 1.0n_5,\ 1.0n_{20},\ 0.71n_{12},\ 0.71n_{17},\ 0.71n_1,\ 0.71n_4,\ 0.71n_2,\ 0.71n_{15},\ 0.71n_8,\ 0.29n_{11},\ 0.29n_7,\ 0.29n_{13}, 0.29n_{14}, 0.29n_{18}\}$
   Based on the Assumption 3, the weights of the news items in $News_{CF}$ and $News_{CB}$ are considered 1.0. Thus, the values of α and β are multiplied to the weights of the news items. $News_{HYPNER}$ is rearranged based on the rating that user u_1 has given to each news category. Table 11 shows the news items which are passed to the next procedure.
3. Section C – 3 – based on the Assumption 4, the *ReadingRate* of the user u_1 is 12.3. Therefore, 13 news items from the top list of Table 11 are recommended to the user u_1. The final $News_{HYPNER}$ is as follows:

$$News_{HYPNER} = \{n_3, n_5, n_{12}, n_{17}, n_1, n_{15}, n_8, n_{20}, n_{11}, n_7, n_4, n_2, n_{14}\}$$

TABLE 11. Selected news items based on user u_1's rates.

News Items	News Weight	News Category	Rates given by u_1
n_3	1	Music	1
n_5	1	Football	1
n_{12}	0.71	Football	1
n_{17}	0.71	World News	1
n_1	0.71	Tech	1
n_{15}	0.71	World News	1
n_8	0.71	World News	1
n_{20}	1	Cycling	0.8
n_{11}	0.29	Cycling	0.8
n_7	0.29	Games	0.8
n_4	0.71	Film	0.5
n_2	0.71	Film	0.5
n_{14}	0.29	Business	0.5
n_{13}	0.29	Food	0.1
n_{18}	0.29	Rugby	0

IV. RESULTS AND DISCUSSIONS

In this section, the experiment results of HYPNER are presented and discussed. To fairly evaluate the performance of HYPNER, it is compared to SCENE [18]. We implemented both the HYPNER and SCENE using Java programming language. Comprehensive experiments were conducted on a PC with Pentium V Intel (R) Core (TM) i5-3470 CPU @ 3.2 GHz PC with 8GB memory and Windows 10 professional 64-bits platform. Since the most important performance metric in news recommendation is accuracy, therefore the accuracy metrics, namely: Precision, Recall, $F1$-score, Micro-Averaged and Macro-Averaged of Precision, Recall, and $F1$-score, and diversity are considered. These performance metrics are as defined below:

1. Precision is the portion of recommended items that is in fact relevant and is defined as (18):

$$precision = \frac{tp}{tp + fp} \tag{18}$$

where *tp* (true positive) is the number of recommended items that is relevant and *fp* (false positive) is the number of recommended items that is irrelevant.

2. Recall is the portion of relevant items that is recommended to the active user and is defined as (19):

$$recall = \frac{tp}{tp + fn} \tag{19}$$

where *fn* (false negative) is the number of unsuccessful recommended items.

3. $F1$-score (F-measure), also known as the $F1$-measure, is the harmonic mean of precision and recall as shown in (20).

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{20}$$

4. Micro-averaged-precision and recall can be computed as (21) and (22), respectively:

$$Micro\text{-}Averaged\text{-}precision = \frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n} tp_i + \sum_{i=1}^{n} fp_i} \tag{21}$$

$$Micro\text{-}Averaged\text{-}recall = \frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n} tp_i + \sum_{i=1}^{n} fn_i} \tag{22}$$

where tp_i, fp_i, and fn_i are the true positive, false positive, and false negative, respectively, associated to the *ith* user, and n is the number of users.

5. Micro-averaged-$F1$-score is the harmonic mean of micro-averaged-precision and micro-averaged-recall as shown by (23).

$$Micro\text{-}Averaged\text{-}F1\text{-}Score = 2 \times \frac{Micro\text{-}Averaged\text{-}precision \times Micro\text{-}Averaged\text{-}recall}{Micro\text{-}Averaged\text{-}precision + Micro\text{-}Averaged\text{-}recall} \tag{23}$$

6. Macro-averaged-precision and recall are the mean values of users' precision and recall, respectively, that are computed individually for each user as defined in (24) and (25), respectively.

$$Macro\text{-}Averaged\text{-}precision = \frac{\sum_{i=1}^{n} precision_i}{n} \tag{24}$$

$$Macro\text{-}Averaged\text{-}recall = \frac{\sum_{i=1}^{n} recall_i}{n} \tag{25}$$

where $precision_i$ and $recall_i$ are the *ith* user's computed accuracy metrics in news recommendation.

7. Macro-averaged-$F1$-score is the harmonic mean of macro-averaged-precision and macro-averaged-recall as shown in (26).

$$Macro\text{-}Averaged\text{-}F1\text{-}Score = 2 \times \frac{Macro\text{-}Averaged\text{-}precision \times Macro\text{-}Averaged\text{-}recall}{Macro\text{-}Averaged\text{-}precision + Macro\text{-}Averaged\text{-}recall} \tag{26}$$

8. The news set diversity is defined as the average dissimilarity between news items that are recommended to a given user. The average dissimilarity of a given news set N, is computed as (27).

$$diversity(N) = \frac{2}{p(p-1)} \sum_{n_i \in N} \sum_{n_j \in N, n_j \neq n_i} (1 - Sim(n_i, n_j)) \tag{27}$$

where p is the number of members in set N and dissimilarity between news items n_i and n_j is represented as $1 - Sim(n_i, n_j)$, in which $Sim(n_i, n_j)$ denotes the news metadata similarity between the news items n_i and n_j [18].

TABLE 12. The parameter setting of the real dataset in the experiments.

No. of News Items	No. of Users	No. of News Readings	Descriptions
77,544	1,619	2,316,204	Available dataset [1]
38,737	1,009	1,161,798	Users with at least four news reading per day. Used in Experiment I.
26,445	107	271,415	Users with at least ten news reading per day. Used in Experiment II and Experiment III.

These metrics are measured per each given user that is randomly selected from the dataset. Each experiment is executed ten times and the average values are reported. To get results from the experiments, real dataset [1] is used.

The dataset was retrieved from the crawled Twitter information streams, includes 1,619 users who contributed at least 20 tweets in total and at least one tweet in each month of observation period, i.e. from October 2010 to January 2011. This dataset contains 77,544 news items and 2,316,204 news reading records. Data cleansing is performed on this dataset as to meet with the same criteria that have been used in SCENE [18]. Based on the criteria set on the designed experiments e.g. each user has read at least four news items per day, different sizes of dataset are utilised in the experiments. Table 12 presents the characteristics and statistical information of the dataset size and the criteria set for each experiment.

Three experiments were conducted to evaluate the proposed models of user profile and news metadata construction, the proposed clustering algorithm, and the proposed framework HYPNER. These series of experiments are designed to answer the following questions: (a) How do the proposed models of user profile and news metadata construction perform? (b) How does the proposed ordered clustering perform? (c) How does the proposed framework, HYPNER, perform as compare to SCENE [18], one of the notable works in news recommendation system? The results are presented in the following sections.

A. EXPERIMENT I - EVALUATE THE PROPOSED MODELS OF USER PROFILE AND NEWS METADATA CONSTRUCTION

This experiment aims to evaluate our proposed models of user profile and news metadata construction. In particular, it aims at evaluating the effect of having the new properties, namely: *Hotness*, *ReadingRate*, and *HotnessRate*. We have designed

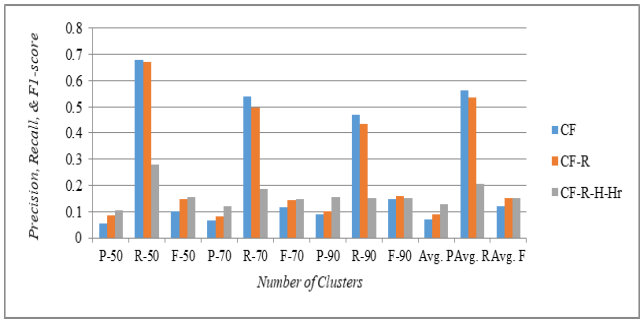

FIGURE 8. The accuracy results of the typical CF-based news recommendation (*CF*), typical CF-based news recommendation with *ReadingRate* (*CF-R*), and typical CF-based news recommendation with *ReadingRate*, *Hotness*, and *HotnessRate* (*CF-R-H-Hr*).

and implemented three different news recommendations in which k-means clustering algorithm is employed in each of them. They are:

(i) The typical CF-based news recommendation (*CF*) – The recommended news set to a given user is derived based on the user's neighbours behaviours that are in the same cluster. None of the above properties are applied in *CF*.

(ii) The typical CF-based news recommendation with *ReadingRate* (*CF-R*) – Similar to (i) above, however the number of recommended news items is limited to the coefficient of *ReadingRate*.

(iii) The typical CF-based news recommendation with *ReadingRate*, *Hotness*, and *HotnessRate* (*CF-R-H-Hr*) – The primitive dataset that is employed in the typical CF-based news recommendation is prioritised and tailored based on the news *Hotness* and user *HotnessRate*, while the number of news items is limited to the coefficient of *ReadingRate*.

Evaluation was measured in terms of accuracy based on the Precision (P), Recall (R), and $F1$-score (F) metrics. Each experiment was run ten times with one hundred users were randomly selected in each execution. Results of this experiment are shown in Fig. 8.

As depicted in Fig. 8, the number of clusters is varied in different executions ($k = 50$, 70, and 90) of the news recommendation. Based on the results, the following can be observed:

(i) The best result achieved by *CF* is when k is equal to 90 where $P = 0.0886$, $R = 0.4701$, and $F = 0.1491$.

(ii) The best result achieved by *CF-R* is when the number of clusters in k-means is 90. In comparison to *CF*, it is observed that by employing *ReadingRate* the accuracy in terms of $F1$-score increases by 50.40%, 20.54%, and 6.37% when $k = 50$, 70, and 90, respectively while on average the increment is 25.77%.

(iii) The best result achieved by *CF-R-H-Hr* is when $k = 50$ clusters with $F = 0.1543$. In comparison to *CF*, it is observed that employing *Hotness*, *HotnessRate*, and *ReadingRate* improves the accuracy of the news recommendation with increment of 55.23%, 23.68%, and 2.82% when $k = 50$, 70, and 90, respectively while on average the increment is 27.24%.

(iv) The more properties utilised, the higher the accuracy. When $k = 50$, the value of F for *CF* is 0.0994; while considering the *ReadingRate*, the value of F increased to 0.1495 as shown by *CF-R*. Utilising all the properties, the value of F increased to 0.1543 as shown by *CF-R-H-Hr*. Similar trend can be seen when $k = 70$. However, there is a slight drop in *CF-R-H-Hr* as compared to *CF-R* when $k = 90$. Nevertheless, on average *CF-R-H-Hr* achieved the highest accuracy with average of $F = 0.1511$.

The results of the experiment clearly show that our proposed models of user profile and news metadata construction have improved the accuracy of the news recommendation system.

B. EXPERIMENT II - EVALUATE THE PROPOSED ORDERED CLUSTERING

The second experiment aims to evaluate and compare the effect of utilising our proposed clustering algorithm, Ordered Clustering (OC). Three different news recommendations are designed and implemented such as follows:

(i) The typical CF-based news recommendation system which employs the k-means, *ReadingRate*, *Hotness*, and *HotnessRate* (*CF-kmeans*).

(ii) The CF-based method of HYPNER with Ordered Clustering.

(iii) The Content-based filtering method (*CB*) of HYPNER where the Ordered Clustering algorithm and the proposed models in user profile and news metadata construction are employed.

The experiment reported the results in terms of accuracy by measuring Micro-Averaged Precision (Mic-P), Macro-Averaged Precision (Mac-P), Micro-Averaged Recall (Mic-R), Macro-Averaged Recall (Mac-R), Micro-Averaged $F1$-score (Mic-F), and Macro-Averaged $F1$-score (Mac-F). In each execution, one hundred users were selected randomly. Fig. 9 shows the results of this experiment.

Based on Fig. 9, the results show that:

(i) *CB* has the highest accuracy, i.e. Mic-$F = 0.3521$ in comparison to *CF* with Mic-$F = 0.2096$ and *CF-kmeans* with Mic-$F = 0.0829$.

(ii) The Micro-Averaged of Precision, Recall, and $F1$-score of *CB* are 115.86%, 10.36%, and 67.99% (respectively) higher than *CF* while the Macro-Averaged of Precision, Recall, and $F1$-score of *CB* are 182.17%, 64.78%, and 132.54% (respectively) higher than *CF*.

(iii) The *CF* method gained an increment of 152.83% in F1-score as compared to *CF-kmeans*, while the improvement gained by *CB* method with respect to the same metric is 324.73% compared to the *CF-kmeans*.

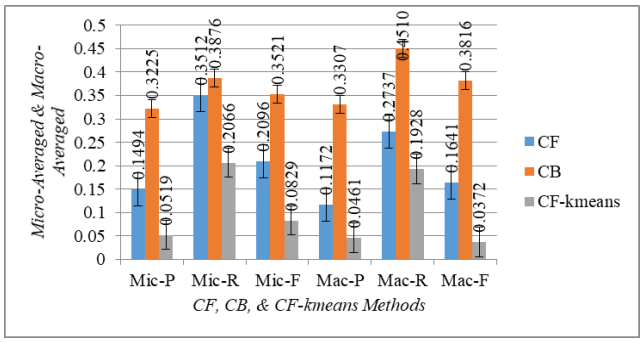

FIGURE 9. The Micro-Averaged and Macro-Averaged Precision, Recall, and $F1$-score Results of the CF-based Method of HYPNER (*CF*), CB-based Method of HYPNER (*CB*), and *CF-kmeans* Method.

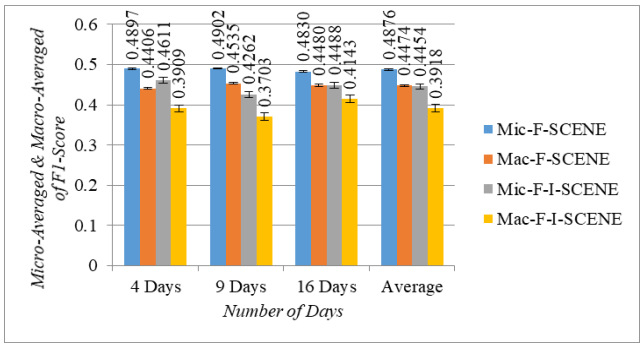

FIGURE 10. Comparison Results between SCENE, I-SCENE and error percentage computation.

(iv) The results show that utilising OC algorithm (*CF*) as opposed to k-means (*CF-kmeans*) has increased the accuracy. The Mic-F of *CF* is 0.2096 while Mic-F of *CF-kmeans* is 0.0829. This is due to the fact that *CF* employs the OC algorithm which is designed based on the news nature that assists the recommender system to recommend accurately.

(v) Utilising OC algorithm and the proposed models in user profile and news metadata construction has further increased the accuracy where Mic-F of $CF = 0.2096$ while Mic-F of $CB = 0.3521$.

The results of the experiment clearly show that our proposed OC algorithm as well as the models of user profile and news metadata construction have improved the accuracy of the news recommendation system.

C. EXPERIMENT III: EVALUATE HYPNER

In the third experiment the performance of HYPNER is evaluated and compared to the previous work, SCENE [18] by re-implementing it and naming it I-SCENE. To evaluate I-SCENE, different time ranges were considered and the accuracy metrics were computed that include Micro-Averaged and Macro-Averaged of Precision, Recall, and $F1$-score. The same time range used in the SCENE experiments was used so that fair comparison between I-SCENE and SCENE can be achieved.

The results of I-SCENE were first compared to the results of the original SCENE. Fig. 10 shows a comparison between SCENE and I-SCENE by Micro-Averaged $F1$-score (Mic-F) and Macro-Averaged $F1$-score (Mac-F), and their error percentage calculation. The average error in Micro-Averaged is 8.66% and in Macro-Averaged is 12.38%. By running the Mann-Whitney Test [17] on SCENE and I-SCENE results, it shows with 8.09% under estimate calculation error, the results of the I-SCENE are comparable to the original results from SCENE.

After I-SCENE has been validated, then comparison is performed between HYPNER and I-SCENE. Both HYPNER and I-SCENE were run on news articles on multiple time range with different intervals, i.e. 2 days, 4 days, 9 days, and 16 days. Experiment evaluation is performed in terms of accuracy by computing Micro-Averaged Precision, Macro-Averaged Precision, Micro-Averaged Recall, Macro-Averaged Recall, Micro-Averaged $F1$-score, Macro-Averaged $F1$-score, and diversity.

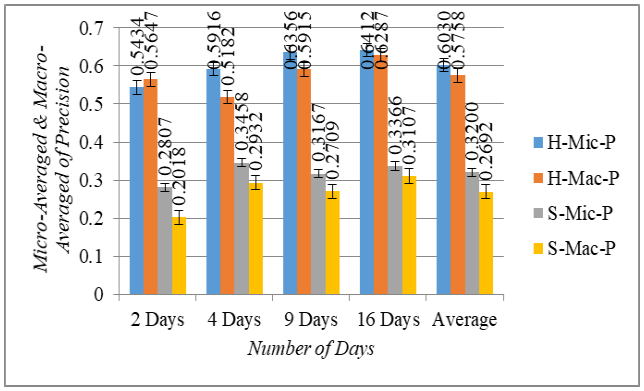

FIGURE 11. The micro-averaged and macro-averaged of precision results of HYPNER and I-SCENE in different time ranges.

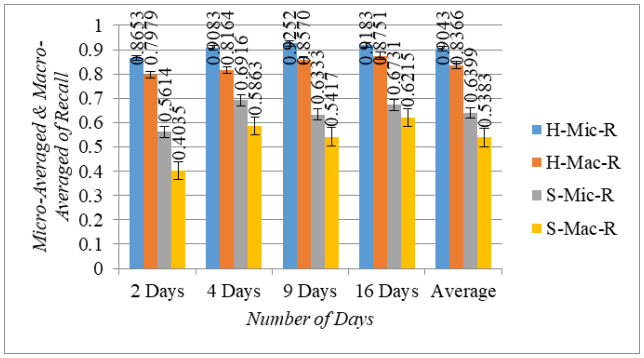

FIGURE 12. The micro-averaged and macro-averaged of recall results of HYPNER and I-SCENE in different time ranges.

Fig. 11, 12, and 13 show the results of Micro-Averaged and Macro-Averaged of Precision, Recall, and $F1$-score, respectively of HYPNER and I-SCENE. The results show:

(i) Regardless the time range (2, 4, 9, 16 days), the Micro-Averaged and Macro-Averaged of Precision, Recall, and $F1$-score of HYPNER is higher than I-SCENE.

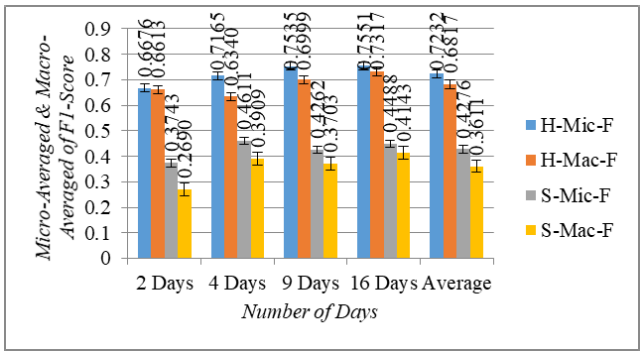

FIGURE 13. The micro-averaged and macro-averaged of $F1$-score Results of HYPNER and SCENE in different time ranges.

(ii) In most cases, for both HYPNER and I-SCENE, when the number of days increased, the Micro-Averaged and Macro-Averaged of Precision, Recall, and $F1$-score also increased.

(iii) The best P of HYPNER can be seen on the 16 days where Mic-P = 0.6412 and Mac-P = 0.6287, while improvement gained with regards to P as compared to I-SCENE is 104.14% on average.

(iv) The best Mic-R of HYPNER can be seen on the 9 days with 0.9252 while the best Mac-R = 0.8751 on the 16 days, while improvement gained with regards to R as compared to I-SCENE is 50.5% on average.

(v) The best $F1$-score of HYPNER can be seen on the 16 days where Mic-F = 0.7551 and Mac-F = 0.7317, while improvement gained with regards to $F1$-score as compared to I-SCENE is 81.56% on average.

The proposed clustering algorithm and new models in user profile news items as well as metadata construction were able to create a huge potential in identifying users' interest accurately and making recommendations that better match their behaviour. The results indicate that HYPNER is successful in news selection and recommendation. Utilising named entities instead of TF-IDF summarisation, new models in user profile, news metadata construction, and new clustering algorithm, i.e. Ordered Clustering, are the main strengths of HYPNER.

The final news set to recommend by HYPNER has a significant diversity on topic categories. Multiple memberships in Ordered Clustering help to arrange news items in diverse distributions. The proposed model increased diversity in news recommendation based on SCENE [18]. Table 13 shows diversity evaluation on the recommended news set by both I-SCENE and HYPNER. The main observation from the results is that increasing the recommended news set improved the diversity because news selection is performed within similar topic categories. Overall, HYPNER improved diversity on average by 5.33%.

V. CONCLUSION

News recommendation system is an automated approach built to provide the most appropriate information from the vast amount of data on the Internet. The main aim of a news recommendation system is to recommend news items that suit with the user's needs without manual exertion from the users. This paper was set to improve accuracy in news recommendation by highlighting the issues of clustering, news and user modelling, news rating, and news selection. A personalised news recommendation framework, HYPNER, has been proposed and the results showed that HYPNER achieved 81.56% improvement in terms of $F1$-score and 5.33% in terms of diversity as compared to an existing recommender system called SCENE. The solutions can be further investigated on other items of recommendation systems such as music, video or documents.

TABLE 13. Diversity evaluation on the recommended results.

Methods	Top@10	Top@20	Top@30	Average
I-SCENE	0.6930	0.6671	0.6059	0.6553
HYPNER	0.7415	0.6938	0.6362	0.6905
Improvement	7.00%	4.00%	5.00%	5.33%

ACKNOWLEDGMENT

All opinions, findings, conclusions, and recommendations in this article are those of the authors and do not necessarily reflect the views of the funding agencies. The authors would like to thank the anonymous reviewers for their comments.

REFERENCES

[1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao, "Analyzing user modeling on Twitter for personalized news recommendations," in *Proc. Int. Conf. User Modeling, Adaptation, Personalization*, 2011, pp. 1–12.

[2] D. Billsus and M. J. Pazzani, "A personal news agent that talks, learns and explains," in *Proc. 3rd Annu. Conf. Auton. Agents (AGENTS)*, 1999, pp. 175–268.

[3] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, 2002.

[4] K. Choi and Y. Suh, "A new similarity function for selecting neighbors for each target item in collaborative filtering," *Knowl.-Based Syst.*, vol. 37, pp. 146–153, Jan. 2013.

[5] M. Claypool, A. Gokhale, and T. Miranda, "Combining content-based and collaborative filters in an online newspaper," in *Proc. ACM SIGIR Workshop Rec. Syst.-Implement. Eval.*, 1999.

[6] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: Scalable online collaborative filtering," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 271–280.

[7] B. Fortuna, C. Fortuna, and D. Mladenić, "Real-time news recommender system," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2010, pp. 583–586.

[8] E. Gabrilovich, S. Dumais, and E. Horvitz, "Newsjunkie: Providing personalized newsfeeds via analysis of information novelty," in *Proc. 13th Conf. World Wide Web (WWW)*, 2004, pp. 482–490.

[9] L. Hamers, Y. Hemeryck, G. Herweyers, M. Janssen, H. Keters, R. Rousseau, and A. Vanhoutte, "Similarity measures in scientometric research: The jaccard index versus Salton's cosine formula," *Inf. Process. Manage.*, vol. 25, no. 3, pp. 315–318, Jan. 1989.

[10] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.

[11] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom, "Ontology-based news recommendation," in *Proc. 1st Int. Workshop Data Semantics (DataSem)*, 2010, pp. 1–6.

[12] S. Jiang and W. Hong, "A vertical news recommendation system: CCNS—An example from Chinese campus news reading system," in *Proc. 9th. Int. Conf. Comput. Sci. Educ. (ICCSE)*, 2014, pp. 1105–1114.

[13] N. Jonnalagedda and S. Gauch, "Personalized news recommendation using Twitter," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. (WI), Intell. Agent Technol. (IAT)*, Nov. 2013, pp. 21–25.
[14] D. Khattar, V. Kumar, M. Gupta, and V. Varma, "Neural content-collaborative filtering for news recommendation," in *Proc. NewsIR Workshop*, 2018, pp. 45–50.
[15] M. Kompan and M. Bielikova, "Content-based news recommendation," in *Proc. 11th. Int. Conf. E-Commerce Web Technol. (Ec-Web)*, 2010, pp. 61–72.
[16] W.-K. C. Leung, "Enriching user and item profiles for collaborative filtering: From concept hierarchies to user-generated reviews," Hong Kong Polytechnic Univ., Hong Kong, Tech. Rep., 2009.
[17] L. Li, D.-D. Wang, S.-Z. Zhu, and T. Li, "Personalized news recommendation: A review and an experimental investigation," *J. Comput. Sci. Technol.*, vol. 26, no. 5, pp. 754–766, Sep. 2011.
[18] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan, "SCENE: A scalable two-stage personalized news recommendation system," in *Proc. 34th. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2011, pp. 125–134.
[19] L. Li, L. Zheng, and T. Li, "LOGO: A long-short user interest integration in personalized news recommendation," in *Proc. 5th ACM Conf. Rec. Syst.*, 2011, pp. 317–320.
[20] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 661–670.
[21] C. Lin, R. Xie, L. Li, Z. Huang, and T. Li, "PRemiSE: Personalized news recommendation via implicit social experts," in *Proc. 21st. ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 1607–1611.
[22] C. Lin, R. Xie, X. Guan, L. Li, and T. Li, "Personalized news recommendation via implicit social experts," *Inf. Sci.*, vol. 254, pp. 1–18, Jan. 2014.
[23] J. Liu, P. Dolan, and E. R. Pedersen, "Personalized news recommendation based on click behavior," in *Proc. 15th Int. Conf. Intell. User Interfaces (IUI)*, 2010, pp. 31–40.
[24] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang, "Content-based collaborative filtering for news topic recommendation," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 217–223.
[25] A. Montes-García, J. M. Álvarez-Rodríguez, J. E. Labra-Gayo, and M. Martínez-Merino, "Towards a journalist-based news recommendation system: The wesomender approach," *Expert Syst. Appl.*, vol. 40, no. 17, pp. 6735–6741, Dec. 2013.
[26] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proc. Int. MultiConf. Eng. Comput. Sci.*, 2013, pp. 380–384.
[27] R. L. Ott and M. T. Longnecker, "An introduction to statistical methods and data analysis," Cengage, Boston, MA, USA," Tech. Rep., 2015.
[28] O. Phelan, K. McCarthy, M. Bennett, and B. Smyth, "Terms of a feather: Content-based news recommendation and discovery using Twitter," in *Proc. 33rd. Eur. Conf. IR Res. (ECIR)*, 2011, pp. 448–459.
[29] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in *Proc. ACM Conf. Comput. Supported Cooperat. Work (CSCW)*, 1994, pp. 175–186.
[30] T. Reuters, "OpenCalais," Tech. Rep., Jun. 2008.
[31] W. Saber, M. Nasr, and M. Saied, "A hybrid news recommender system," in *Proc. Res. World Int. Conf.*, 2018.
[32] P. Viana and M. Soares, "A hybrid recommendation system for news in a mobile environment," in *Proc. 6th Int. Conf. Web Intell., Mining Semantics (WIMS)*, 2016, pp. 1–9.
[33] S. Wang, B. Zou, C. Li, K. Zhao, Q. Liu, and H. Chen, "CROWN: A context-aware RecOmmender for Web news," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Apr. 2015, pp. 1420–1423.
[34] L. Zheng, L. Li, W. Hong, and T. Li, "PENETRATE: Personalized news recommendation using ensemble hierarchical clustering," *Expert Syst. Appl.*, vol. 40, no. 6, pp. 2127–2136, May 2013.
[35] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li, "DRN: A deep reinforcement learning framework for news recommendation," in *Proc. Int. Conf. World Wide Web*, 2018, pp. 167–176.

ASGHAR DARVISHY received the Ph.D. degree in intelligent computing (IC) focusing on recommendation systems from Universiti Putra Malaysia (UPM), in 2019. He is currently a member of Faculty with the Department of Software Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran. His main research interests include database, data warehouse, data quality, data mining, big data, data science, business intelligence, intelligent computing, and recommendation systems.

HAMIDAH IBRAHIM received the Ph.D. degree in computer science from the University of Wales Cardiff, U.K., in 1998. She is currently a Full Professor with the Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM). Her current research interests include databases (distributed, parallel, mobile, biomedical, and XML) focusing on issues related to integrity maintenance/checking, ontology/schema/data integration, ontology/schema/data mapping, cache management, access control, data security, transaction processing, query optimization, query reformulation, preference evaluation–context-aware, information extraction, concurrency control, and data management in grid and knowledge-based systems.

FATIMAH SIDI received the Ph.D. degree in management information system from Universiti Putra Malaysia, Malaysia (UPM), in 2008. She is currently working as an Associate Professor in the discipline of computer science with the Department of Computer Science, Faculty of Computer Science and Information Technology, UPM. Her current research interests are knowledge and information management systems, data and knowledge engineering, and database and data warehouse.

AIDA MUSTAPHA received the B.Sc. degree in computer science from Michigan Technological University, in 1998, the M.Sc. (IT) degree in computer science from UKM, Malaysia, in 2004, and the Ph.D. degree in artificial intelligence focusing on dialogue systems. She is currently an Active Researcher in the area of computational linguistics, soft computing, data mining, and agent-based systems.