Personal or General? A Hybrid Strategy with Multi-factors for News Recommendation

ZHENYA HUANG, University of Science and Technology of China, China and State Key Laboratory of Cognitive Intelligence, China
BINBIN JIN, Huawei Cloud Computing Technologies, Co., Ltd., China
HONGKE ZHAO, Tianjin University, China
QI LIU and DEFU LIAN, University of Science and Technology of China, China and State Key Laboratory of Cognitive Intelligence, China
BAO TENGFEI, Beijing Bytedance Technology Co., Ltd., China
ENHONG CHEN, University of Science and Technology of China, China and State Key Laboratory of Cognitive Intelligence, China

News recommender systems have become an effective manner to help users make decisions by suggesting the potential news that users may click and read, which has shown the proliferation nowadays. Many representative algorithms made great efforts to discover users' preferences from the histories for triggering news recommendations. However, there exist some limitations due to the following two main issues. First, they mainly rely on the sufficient user data, which cannot well capture users' temporal interests with very limited records. Second, always perceiving users' histories for recommendation may ignore some important news (e.g., breaking news). In this article, we propose a novel Multi-factors Fusion model for news recommendation by integrating both user-dependent preference effect and user-independent timeliness effect together. First, to track the preference of a certain user, we decompose her reading history into two user-related factors, including the long-term habit and the short-term interest. Specifically, we extract her persistent habit by exploring the category effect of news that she focuses on from her whole records. Then, we characterize her temporary interests by proposing a recurrent neural network of analyzing the homogeneous relations between her latest clicked news and the candidate ones. Second, to describe the user-independent news timeliness effect, we propose a novel survival analysis model to estimate the instantaneous click probability of a certain news as the occurring probability of an event, where much sensational news tends to be picked out. Last, we fuse all effects to determine the probability of a user clicking on a certain news under the independent event assumption. We conduct extensive experiments on two real-world datasets. Experimental

This work was done when Binbin Jin was an intern at Beijing Bytedance Technology Co., Ltd.
This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2021YFF0901003), the National Natural Science Foundation of China (Grants No. 62106244, 61922073, U20A20229, and 72101176), and the Fundamental Research Funds for the Central Universities (Grant No. WK2150110021).
Authors' addresses: Z. Huang, Q. Liu, D. Lian, and E. Chen (corresponding author), Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China, Hefei, China, 230027 and State Key Laboratory of Cognitive Intelligence, Hefei, China, 230088; emails: {huangzhy, qiliuql, liandefu, cheneh}@ustc.edu.cn; B. Jin, Huawei Cloud Computing Technologies, Co., Ltd., Hangzhou, China, 310052; email: jinbinbin1@huawei.com; H. Zhao, College of Management and Economics, Tianjin University, Tianjin, China, 300072; email: hongke@tju.edu.cn; B. Tengfei, Beijing Bytedance Technology Co., Ltd., Beijing, China, 100083; email: baotengfei@bytedance.com.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Association for Computing Machinery.
1046-8188/2023/04-ART44 $15.00
https://doi.org/10.1145/3555373

results demonstrate that our model can generate better news recommendations on both general scenario and cold-start scenario.

CCS Concepts: • **Information systems** → **Recommender systems**; **Information extraction**; *Data mining;*

Additional Key Words and Phrases: News recommendation, user-dependent preference, user-independent timeliness, survival analysis

ACM Reference format:
Zhenya Huang, Binbin Jin, Hongke Zhao, Qi Liu, Defu Lian, Bao Tengfei, and Enhong Chen. 2023. Personal or General? A Hybrid Strategy with Multi-factors for News Recommendation. *ACM Trans. Inf. Syst.* 41, 2, Article 44 (April 2023), 29 pages.
https://doi.org/10.1145/3555373

1 INTRODUCTION

Online news platforms, such as Google News[1] and Toutiao,[2] have shown much proliferation nowadays. Compared with the traditional media forms, e.g., newspaper, broadcast, and TV, these platforms can aggregate and collect massive emerging news articles without being limited by time length and space and distribute them to users in time. Therefore, millions of users have been attracted, as they can save much effort searching and getting real-time news information every day.

In real-world scenarios, since there is a large number of news emerging and updating frequently every day, users can easily get caught in a dilemma of information explosion because they are generally difficult to seek the news which they are interested in [70]. To improve user experience, recommender systems become an effective manner to help users make decisions by suggesting the potential news that users may click and read [27]. Toward this goal, learning from the experience in various representative fields, e.g., e-commerce [19, 41], movie [48, 81], and POI [53, 77], the general algorithms always try to discover the user preferences for making news recommendations, such as collaborative filtering [10] and content-based filtering [14, 38]. Basically, collaborative filtering assumes that users may share the same preference with others who have the similar behaviors. Moreover, since users are usually attracted by the news item information such as title, and so on, content-based filtering, as the mainstream approaches in news recommendation [64, 67], perceives user preference by analyzing the content of news that they clicked in history as the evidence. Although they have made some achievements in the past, always recommending news following user historical preferences may be limited in practice sometimes [32, 47]. First, such methods require the sufficient user histories for optimization and therefore cannot deal with the cold-start users whose historical records are few or even empty. More importantly, most of them may ignore the specific news effect, which is different from the general scenarios like e-commerce. For example, as much literature has suggested [51, 68], timeliness, as one of the unique news factors for characterizing the lifecycle, though not directly related to users, would also affect their decisions, since users may go through the breaking news every day without following their personal interests [57]. In summary, most mainstream approaches cannot well satisfy the news recommendations, because both the user-dependent preference and user-independent timeliness are not explored sufficiently.

In this article, we provide a focused study for news recommendations by addressing the above problems. However, there are several major challenges on both sides. On one hand, learning the user-dependent preference should perceive the user's news-reading histories, where the preference is generally coupled with two parts including the long-term habit and short-term interest.

[1] https://news.google.com.
[2] https://www.toutiao.com.

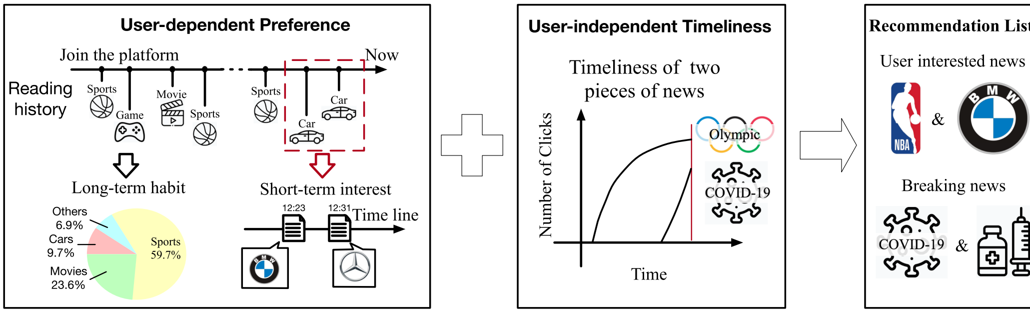


Fig. 1. An illustrative example of simultaneously recommending user interested news and breaking news to a user. The left part refers to a user's reading history, which is utilized to analyze her long-term habit and short-term interest. The middle part refers to the timeliness of two pieces of news. The right part denotes the recommendation list to the user.

Taking an example shown in the left part of Figure 1, given a user's reading records, the long-term habit indicates what kinds of news she likes to read consistently (e.g., sports and movies), since she always scans the "sports" news and "movie" in the history. Comparatively, the short-term one reflects that she may change the preference due to some possible temporary demands. Focusing on her latest records in Figure 1, we can find that she goes through many "car" articles in a short period of time (as she may have the plan to buy a car in the near future) although she seldom pays the attention in the earlier. Based on this observation, it makes sense to recommend either *NBA* or *BMW* news to her at this time. In the literature, existing work [1, 75] for news recommendation usually mixes both factors up instead of separately distinguishing the effect of each. In this article, we argue either of them can affect a user's behavior from different perspective, and therefore, an appropriate approach for addressing this issue is required.

On the other hand, in real-world scenarios, news is emerging and updating rapidly worldwide, which reflects the unique timeliness effect to demonstrate its own characteristics. Driven by this factor, users can always be attracted by some breaking news. For example, in Figure 1, the "COVID-19" news has boosted the concerns throughout the world since the year 2020, and many people consistently care about the change of the pandemic. Therefore, it is also reasonable to recommend "COVID-19" news to users, though such effect is not related to their own preference. However, modeling such user-independent timeliness effect is even harder due to the following problems: First, news is highly time-sensitive, leading to the difficulty of describing its lifecycle. Specifically, news can be updating fast with a short lifecycle, and can reach to be hot in a short time, but cool down rapidly [51, 55, 68, 70]. Therefore, it is not desirable to recommend the news if its heat is no longer growing or even declining. In Figure 1, although the "Olympic" news has accumulated a large number of clicks earlier, its low growth rate at present still reflects that it is no longer attractive. Moreover, some of breaking news articles are unforeseen and cannot find any clues previously in the whole records, e.g., there is no warning sign of "COVID-19" reported before 2020. Therefore, how to track the dynamics of news timeliness remains underexplored. In the literature, many efforts straightforwardly model such timeliness effect as the "popularity" from a general public perspective, where some additional features, such as the number of comments [60], ratings [31], or shares [54] are integrated as the indicators. However, these solutions are not the ideal metrics, as they can only reflect news characteristics from the static view and require much accumulation of historical data records, which obviously cannot well explore the timeliness effects of news sufficiently.

To address the above challenges, we propose a novel **Multi-Factors Fusion (*MFF*)** model for news recommendation by integrating both user-dependent preference effect and user-independent timeliness effect together. Specifically, for tracking the preference of a certain user, we decompose her reading historical records into two independent factors including the long-term habit and the short-term interest. We first aggregate all her browsed news to extract her persistent habit by exploring the news categories in all. Then, we try to characterize her temporary interests by proposing a recurrent neural network of analyzing the homogeneous relations between her latest clicked news and the candidate one, and therefore, her short-term interest can be well enhanced. For describing the timeliness effect of news, we consider each click behavior from any user as an event and assume the time elapsed between two adjacent events on the same news follows a particular distribution. Intuitively, the distribution of sensational news tends to have a low expectation. In other words, the sensational news is clicked more frequently over a fixed period of time, so the expected time duration between two clicks is shorter. Then, a component incorporating survival analysis techniques is designed to describe this distribution and estimate the instantaneous click probability so the news lifecycle could be well characterized. Last, we integrate all factor effects from user preference and news timeliness to determine the probability of the user clicking on a certain news under the independent event assumption. Moreover, since our model not only incorporates the user's personal effect but also the the general news effect, it can alleviate the cold-start problem of recommendation for new users.

We conduct extensive experiments on two real-world datasets. Our experimental results fully validate the effectiveness of *MFF* on not only the general recommendation scenario but also the cold-start recommendation scenario. In addition, we demonstrate the capacity of *MFF* on modeling the timeliness of news.

2 RELATED WORK

In this section, we summarize the related research work with three main categories, including recommender system from both the general scenario and news platform, news timeliness effect and survival analysis technique.

2.1 News Recommendation

Recommender system is one of the most influential techniques to help users make decisions, which could alleviate the information overload effectively in real life. It has been widely studied and applied in many real-world domains, such as e-commerce [5, 19, 41], social network [36, 74, 84], movie [48, 81], POI [53, 77], intelligent education [21, 22], marketing [79], and advertisement [85, 86]. The key behind the systems is to design a perception model that could track users' preferences, based on which recommends the suitable items (e.g., movie, book, question, location). From a general perspective, traditional recommendation algorithms could be divided into three rough categories, including content-based ones [81], collaborative filtering [6], and hybrid strategy [9]. Specifically, content-based models try to select items with similar content for users, where several content features including category, text, and so on, can be integrated by many feature engineering methods. Collaborative filtering perceives users' interests by assuming they share the same preference with neighbors who have similar behaviors, which produces representative methods including factorization models [56], ranking models [4], and so on. One step further, hybrid ones take advantage of both. Recently, deep learning-based techniques have been explored for recommender systems, which achieve much progress [14, 18, 35, 59, 63, 74]. For example, neural collaborative filtering learned higher-order user-item interactions for user preference learning [18]. Chen et al. explored deeper semantics for item content relationship learning [7]. Moreover, advanced techniques incorporate more data types for recommendation such as graph data [59, 65], media data [3], multi-modal data, [66] and so on. For example, Zhang et al. [74] propose a multi-graph-based model for social recommendation. Readers who are interested in general recommendation could refer to several surveys in the community [71].

For personalized news recommender systems, similar to general ones, accurately modeling users' interests would also be the basic task for generating the satisfactory news lists that are more in line with their tastes. Learning from the experiences in several domains above, some works perceive which news that users are satisfied with through analyzing their behaviors when browsing news on the platforms [30, 45, 46, 82, 87]. For example, Du et al. [13] and Zheng et al. [82] treated users' return time as a measure of user satisfaction so they employed the Poisson point process or Hawkes process to model user return time. Kim et al. [30] and Zhou et al. [87] noticed the dwell time could reflect whether or not the user was satisfied when reading the news, while Lu et al. [46] and Wu et al. [69] considered the factor of reading speed in recommender systems. However, these methods usually need additional side information describing users' behaviors (e.g., dwell time and return time), which is hard to accumulate in practice.

Different from general scenarios, news recommendation should pay more attention on how to explore and integrate news-specific information during the process, because users on news platforms always browse the news with attractive content such as title, and so on [64, 67]. Therefore, mainstream news recommendation approaches try to explore user-dependent preferences by aggregating all contents of news in her reading history. In recent years, by taking advantage of advanced deep learning and natural language processing techniques, several works try to recommend relevant news with similar semantics in the deep latent space based on users' reading histories [1, 42, 51, 58, 64, 67]. For example, Lian et al. [42] designed an inception network with the attention mechanism to automatically select and combine salient features extracted from users' records. An et al. [1] combined CNN and LSTM to better represent the historical clicked news and also proposed two ways to merge the user embedding into their model. Moreover, to enhance the performance of news semantics, some studies incorporate external knowledge from other data sources such as knowledge graph [34, 61, 64] or microblogs [11] into their frameworks so user preference could be deeply mined.

However, users on news platforms usually show their preferences with different factors. On one hand, they always select news articles by their persistent habit (e.g., reading "sport" news in Figure 1). On the other hand, they could drift their interests by some temporary factors (e.g., browsing "car" news in Figure 1). Therefore, different from most of existing works that mix up both user factors, in this article, we try to explore user-dependent preference one step further by distinguishing them into two independent factors, including the long-term habit and short-term interest, each of which can decide the users' news click behaviors simultaneously.

2.2 Timeliness of News

In real-world scenarios, a large number of news articles are emerging and updating worldwide every day, which reflects the strong timeliness characteristics [70]. Different from traditional scenarios above, users on news platforms can always be attracted by some breaking news without their preferences [57]. Therefore, it is necessary to specifically consider news timeliness effects for recommendation. However, as much literature indicates [51, 55, 68, 70], such timeliness of news remains great difficulty to be described. Specifically, news is highly time-sensitive, which can update fast with a short lifecyle, each of which can reach hot in a short time, but expires rapidly. Therefore, it is not desirable to recommend the news if its heat is not growing or even declining at a certain moment. Moreover, some breaking news articles are unforeseen and cannot find any clues previously in the whole records. Therefore, how to describe and track such timeliness dynamics is one of most important factors in news recommendation. In the literature, several works straightforwardly model the timeliness as the "popularity" from a general perspective, where several indicators are designed in various ways [24, 31, 43, 50]. For example, Naseri et al. [50] and Liao et al. [43] took the number of views as a measurement for news popularity, while Tatar et al. [60] and Tsagkias et al. [62] adopted the number of comments. In addition, some effective indicators can be described as a kind of explicit features in the news systems including the number of votes [37], ratings [31], and shares [54]. These works suggest that it is feasible to incorporate the simple "popularity" effect for news recommendations. Along this line, mainstream approaches try to manually design a bundle of popular related features [9, 26]. For example, Darvishy et al. [9] adopted the number of views and further defined the hotness of a news article as a kind of important feature. Jonnalagedda et al. [26] computed the popularity through the cosine similarity between the news and the related tweets.

Unfortunately, these straightforward "popularity" indicators usually require a long time and much effort to collect the data after posting the news, which is limited in practice. More importantly, the results only reflect the consistent news characteristics over a long period of time in the past from a static view, which cannot precisely describe the trend of news timeliness effect in time. To remedy this issue, in this article, we introduce the survival analysis technique to characterize the news timeliness as the time elapsed between two adjacent click events on the same news by any two users. Intuitively, the breaking news would attract more visitors so the expected time elapsed would be shorter. Then, we analyze the trend of click probability with respect to the time elapsed. Once we know the time elapsed since the latest click of one news piece, we can further predict whether the news will be clicked due to the timeliness of news.

2.3 Survival Analysis

Describing the click probability of a news article over a period of time in the future is a non-trivial task, because the click behavior (click event) does not always occur. Actually, for most instances, the exact time of the click is unobservable due to the limitation of observation period, which is called "censoring" [25]. In fact, the censoring phenomenon exists widely in different scenarios, such as employee turnover [17], customer churn [80], patient death [40]. If we want to analyze when someone with lung cancer would be die, it is necessary to collect a large number of data of patients who have died of lung cancer. However, at present, many patients are still struggling with or recovering from the disease. These cases lack the exact time of death and are called censored data. Therefore, directly learning on these instances will make the results unreliable [39]. To better estimate the probability of event occurrence at each time, a key technique addressing the censoring phenomenon is survival analysis. In this domain, there are two main streams. The first view is based on traditional statistic theories [39, 73]. These methods heavily depend on pre-assumed distributional forms for the survival rate function. The second view is based on machine learning perspective, including SVM [28], multi-task learning [78], and deep learning [33, 72, 83]. Survival analysis has been applied to various application fields, such as check-in location prediction [72], donation recurrence and retention in the crowdfunding area [78], fraud early detection in online platforms [83], notifications pushing for mobile applications [73]. In the news recommendation area, this technique is also employed to predict the return time of users [13, 82].

In our study, we define a news click as an event so the news article that is exposed to users but not clicked on can be considered as censored data. To fully utilize censored data, we take advantage of survival analysis techniques to describe the click probability on news with respect to its time elapsed. Specifically, a breaking news will lead to a high probability of a click in a short period of time. In this way, we can predict whether the news will be clicked at a future time. To the best of our knowledge, this is the first attempt to extend survival analysis to the click event prediction in news recommendation.

Table 1. Several Key Mathematical Notations

Notations	Type	Description
U	scalar	the number of users
N	scalar	the number of news articles
T	vector	a sequence of word indexes denoting the title of the news
c	scalar	a word index denoting the tag of the news
t	scalar	the timestamp when the click occurs
Δt	scalar	the time elapsed since the latest click by any user
$\boldsymbol{C}$	matrix	the embedding matrix of all tags with $\|C\|$ rows
$\boldsymbol{W}$	matrix	the embedding matrix of all words with $\|W\|$ rows
$\boldsymbol{U}$	matrix	the embedding matrix of all users with $\|U\|$ rows
$f(t)$	function	click probability density function denoting the probability when the click event occurs at time t
$S(t)$	function	click survival function denoting the probability of the click event having not occurred by time t
$\lambda(t)$	function	click hazard function denoting the instantaneous click probability at time t given the click event does not occur before

3 PROBLEM DEFINITION

We formally define our problem as follows: In general, suppose there are U users and N news in an online news platform. For a given user u, we denote her click history as $[n_1^u, n_2^u, \ldots, n_{N_u}^u]$, where $n_i^u(i \in \{1, \ldots, N_u\})$ is the ith news clicked by user u, and N_u is the total number of user u's clicked news. For each news n_i^u, it is composed of a triple, i.e., $n_i^u = (T_i^u, c_i^u, t_i^u)$. Specifically, T_i^u is a title that consists of a sequence of words, i.e., $T_i^u = [w_{i1}^u, w_{i2}^u, \ldots]$. c_i^u is a tag of the news representing its category (e.g., Movie). t_i^u is a timestamp when the user u clicks on the news. Then, given a user's click history and a piece of candidate news n_{cand} with its title T_{cand} and tag c_{cand}, our goal is to predict whether she will click n_{cand}, which has not been seen by her before.

In the following sections, for convenience, we will omit the superscript u. Bold letters denote the matrices or vectors, whereas non-bold letters denote scalars. For better illustration, we summarize several key mathematical notations in Table 1.

4 METHODOLOGY

In this section, we introduce our model. Figure 3 illustrates the graphical architecture, which consists of two key components, including user-dependent preference module and user-independent timeliness module. Moreover, we design a news encoder to extract the news content semantics initially and propose to integrate both effects for news recommendations. In the following, we explain the model techniques in detail.

4.1 News Encoder

In the online news platforms, users can be easily attracted by some concise but informative descriptions of news first, such as the news title and the corresponding category tag, and then decide to search the whole news body. For simplicity in the modeling, we learn the news semantic meaning with considering its title and tag information. Please note that it can be easy to generalize to other news information like the body content. In this subsection, without loss of generality, we utilize the same notations (i.e., T, c) to denote the title and tag whatever they belong to candidate news or clicked news.

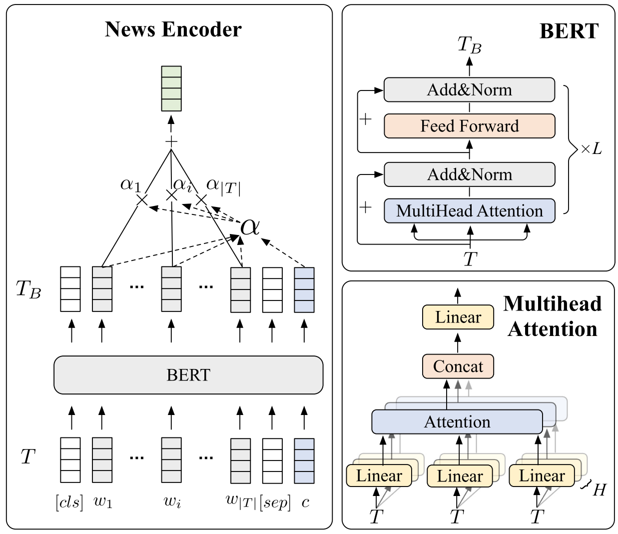


Fig. 2. Details of the news encoder.

Initially, we convert the tag c and each word w_i in the title into the dense vector ($\boldsymbol{c}$, $\boldsymbol{w}_i$) via embedding matrices $\boldsymbol{C} \in \mathbb{R}^{|C| \times D}$ and $\boldsymbol{W} \in \mathbb{R}^{|W| \times D}$, where $|C|$ denotes the number of tags, $|W|$ denotes the word vocabulary size, and D denotes the dimension of the embedding. After that, we obtain a sequence of word embeddings for the title. In this article, we propose a novel news encoder to aggregate both the title and tag together for modeling the news semantics, where the encoder architecture is shown in Figure 2. Here, we adopt BERT, which has shown the dominant performance in various natural language processing tasks, including token tagging, span prediction, and so on [12, 23, 76].[3] Specifically, it can model the global complex relations in a sentence, where much valuable information from a large volume of unlabeled data through the pre-training stage can also be captured. Learning from this experience, we adopt BERT model as the backbone, where we make the modification to put the tag c along with the title T as the input so the tag semantics can be adapted to the different news articles. Then, we propose a tag-aware attention mechanism to aggregate all word vectors into one news representation with category information.

Mathematically, as shown in Figure 2, given the title word sequence of a piece of news $T = [w_1, w_2, \ldots]$ with its corresponding category tag c, they are stacked with two special embeddings indicating the start and stop tokens (i.e., $\boldsymbol{T} = [[\boldsymbol{cls}], \boldsymbol{w}_1, \ldots, \boldsymbol{w}_{|T|}, [\boldsymbol{sep}], \boldsymbol{c}] \in \mathbb{R}^{(|T|+3) \times D}$, where $|T|$ is the length of the title). Then, they are fed into BERT to learn context-aware vectors. Formally, multi-head attention layer computes output matrix as:

$$
\begin{aligned}
&MultiHead(\boldsymbol{T}) = (\boldsymbol{head}_1 \oplus \cdots \oplus \boldsymbol{head}_H)\boldsymbol{W}^O,\\
&\boldsymbol{head}_j = Attention\left(\boldsymbol{T}\boldsymbol{W}_j^Q, \boldsymbol{T}\boldsymbol{W}_j^K, \boldsymbol{T}\boldsymbol{W}_j^V\right),\\
&Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\boldsymbol{Q}\boldsymbol{K}^\top/\sqrt{D})\boldsymbol{V},
\end{aligned}
$$

where $\{\boldsymbol{W}^O, \boldsymbol{W}_j^Q, \boldsymbol{W}_j^K, \boldsymbol{W}_j^V\}$ are projection matrices, $\oplus$ is the concatenation operation, H is the number of attention heads. Here, our attention mechanism is different from the one in BERT. Specifically, each word in the news title is forbidden to see the tag, since it is supposed to focus on the content of title. On the contrary, the tag can interact with all words so its representation

[3]Please note that we do not emphasize the difference among BERT-based models.

can be adapted to the different news articles. Besides, the Feed Forward and Add&Norm layers are calculated as:

$$FF(\boldsymbol{x}) = \boldsymbol{W}^{F2} max(\boldsymbol{0}; \boldsymbol{W}^{F1}\boldsymbol{x} + \boldsymbol{b}^{F1}) + \boldsymbol{b}^{F2},$$
$$Add\&Norm(\boldsymbol{x}) = LayerNorm(\boldsymbol{x} + Sublayer(\boldsymbol{x})),$$

where $\{\boldsymbol{W}^{F1}, \boldsymbol{W}^{F2}\}$ and $\{\boldsymbol{b}^{F1}, \boldsymbol{b}^{F2}\}$ are weight matrices and bias vectors, respectively. $Sublayer(\boldsymbol{x})$ is the function implemented by the sub-layer itself (i.e., multi-head attention or feed forward).

After stacking the above operations L times, we finally get a sequence of context-aware word embedding (i.e., $\boldsymbol{T}_B = \text{BERT}(\boldsymbol{T})$). Generally, to obtain the news representation, an intuitive way is to aggregate all embeddings in $\boldsymbol{T}_B$ together except $[cls]$ and $[sep]$ through an average pooling operation. However, it is obvious that all words in the news are not equally important and they should be treated differently. Actually, in reality, the news tag can often help us recognize the significant words. For example, when we mention *Iron man*, we know it is a character in the movie. To this end, we propose a tag-aware attention mechanism to learn a news representation. Given the context-aware word embeddings and the tag embedding (i.e., $\boldsymbol{T}_B$), we compute the tag-aware news representation as:

$$\boldsymbol{s} = \sum_{\boldsymbol{w}_i \in \boldsymbol{T}_B \setminus \{[\boldsymbol{cls}], [\boldsymbol{sep}], \boldsymbol{c}\}} \alpha_i \boldsymbol{w}_i, \tag{1}$$

$$\alpha_i = \frac{exp(\boldsymbol{w}_i^\top \boldsymbol{c})}{\sum_{\boldsymbol{w}_j \in \boldsymbol{T}_B \setminus \{[\boldsymbol{cls}], [\boldsymbol{sep}]\}} exp(\boldsymbol{w}_j^\top \boldsymbol{c})}. \tag{2}$$

Finally, we acquire a news representation $\boldsymbol{s} \in \mathbb{R}^D$ that has captured the deep semantic meaning of news. Please note that our news encoder can be easily extended to model any useful information of news such as news body content and even the users' reviews.

4.2 Modeling User-dependent Preference

As mentioned in Section 1, making decisions on reading what news for users is always driven by their personal preference, which is one of the important internal factors. In our model, we distinguish this user-dependent preference into two parts including the long-term habit and the short-term interest, which is shown in the left part of Figure 3. In this subsection, we introduce each of the technical details.

4.2.1 Long-term Habit. Given a user's reading records, the long-term habit can be equivalent to the prior knowledge indicating what kinds of news she likes to read persistently. For example, as shown in Figure 1, we can conclude the user likes to read news articles about "sports" most, followed by "movies," since she always reads the relevant news articles in the history. Therefore, it is reasonable to recommend a piece of news about *NBA* to her. In most existing studies [1], a general way is to optimize an embedding matrix that represents the long-term habits of different users. However, this approach can not well model the users' prior habit knowledge. To this end, we propose a tag-aware user embedding to perceive the user's long-term habit. Specifically, we first look up her latent factors $\boldsymbol{u}$ via a user embedding matrix $\boldsymbol{U} \in \mathbb{R}^{|U| \times D}$, where $|U|$ denotes the number of users. Then, we aggregate her clicked news in history and obtain the tag distribution $\boldsymbol{l} \in \mathbb{R}^{|C|}$ showing her long-term habit w.r.t. categories. Finally, we obtain a tag-aware user embedding $\boldsymbol{u}_l$ as:

$$\boldsymbol{u}_l = \boldsymbol{u} + \boldsymbol{C}^\top \boldsymbol{l}, \tag{3}$$

where $\boldsymbol{C}$ is the tag embedding matrix.

Next, to decide which news the user would read, given the candidate news n_{cand}, we first get its news representation $\boldsymbol{s}_{cand}$ through the news encoder. Then, we predict whether the user will

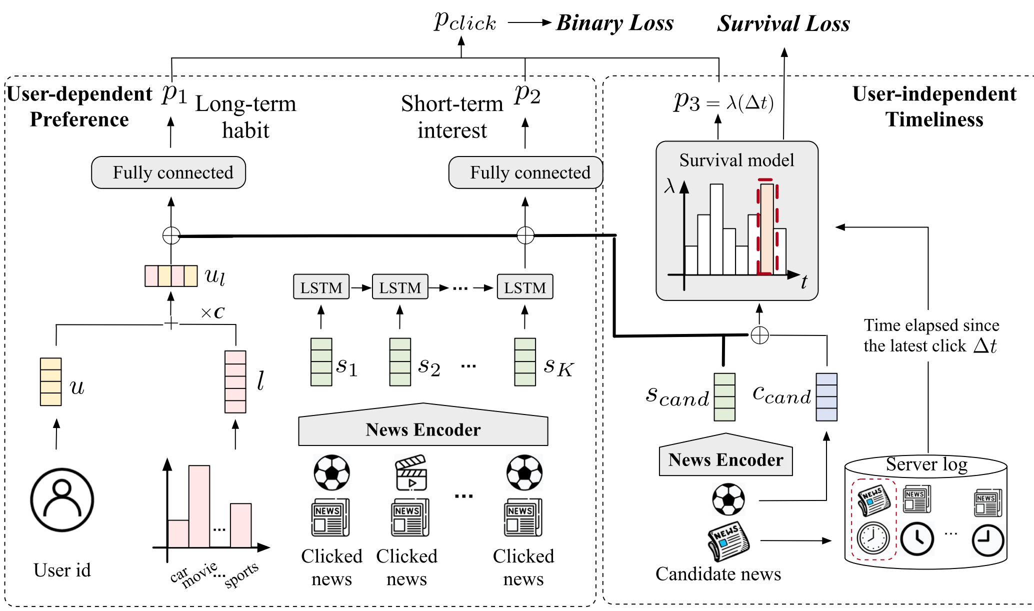


Fig. 3. The graphical architecture of *MFF*. The left component is designed to model the user-dependent preference. The right component is designed to model the user-independent timeliness.

click on this news due to this long-term habit of the user, where the probability is defined as:

$$p_1(click|n_{cand}, u) = \sigma(\mathbf{W}_1(\boldsymbol{u}_l \oplus \boldsymbol{s}_{cand}) + b_1), \tag{4}$$

where $\{\boldsymbol{W}_1, b_1\}$ are the weight vector and bias. $\sigma(\cdot)$ is the non-linear activation function, which is stated as the $sigmoid(\cdot)$ in this article.

4.2.2 Short-term Interest. In addition, in reality, a user is easier to change the preference due to some possible temporary demands. That is to say, she usually likes to read similar articles with same categories in a certain short period of time. From the illustrative example in Figure 1, the user goes through many "car" articles on her latest records (as she may have the plan to buy a car in the near future) although she seldom pays the attention to the relevant news in the earlier time. Thus, she would be probably interested in some related news that is different from her habit. To model this factor, we have to explore homogeneous relations between the candidate news and her latest clicked news. Different from previous works [1, 75] that take all pieces of clicked news into account, we argue that only a set of them would contribute the most to this factor. Given the K latest clicked news of a certain user, we feed them into the news encoder and get K news representations denoted as $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_K\}$. Then, we provide a sequential encoding model to learn the relations of news that users read in the latest K times. The idea can be implemented by many models [14, 67, 75], and in this article, since we do not emphasize their differences, we implement it with one of the most commonly used LSTM [15, 44]. Specifically, given the ith news representation $\boldsymbol{s}_i (i = 1, \ldots, K)$, $(i-1)$-th memory cell $\boldsymbol{z}_{i-1} \in \mathbb{R}^D$ and hidden state $\boldsymbol{h}_{i-1} \in \mathbb{R}^D$, the ith memory cell $\boldsymbol{z}_i \in \mathbb{R}^D$ and hidden state $\boldsymbol{h}_i \in \mathbb{R}^D$ are computed as:

$$\boldsymbol{z}_i, \boldsymbol{h}_i = LSTM(\boldsymbol{s}_i, \boldsymbol{z}_{i-1}, \boldsymbol{h}_{i-1}; \boldsymbol{\theta}), \tag{5}$$

where $\boldsymbol{\theta}$ is the parameters in LSTM. Then, we treat $\boldsymbol{z}_K$ representing the user's short-term interest.

Similar to Equation (4), we can predict the click probability on the candidate news driven by this short-term interest factor as:

$$p_2(click|n_{cand}, n_1, \ldots, n_K) = \sigma(\boldsymbol{W}_2(z_K \oplus \boldsymbol{s}_{cand}) + b_2), \tag{6}$$

where $\{\boldsymbol{W}_2, b_2\}$ are the weight vector and bias.

4.3 Modeling User-independent Timeliness

In addition, news is emerging and updating rapidly every day, which reflects several unique characteristics. Therefore, users can always be attracted by the ones that are driven by many other user-independent factors. As mentioned in Section 1, timeliness is one of the most significant factors describing news lifecycle [51, 68, 70]. Recalling the example in Figure 1, the user is now consistently concerned about the change of the "COVID-19" pandemic and "Olympics2020." However, these two news reflect different timeliness stage at present. Specifically, the "COVID-19" news at present boosts a rapid growth attention, while the "Olympics2020" news is no longer hot and may disappear in the near future. Therefore, it is better to choose the pandemic news for recommendation at present and, however, it is difficult to describe the news timeliness effect. Moreover, some of the breaking news articles, e.g., "COVID-19," are unforeseen previously and cannot find any clues in the whole records, which even exacerbates the modeling difficulty. To characterize this timeliness factor, most of existing solutions straightforwardly describe its effect as the "popularity" metric and design many indicators that consist of some additional features collected in the systems, such as the number of comments [60], ratings [31], or shares [54]. However, this is not an ideal solution, since the popularity indicators only reflect the news timeliness from a static view and require a long period of time with data accumulation for statistics. Therefore, they cannot well explore the news timeliness effect sufficiently, since they fail to describe the news lifecycle at a certain time. To this end, we propose an innovative way to model the timeliness of news that is shown in the right part of Figure 3. Particularly, considering the "breaking" news is clicked far more than general news over a fixed period of time, the average time elapsed between two adjacent clicks is smaller. Based on this idea, we attempt to perceive the timeliness of a news article according to the time elapsed since the latest click by any user with the help of survival analysis techniques [2], which aim to describe the instantaneous click probability over time.

Specifically, survival analysis is a sub-field of statistics that is qualified for predicting the probability of the occurrence of an event in a future time as well as estimating the time duration until one event occurs. It is originally applied to the medical field for analyzing when a biological organism dies [40]. In our study, formally, for each candidate news n_{cand}, we consider each click behavior from any user as an event and assume the time elapsed between two adjacent events on the same news follows a particular distribution. Its expected value becomes small if the news is much appealing to users and becomes large otherwise. We define the click time $\hat{T}$ as a continuous random variable indicating the waiting time until the occurrence of click since the latest click behavior from any other user, with click probability density function $f(t) = \lim_{dt \to 0} \frac{P(t \leq \hat{T} < t+dt)}{dt}$. The click survival function $S(t)$ indicates the probability of the click event having not occurred by time t:

$$S(t) = P(\hat{T} \geq t) = \int_t^{\infty} f(x)dx. \tag{7}$$

The click hazard function $\lambda(t)$ refers to the instantaneous click probability at time t given click behavior does not occur before:

$$\lambda(t) = \lim_{dt \to 0} \frac{P(t \leq \hat{T} < t + dt | \hat{T} \geq t)}{dt} = \frac{f(t)}{S(t)}. \tag{8}$$

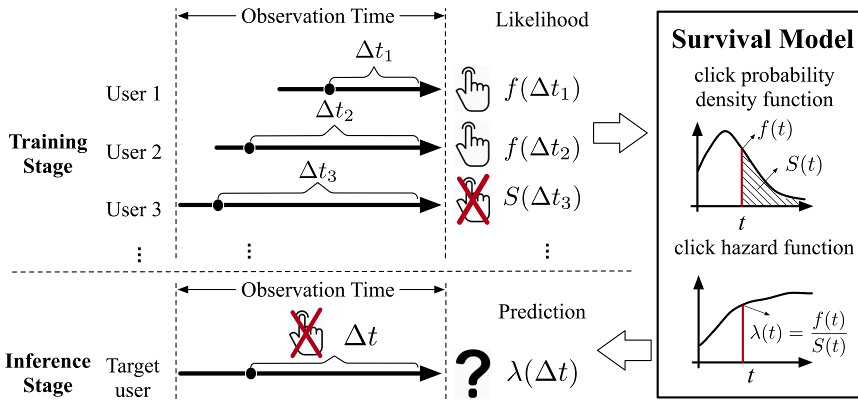


Fig. 4. An illustrative example for training a survival model that is used to inference the click probability due to the timeliness. During the training stage, for a positive instance, its click probability (e.g., $f(\Delta t_1)$ or $f(\Delta t_2)$) should be maximized, while for a negative instance, its accumulative probability (e.g., $S(\Delta t_3)$) should be maximized. During the inference stage, conditioning on the fact that the candidate news has not been clicked in a while, the click probability should be predicted based on the click hazard function.

In this article, the timestamp is discrete, so we approximate the click hazard function as:

$$\lambda(t) = P(\hat{T} = t | \hat{T} \geq t). \tag{9}$$

In practice, we first collect logs of the candidate news from all users in the past. Then, we find the latest click record and compute the time elapsed. After that, the time elapsed is discretized into several pieces where the span of each piece is denoted as ΔT. We also mark the corresponding index of each time piece as Δt. Please note that ΔT is an important hyper-parameter that could affect the performance of modeling the news timeliness effect. Specifically, ΔT describes the interval size of time piece, which can determine the time range in which to predict whether the news will be clicked. Therefore, if we find one news has a strong timeliness with a short lifecycle, then we should set ΔT with a small value that could track the news dynamics more accurately. We will make some analysis in the Section 5.5.2. Given the features of a candidate news (i.e., its representation $\boldsymbol{s}_{cand}$, tag representation $\boldsymbol{c}_{cand}$), we attempt to estimate the instantaneous rate for each time slot since the latest click. Formally, we have:

$$\boldsymbol{\lambda} = \sigma(\boldsymbol{W}_3(\boldsymbol{s}_{cand} \oplus \boldsymbol{c}_{cand}) + \boldsymbol{b}_3), \tag{10}$$

where $\{\boldsymbol{W}_3, \boldsymbol{b}_3\}$ are the weight matrix and bias vector. Each element in $\boldsymbol{\lambda}$ indicates the instantaneous click rate in a short period of time, which means $\boldsymbol{\lambda} = \lambda(0) \oplus \lambda(1) \oplus \cdots$. As shown in the bottom part of Figure 4 (i.e., Inference stage), suppose we have a well-trained survival model (training details can be found in Section 4.4), conditioning on the fact that the candidate news has not been clicked in a while and time slot index is Δt, the click probability due to the timeliness can be inferred by the click hazard function:

$$p_3(click | n_{cand}) = \lambda(\Delta t). \tag{11}$$

4.4 Model Fusion and Training

In this subsection, we will illustrate how to fuse the user-dependent preference and user-independent timeliness together for making the final recommendations. Then, we introduce the objective function of how to train our proposed model.

In Sections 4.2 and 4.3, we have obtained the click probabilities under three factors, including long-term habit p_1 (Equation (4)), short-term interest p_2 (Equation (6)), and news timeliness p_3

(Equation (11)). Following the intuition, the model should integrate them in all for news recommendations. To achieve this goal, a general way is to get their geometric mean as the overall click probability:

$$p_{click} = \sqrt[3]{p_1 p_2 p_3}. \tag{12}$$

However, in this article, we assume a user will click the candidate news due to any one of the three factors, which follows the independent event assumption. Formally, we define the overall click probability as:

$$p_{click} = 1 - (1 - p_1)(1 - p_2)(1 - p_3). \tag{13}$$

For each instance, it has a piece of candidate news, a user with her click history, and the label y (y equals 1 if she clicks the news and equals 0 otherwise.). To reduce the empirical risk, a widely used objective function is to minimize the cross-entropy (which is also called the binary loss) as:

$$\mathcal{L}_{bin} = -y \log(p_{click}) - (1 - y) \log(1 - p_{click}). \tag{14}$$

However, we empirically find that the binary loss cannot optimize the click probability density function $f(t)$ closer to the real distribution. To address this issue, we also propose a novel survival loss. As shown in the top part of Figure 4, we adopt maximum likelihood estimation to maximize the $f(t)$ for the positive instances (i.e., $y = 1$) and $S(t)$ for the negative instances (i.e., $y = 0$), since the exact click time of negative ones has not been observed. For example, as shown in the top part of Figure 4 (i.e., Training stage), a piece of news is exposed to three users at different time, and we find *User 1* and *User 2* click on this news while *User 3* does not. We search the adjacent click event by others in the observation time, which represents the time window before the click/unclick action. Then, time duration of them is calculated and denoted as $\Delta t_1, \Delta t_2, \Delta t_3$. For *User 1* and *User 2*, since their click behaviors have been observed, we only need to maximize the likelihood of $f(\Delta t_1), f(\Delta t_2)$. For *User 3*, it is unreasonable to maximize the likelihood of $f(\Delta t_3)$, because she does not click the news at that moment. We assume the click event of *User 3* on this news will occur in the future. Therefore, the best choice is to maximize the survival function $S(\Delta t_3)$. With respect to the formulations of $f(t), S(t)$, formally, we first derive the equations of $f(t)$ and $S(t)$ with respect to $\boldsymbol{\lambda}$ in the form of discretization from (Equations (7) and (8)), and then define the logarithm of survival loss with minimization as:

$$f(t) = \lambda(t) \exp\left(-\sum_{x=0}^{t} \lambda(x)\right), \tag{15}$$

$$S(t) = \exp\left(-\sum_{x=0}^{t} \lambda(x)\right), \tag{16}$$

$$\mathcal{L}_{sur} = -y \log(f(\Delta t)) - (1 - y) \log(S(\Delta t)). \tag{17}$$

Combining $\mathcal{L}_{bin}$ (Equation (14)) and $\mathcal{L}_{sur}$ (Equation (17)), given M instances, our overall objective function with minimization is defined as:

$$\mathcal{L} = \min_{\Theta} \sum_{i=1}^{M} \left(\mathcal{L}_{bin}^{i} + \gamma \mathcal{L}_{sur}^{i}\right), \tag{18}$$

where γ is a coefficient to balance two losses, Θ denotes all parameters in our *MFF* updated by Adam optimization algorithm.

Table 2. Basic Statistics of Two Datasets

Statistics	Toutiao	Adressa
Number of users	50,000	31,596
Number of news	731,612	6,128
Number of category tags	146	48
Number of logs	2,123,700	985,329
Positive and negative ratio	≈3:4	-
Avg. time elapsed (seconds)	13,758.8	1,840.1
Avg. number of words per title	23.7	7.1

5 EXPERIMENTS

In this section, we first introduce two real-word datasets we used and show some basic statistics and distributions. Then, we illustrate the experimental setup and baselines in detail. Finally, we conduct extensive experiments and report the results from different perspectives.

5.1 Datasets Description

We conduct experiments on two real-world datasets for evaluation and describe them as below.

- *Toutiao* is a dataset supplied by Bytedance Co., Ltd and is collected from its server logs of Toutiao. Specifically, for each log, it contains a user ID, a news ID with a category tag and a title, a timestamp, and a label indicating whether or not the user clicks on the news. Since the total number of users is too large, we randomly select 50,000 active users and collect their logs in one week from May 1st, 2019, to May 7th, 2019. We take the top 90% of the data in chronological order as a training set and the rest as a test set.
- *Adressa*[4] is another news dataset that is constructed by Reference [16] from Adressavisen, a Norwegian news portal. Different from Toutiao, Adressa only contains the records of users's clicks on different news. Therefore, for each log, it contains a user ID, a news ID with a tag and a title, a timestamp when the user clicks on the news. To keep high-quality users, we filter out those users whose records are less than 10. As a result, 31,596 users are left. Following References [63, 75], we adopt the leave-one-out strategy. For each user, we hold out her latest interaction as the positive test instance and randomly sample 99 news articles that are not interacted by the user as the negative test instances. In addition, we utilize the remaining data for training as positive instances using a sliding window, each of which samples 4 negative instances.

We summarize the basic statistics of both datasets in Table 2. We also deeply analyze some data analyses of them from the following perspectives: First, for each user, we calculate the portion of news categories she clicked on in the test set that appears in the news categories she clicked on in the training set. The average results of all users on Toutiao and Adressa are 76.36% and 76.48%, respectively. That means almost the three-fourths of news categories are the same in the training and test sets, which demonstrates that users are willing to click on the news followed by their long-term habit with category factor. Second, we analyze the correlation of users' news browsing records. Specifically, for each user, we select two groups of news in her training set including one "Near Group" consisting of her latest five clicked news and the other "Early Group" consisting of her first five clicked news. Taking one news she clicks in the test set, we compute the news content

[4]We use the light version in http://reclab.idi.ntnu.no/dataset/.

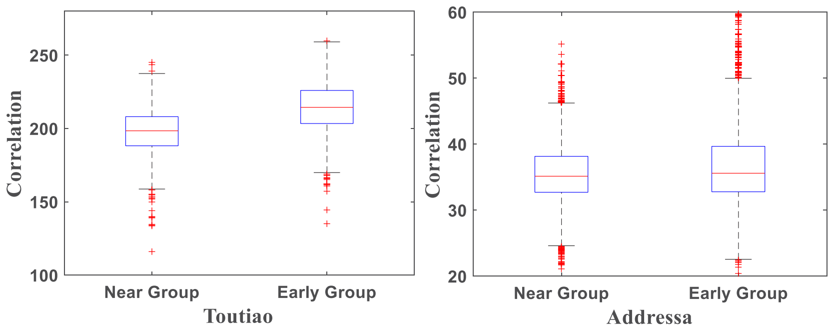


Fig. 5. Correlation comparison of news log records of users on both datasets.

correlations by dot similarities of it and the news in both groups. Figure 5 reports the correlation comparison result of all user log instances on different groups in box figures. From the figure, news in test set is more relevant to the news in "Near Group" than that in "Early Group." This observation could demonstrate that user preferences can be more susceptible to recent records rather than earlier histories, which demonstrates the rationality of our short-term interest idea. Third, we summarize the distributions in Figure 6. Specifically, the top two charts show the distributions of the time elapsed between two adjacent clicks on the same news in two datasets. Since the number of news in Toutiao is much larger than that in Adressa, users in Toutiao have more choices when reading news. As a consequence, the click frequency of each news in Toutiao is lower than that in Adressa, so the average time elapsed in Toutiao is much higher (13,758.8 seconds versus 1,840.1 seconds). The middle two charts illustrate the top 10 category tags of news distributions in two datasets, which show the similar patterns. In the bottom two charts, we demonstrate the distributions of the number of title words. The average number per title is 23.7 and 7.1 words, respectively, showing that average length of title in Toutiao is about three times as long as that in Adressa. These findings show that news is highly time-sensitive with short lifecycle.

5.2 Experimental Setup

5.2.1 Parameter Setting. We implement our model *MFF* based on TensorFlow. We now specify some hyper-parameters. In the news encoder (Figure 2), following Reference [12], we set the hyper-parameters in our BERT module with the same setting with BERT_{BASE} except the number of layers (i.e., L), which equals 6 rather than 12 to improve computing efficiency. Moreover, some important hyper-parameters would affect the performance of our model, including the dimension of word embedding D, the time span ΔT, the number of candidate news K (Equation (6)), and the coefficient γ (Equation (18)). We will discuss the sensitivity of them in the Section 5.5.2. Last, for our model training, we set the learning rate as 2e-5 and mini-batch as 64. We utilize dropout with probability 0.2 to prevent overfitting.

To make our news encoder capture deep semantics from the news title, we first pre-train the BERT parameters via two tasks as Reference [12] does. Since the languages for both datasets are different, parameters in BERT should be pre-trained via large-scale corpus in corresponding languages. For Toutiao, 1 billion sentences are crawled from websites for pre-training stage. For Adressa, due to the lack of existing corpus in Norwegian, we directly utilize the pre-trained model provided by Google,[5] which includes 104 languages. The rest of parameters are randomly

[5] https://github.com/google-research/bert.

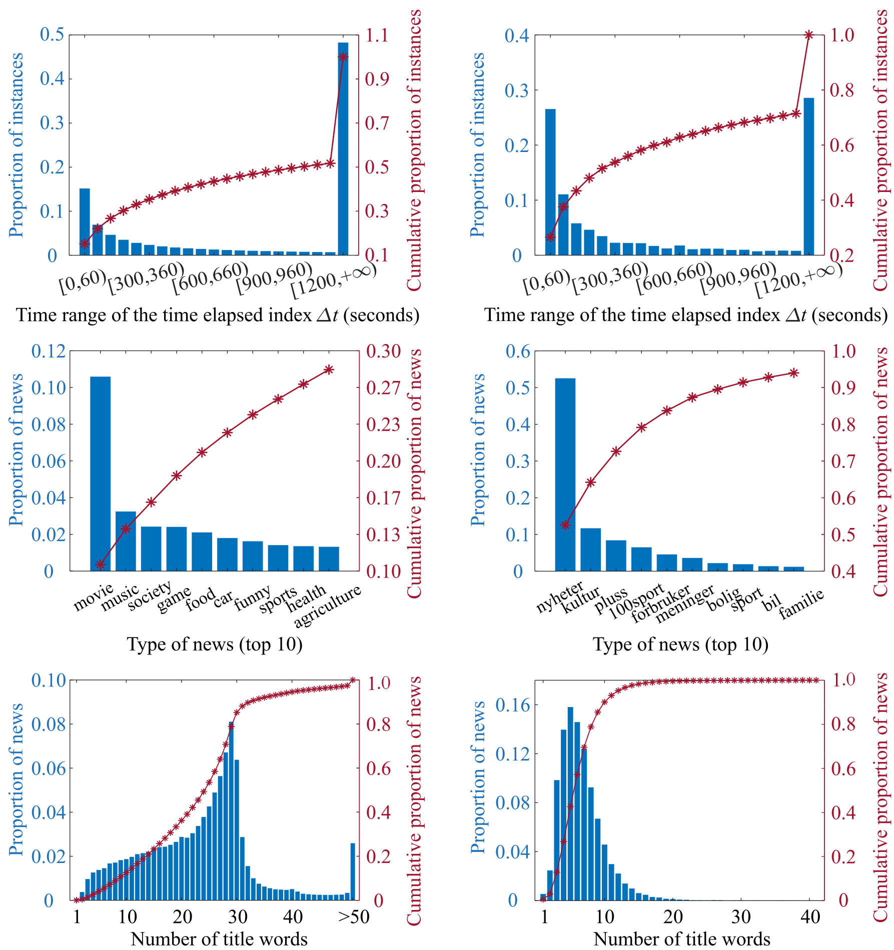


Fig. 6. Statistical distributions. Left: Toutiao; Right: Adressa. Top: distributions of the time elapsed of log instances; Middle: distributions of different types of news; Bottom: distributions of the number of title words.

initialized with a Xavier uniform initializer [49]. Then, all parameters Θ in our model are fine-tuned through the training stage.

5.2.2 Evaluation Metrics. In the experiments, we adopt four widely used metrics including AUC, $F1$, MRR, and $NDCG@5$. When computing AUC and $F1$, we treat all instances as independent ones. For each log instance i, we assume its real label and our predicted score are y_i and p_i, respectively, so AUC and $F1$ are formulated as:

$$AUC = \frac{|\{(i,j)|y_i = 1, y_j = 0, p_i > p_j\}|}{|\{i|y_i = 1\}||\{j|y_j = 0\}|},$$
$$F1 = \frac{2 * precision * recall}{precision + recall}.$$

Different from AUC and $F1$, MRR and $nDCG@5$ are calculated on a per-user basis. Assuming there are N users, each of which has several instances, we rank instances of each user by their predicted scores. In addition, the real label and predicted score of jth instance for ith user are, respectively, denoted as $y_{i,j}$ and $p_{i,j}$. *MRR* and *NDCG@5* are formulated as:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\min_{y_{i,j}=1} j},$$

$$NDCG@5 = \frac{1}{N}\sum_{i=1}^{N}\frac{\sum_{j\le 5} y_{i,j}/\log_2(j+1)}{\max_{\pi}\sum_{j\le 5} y_{i,\pi(j)}/\log_2(\pi(j)+1)},$$

where π is an arbitrary permutation of the rank list. Note that all metrics are the larger, the better.

5.3 Baselines

To validate the effectiveness of *MFF*, we compare it against 11 popular methods, which are divided into four groups.

Feature engineering–based methods usually focus on building a set of features to better represent the news. Then, some classic machine learning models are employed to predict the click probability. Particularly, we choose *Pop* [52], *SVD* [6], *FM,* and *FM+* [56] as baselines.

- *Pop* is a popularity model that recommends a set of news with higher click frequency. Since news is highly time-sensitive, it is a competitive baseline.
- *SVD* is a classic collaborative filtering model widely used in recommender systems. Each instance only consists of user ID and candidate news ID.
- *FM* is a feature-based factorization model. The input features of news consist of TF-IDF features from its title and one-hot vector of its tag. We treat all clicked news as one piece of news and then concatenate the features of candidate news and clicked news to feed into FM.
- *FM+* improves *FM* by adding the popularity features. Specifically, besides the features fed into *FM*, we also input the time elapsed since the latest click of the candidate news as one of the typical signal features of popularity.

Survival analysis methods aim to predict the occurrence of specific event (i.e., the click event) at a future time. Note that this kind of model predicts the click probability of a certain news during a period of time so the recommended news is not personalized. Particularly, we choose *COX* [8] and *DeepHit* [33] as baselines.

- *COX* is the most commonly used semi-parametric model in survival analysis. It can predict the probability of the news being clicked.
- *DeepHit* is another survival analysis model that adopts the deep neural network to construct the relationship between the click behavior and the covariates.

Content-based methods are based on the description of the news. These methods are suitable for news recommendation, since users usually are attracted by the title of news. Particularly, we choose *CNN* [29] and *DSSM* [20] as baselines.

- *CNN* is a typical convolutional neural network with max pooling to learn a news representation from its title by keeping most salient features.
- *DSSM* is a deep structured semantic model with word hashing via character trigram and multiple dense layers. All clicked news articles are concatenated as one piece of news, which is used to compute the similarity with candidate news.

Session-based methods are the mainstream methods in news recommendation during past few years. They capture the users' preferences and patterns from the sequence of their click history and then infer the click probability of the candidate news. Particularly, we choose *GRU4REC* [19], *DKN* [64], and *LSTUR* [1] as baselines. Note that *DKN* and *LSTUR* can also fall into content-based

Table 3. Performance Comparison of Different Models on Four Metrics Including *AUC*, *F1*, *MRR*, and *NDCG@5*

(a) **Toutiao**

	models	AUC	Imp	F1	Imp	MRR	Imp	NDCG@5	Imp
Feature Engineering	*Pop*	-	-	-	-	55.47	23.87	39.34	33.93
	SVD	58.42	11.86	57.54	2.87	62.13	10.59	45.51	15.78
	FM	59.23	10.33	57.36	3.19	62.81	9.39	46.39	13.58
	FM+	61.20	6.78	57.93	2.18	65.37	5.11	48.64	8.33
Survival Analysis	*COX*	51.12	27.84	56.86	4.10	58.86	16.73	42.01	25.42
	DeepHit	51.70	26.40	56.86	4.10	60.69	13.21	43.59	20.88
Content Based	*CNN*	58.59	11.54	57.44	3.05	64.46	6.59	47.60	10.69
	DSSM	60.37	8.25	57.63	2.71	64.20	7.02	47.15	11.75
Session Based	*GRU4REC*	54.60	19.69	56.88	4.06	59.83	14.84	43.32	21.32
	DKN	60.47	8.07	57.75	2.49	65.77	4.47	48.88	7.79
	LSTUR	62.13	5.18	57.99	2.07	64.14	7.13	47.53	10.86
Ours	***MFF***	**65.35**	-	**59.19**	-	**68.71**	-	**52.69**	-

(a) **Adressa**

	models	AUC	Imp	F1	Imp	MRR	Imp	NDCG@5	Imp
Feature Engineering	*Pop*	-	-	-	-	45.38	88.54	50.82	74.20
	SVD	95.43	3.84	34.42	104.42	51.83	65.08	56.67	56.22
	FM	97.65	1.47	49.01	43.56	69.64	22.86	74.89	18.21
	FM+	98.38	0.72	60.21	16.86	74.65	14.61	80.02	10.63
Survival Analysis	*COX*	88.66	11.76	60.35	16.59	70.15	21.97	70.96	24.76
	DeepHit	93.37	6.13	63.44	10.91	73.77	15.98	75.76	16.86
Content Based	*CNN*	98.06	1.05	51.24	37.31	70.64	21.12	75.89	16.66
	DSSM	97.90	1.22	49.04	43.47	71.84	19.10	76.90	15.12
Session Based	*GRU4REC*	89.63	10.55	36.28	93.94	54.12	58.09	56.18	57.58
	DKN	98.47	0.63	56.14	25.33	81.36	5.16	84.76	4.45
	LSTUR	98.28	0.82	54.08	30.10	77.32	10.66	81.55	8.56
Ours	***MFF***	**99.09**	-	**70.36**	-	**85.56**	-	**88.53**	-

Imp denotes the relative performance improvements. Note that *Pop* is a statistics metric without models, so it cannot be measured with both *AUC* and *F1* metrics (%).

models, since they explore the semantics of news titles, while *GRU4REC* only utilizes the news ID without any description.

- *GRU4REC* applies RNN for session-based recommendation. The model is fed a sequence of news ID clicked by the user and then predicts next news that is likely to be clicked.
- *DKN* is a deep news recommendation method with CNN and news-level attention mechanism. In addition, it incorporates entities derived from knowledge graph.
- *LSTUR* is a deep news recommendation method combining CNN and LSTM to jointly model user's long-term and short-term representations from the news title, tag, and user ID.

5.4 Experimental Results

5.4.1 Performance Comparison. In this experiment, we demonstrate the comparison results between *MFF* and baselines in Table 3. In addition to the comparative results, we analyze some potential limits and effective mechanism of the baselines.

Table 4. Performance Comparison between *MFF* and Its Variants on Four Metrics

	Toutiao				Adressa			
	AUC	***F1***	***MRR***	***NDCG@5***	***AUC***	***F1***	***MRR***	***NDCG@5***
MFF_L	61.91	58.20	65.65	49.90	98.09	52.18	71.98	77.08
MFF_S	63.23	58.38	66.29	49.81	98.31	57.11	80.98	84.49
MFF_T	64.38	58.49	68.51	52.54	98.49	**70.57**	84.73	87.60
MFF	**65.35**	**59.19**	**68.71**	**52.69**	**99.09**	70.36	**85.56**	**88.53**

MFF_L denotes the factor of the long-term habit. *MFF_S* denotes the factor of the short-term interest. *MFF_T* denotes the factor of timeliness (%).

- We can observe that our *MFF* model outperforms all baselines on both datasets. The results clearly indicate it can well capture the user-dependent preference effect and user-independent timeliness effect and then integrate all factors to benefit a more accurate recommendation.
- Among all baselines, *DKN* achieves the best in most cases. This is because *DKN* benefits a lot from the entities that are recognized with the help of knowledge graph. When reading a piece of news, the entities often convey a lot of information. Therefore, focusing on these entities can help the model generate a better representation of news title and lead to better recommendation performance.
- *GRU4REC* is not competitive compared with other content-based methods (CNN, DSSM), session-based methods (DKN, LSTUR), and ours. The reason is that *GRU4REC* only utilizes the news ID instead of its content to measure the similarity between different news. The results indicate that users can be attracted by news title content, where it is necessary to learn news content semantics (rather than just its ID indicator) for generating news representation, which is useful for news recommendation.
- Survival analysis-based methods perform poorly in our experiments, because they are not personalized. This extremely destroys the users' experience when browsing the news. Therefore, such methods can not directly apply to the real-world application.
- Last, we observe an interesting result that *FM+* is the most competitive model, except for our proposed MFF, which even outperforms several recent approaches (e.g., DKN, LSTUR). This is probably because it considers the news timeliness effect into the modeling, which demonstrates that considering timeliness effect is significant for news recommendation. Moreover, since our model directly describes the news lifecycle with a sophisticated survival analysis-based architecture, it outperforms *FM+* with only considering the news timeliness as the simple "popularity" metric.

5.4.2 Influence of Different Factors. Recall that our model captures the user-dependent preference and user-independent timeliness simultaneously for news recommendation, where we extract three factors for modeling, including long-term habit, short-term interest, and timeliness effect. In this experiment, we aim to illustrate the effectiveness of all three factors. To this end, we construct three variant models based on our MFF. Specifically, we denote *MFF_L* as variant only considering the architecture of long-term habit, *MFF_S* for the variant with only short-term interest, and *MFF_T* just with the timeliness part. Note that the modeling architectures for relevant factors are same as MFF does (recall Figure 3). Therefore, *MFF_L* and *MFF_S* only utilize binary loss to optimize parameters in the network (Equation (14)), while *MFF_T* utilizes both binary loss and survival loss (Equation (18)). The comparison results are reported in Table 4. Specifically, we have the following observations:

- *MFF* achieves the best in most cases. This shows leveraging three factors into a unified model can boost the prediction performance.
- Among three variants, we find *MFF_T* performs best and *MFF_S* ranks the second, followed by *MFF_L*. This suggests the timeliness of news is the most important factor, and our proposed module (i.e., User-independent Timeliness part in Figure 3) can capture this factor through the click hazard function (i.e., $\boldsymbol{\lambda}$ in Equation (10)).
- We can observe *MFF_L* does not perform well compared with the other variants. The reason for this is that *MFF_L* only utilizes the tag distribution and a fine-tuned user embedding (i.e., $\boldsymbol{l}, \boldsymbol{u}$ in Equation (3)) to model user preference without any content of users' click history. Although this approach can extremely reduce the amount of computation, it lacks a lot of potentially useful information. These results demonstrate the content of user's click history can benefit the prediction performance.

5.4.3 Cold-start Recommendation with New Users. As we mentioned in Section 1, news recommendation suffers from the cold-start problems with new users without any click records. Here, we illustrate the capability of *MFF* on such scenario. In this experiment, we also generate the training and test set by different strategies for the same reason as described in Section 5.1. For Toutiao, we only keep users in the test set who have never appeared in the training set. After the filtering, we have 4,083 users left. The ratio of positive and negative instances is approximately 0.65, which is lower than that in the entire dataset (i.e., 0.75 in Table 2). Therefore, it is more difficult to predict click rates for new users. For Adressa, since we adopt leave-one-out strategy, each user will appear in both the training set and test set. Therefore, we have to create some new users according to the custom rule. We randomly select 3,000 users as new users, forming the test set. Correspondingly, instances related to these new users in the training set are removed. Here, we assume these users access the news platform for the first time so their click history is missing. Therefore, there is less potential information to the user and it is harder to model user-dependent preference.

Experimental results of all methods on four metrics are reported in Table 5. Note that *SVD* and *GRU4REC* only involve news ID and user ID, so they are not capable of recommending news when the user does not have any click history. Here, we put all results in Tables 3, 5, and 6 together for analyses. First, compared with the results in common scenario (Table 3), we observe the performances of all methods have declined to varying degree in cold-start scenario (Table 5). Second, since there is no click history of the user, our model *MFF* (Table 5) degenerates to the variant model *MFF_T* (Table 6). However, despite the performance decreasing, *MFF* still dominates all other baselines on four metrics, respectively. These results demonstrate the effectiveness of *MFF* once again, especially on modeling news timeliness effect for recommendation. In addition, we find *FM+* outperforms two competitive baselines (i.e., *DKN*, *LSTUR*), especially on the Adressa dataset. This demonstrates that recommending a set of popular news, though a straightforward way of modeling news timeliness effect, is still a good choice when encountering new users.

5.5 Model Analysis

5.5.1 Click Probability Function Approximation. From the results and analysis from Tables 4 and 5, we can conclude the news timeliness factor produces the most significant effect (compared with the other two) in our model for dominating users' click and reading behaviors. In this experiment, we would demonstrate the capability of *MFF* on perceiving this factor. Specifically, as we illustrated in Section 4.3, we assume the time elapsed between two adjacent clicks on the same news by any two users follows a particular distribution. Then, we attempt to automatically learn this pattern with the help of survival analysis techniques, where we adopt the survival loss (i.e., $\mathcal{L}_{sur}$ in Equation (17)) to optimize the distribution. To verify that our *MFF* model has learned this

Table 5. Performance Comparison of Different Models on Cold Start Problem for New Users without Any Click History

(a) **Toutiao**

	models	*AUC*	*Imp*	*F1*	*Imp*	*MRR*	*Imp*	*NDCG@5*	*Imp*
Feature Engineering	*Pop*	-	-	-	-	54.30	26.13	38.48	36.15
	FM	57.67	10.63	57.34	1.69	62.37	9.81	46.23	13.32
	FM+	59.43	7.35	57.76	0.95	63.87	7.23	47.44	10.43
Survival Analysis	*COX*	50.67	25.91	56.57	3.08	59.19	15.71	41.85	25.19
	DeepHit	51.56	23.74	56.58	3.06	60.92	12.43	43.52	20.38
Content Based	*CNN*	58.08	9.85	57.20	1.94	64.21	6.67	47.57	10.13
	DSSM	58.23	9.57	57.00	2.30	63.91	7.17	46.69	12.21
Session Based	*DKN*	59.40	7.41	57.53	1.36	64.69	5.87	47.87	9.44
	LSTUR	58.31	9.42	57.41	1.57	62.00	10.47	46.13	13.57
Ours	***MFF***	**63.80**	-	**58.31**	-	**68.49**	-	**52.39**	-

(a) **Adressa**

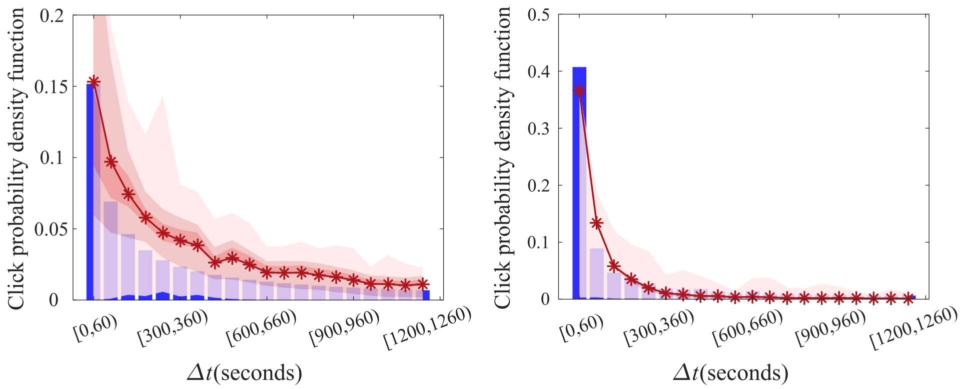
	models	*AUC*	*Imp*	*F1*	*Imp*	*MRR*	*Imp*	*NDCG@5*	*Imp*
Feature Engineering	*Pop*	-	-	-	-	47.16	69.64	52.05	60.02
	FM	96.46	1.22	45.00	41.78	66.19	20.86	70.87	17.53
	FM+	96.59	1.09	61.30	4.08	76.70	4.30	80.60	3.34
Survival Analysis	*COX*	85.46	14.25	54.05	18.04	67.85	17.91	69.42	19.98
	DeepHit	90.10	8.37	58.41	9.23	67.83	17.94	69.55	19.76
Content Based	*CNN*	97.07	0.59	46.58	36.97	66.54	20.23	71.29	16.83
	DSSM	96.84	0.83	46.51	37.17	66.92	19.55	71.71	16.15
Session Based	*DKN*	97.17	0.48	48.36	31.93	68.52	16.75	73.04	14.03
	LSTUR	96.94	0.72	46.15	38.24	67.03	19.35	71.97	15.73
Ours	***MFF***	**97.64**	-	**63.80**	-	**80.00**	-	**83.29**	-

Note that *SVD* and *GRU4REC* are not listed here, since they only involve news ID and user ID and are not capable of making predictions on this task (%).

distribution, we visualize the piecewise approximation for the click probability density function (i.e., $f(t)$ in Equation (15)).

We report the approximation results in Figure 7. Specifically, we gather all log instances in the same time period and calculate their proportion in all log instances (blue bars). In addition, we plot the predicted distribution for each period of time. For a better visualization, we divide each distribution into five intervals, where each interval is filled in the same color. We also calculate the expectation of corresponding distribution predicted by our *MFF*, which marks as red lines with asterisk).

In Figure 7, first, according to the blue bars, we can find the probability in the real datasets follows an exponential distribution. This phenomenon also demonstrates that a piece of news can hardly appeal to readers if it goes unclicked for a long time. According to the predicted results, the learned patterns (i.e., red lines with asterisk) are extremely similar to the real distributions, which proves the capacity of *MFF* for capturing the news timeliness effect. Second, comparing the results between two datasets, we observe the predicted scores for Adressa data are more concentrated so the distribution from 20th to 80th percentile is too narrow to be visible. We guess one possible reason is that most of the news articles in Adressa are highly time-sensitive ones, i.e., in Figure 6, "nyheter" news takes almost 50% proportion among all categories. As a result, news timeliness



(a) Toutiao (b) Adressa

Fig. 7. Illustration of piecewise approximation for the click probability density function w.r.t. the range of the time elapsed index Δt (seconds).

factor would be more dominated in this dataset. Therefore, the click probability tends to be similar in this experiment. On the contrary, in Toutiao, the number distribution of news categories is more balanced (recall the left middle chart of Figure 6), where many news articles belong to not time-sensitive ones, such as "food" and "care," which can last for longer time. As a consequence, the predicted probabilities on Toutiao dataset are more diffuse.

5.5.2 Parameter Sensitivity. We now discuss the sensitivities of some important hyper-parameters in our model including the loss coefficient γ in Equation (18), the time span ΔT in Section 4.3, the clicked news number of users in latest history K in Equation (6), and the embedding size D in news encoder in Section 4.1. Specifically, the loss coefficient γ balances the modeling learning on two losses of the binary loss or survival loss with value varying in the set {0, 0.01, 0.03, 0.05, 0.07, 0.09}. The time span ΔT in Section 4.3 controls the range of news timeliness interval, which helps the training of the survival part in our model. We make the experiments to adjust the time span with the value in {30, 60, 90, 120, 150}. The clicked news number of users K controls how many of the latest news clicked behaviors of users that our model can consider for modeling user-dependent short-term interest factor, where the value varies in the set {0, 2, 4, 6, 8, 10}. Last, the embedding size D greatly affects the representation ability of learned embeddings for exploring news content semantics, which ranges in the set {128, 256, 384, 512, 640, 768}. We report the experimental results in Figure 8. Note that the results in the left four charts are measured on Toutiao, while the rest are measured on Adressa. For better illustration, we only demonstrate the results on *NDCG*@5 metric (we find similar result patterns concerning other metrics). According to Figure 8, we have the following observations and conclusions:

- From the results in the top two charts, we conclude that adding the survival loss of news timeliness $\mathcal{L}_{sur}$ in Equation (17) would produce the performance effect of our model. Specifically, setting a non-zero γ in *MFF* can achieve higher *NDCG*@5 than it with $\gamma = 0$, and a too-large γ is less favorable, since it overwhelms the overall loss and misleads the direction of gradients. As the γ increases, the performance of our model first increases but decreases when γ 0.01, 0.05 in the corresponding datasets. Therefore, we set γ as 0.01 in Toutiao and 0.05 in Adressa for obtaining the best results.
- The impact of time span ΔT is concluded from the third and fourth charts. Specifically, the performance of our model reaches the peak when ΔT equals to 120, 30 in Toutiao,

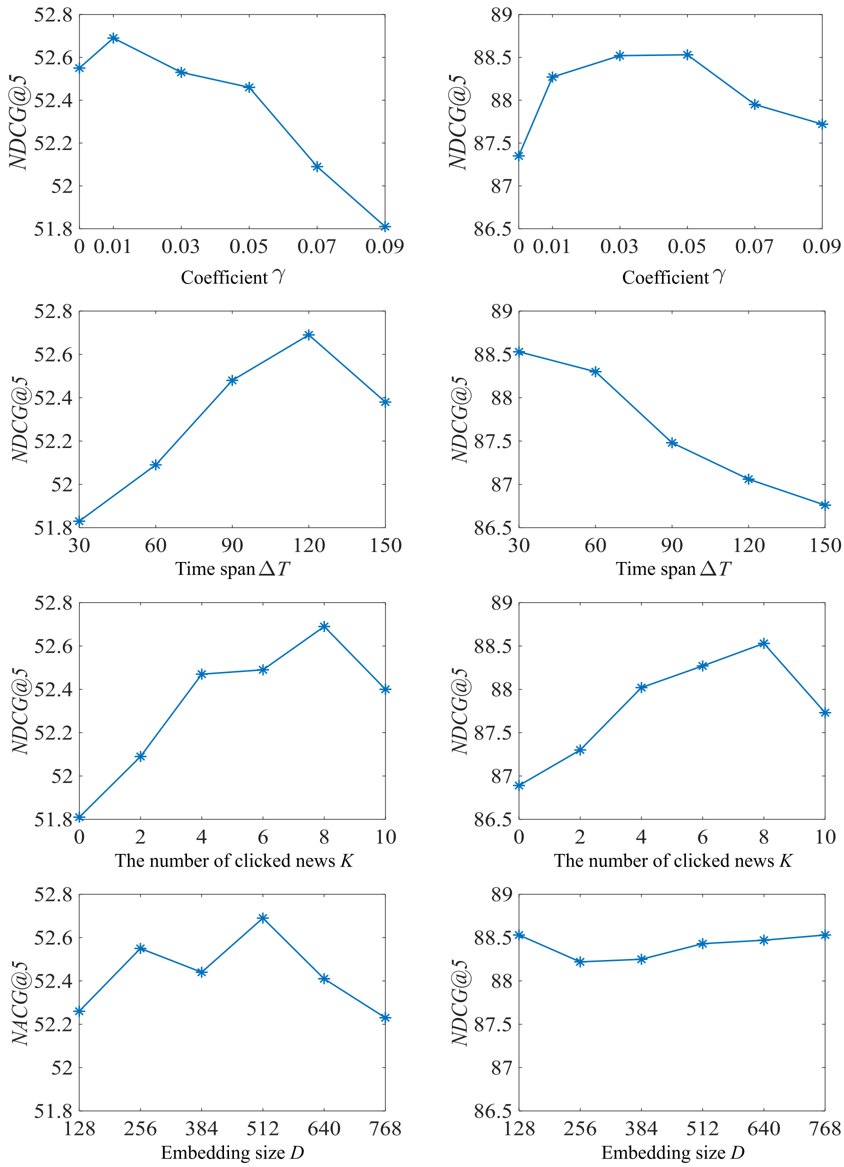


Fig. 8. Effects of different hyper-parameters on $NDCG@5$. Left: Toutiao; Right: Adressa.

Adressa, respectively. This suggests that it is necessary to describe the click hazard function (Equation (11)) over a longer time horizon when training the model on Toutiao. This conclusion is also consistent with the statistical results that the average time elapsed in Toutiao is larger than that in Adressa shown in Table 2 and Figure 6. Combining with the statistics in Figure 7 that the average time elapsed in Addressa is shorter than that in Toutiao, it is appropriate to set a smaller time span ΔT, because most click events happened in a short period of time. In a word, ΔT is a significant hyper-parameter that has a great impact on training the survival model, and the statistical results of time elapsed can help us choose a suitable value. Given the observation, we set ΔT=120, 30 in the corresponding datasets.

Table 6. Model Architecture Analysis

	Toutiao				Adressa			
	AUC	***F1***	***MRR***	***NDCG@5***	***AUC***	***F1***	***MRR***	***NDCG@5***
MFF-NP	63.45	58.65	66.74	50.34	98.71	69.19	83.70	87.03
MFF-AH	63.60	58.40	**68.59**	52.35	98.94	**70.87**	85.22	88.24
MFF-AP	64.40	58.65	67.95	52.11	98.89	70.72	84.60	87.79
MFF-RU	65.02	59.02	68.44	52.23	98.96	70.86	85.05	87.79
MFF-GM	64.89	58.96	68.45	52.64	98.92	70.62	85.21	88.26
MFF	**65.35**	**59.19**	68.38	**52.68**	**99.09**	70.36	**85.56**	**88.53**

MFF-NP removes the pre-training stage of the news encoder. *MFF-AH* replaces LSTM of modeling user short-term interest with the attention mechanism. *MFF-AP* replaces the tag-aware attention mechanism in news encoder with the common average pooling. *MFF-RU* means the user embedding is assigned with a random initialization. *MFF-GM* utilizes geometric mean to replace our click probability p_{click} in Equation (13) (%).

- We explore the influence of the number of the latest clicked news K in the fifth and sixth charts. Specifically, as K increases, the model performance increases at first and reaches the peak when K=8 on both datasets. Therefore, K is set as the value with 8 in our experiment, since it suggests that considering the recent 8 news clicking behaviors can model the short-term interest of users the best, which can help our model *MFF* to accurately predict the click rate of users on the candidate news, benefiting the recommendation performance. Besides, the results clearly show the rationality of distinguishing the user-dependent preference into long-term habit and short-term interest rather than coupling them together in the model.
- Last, we adjust the embedding size D to explore the model effectiveness, where the results are plotted in the bottom two charts. The fact is that *MFF* is not very sensitive to this hyper-parameter. The maximum and minimum values of $NDCG@5$ differ only by 0.47 and 0.31, on both datasets, respectively.

5.5.3 Model Architecture Analysis. At last, we would like to discuss how each sub-architecture of our *MFF* affects recommendation results. In Table 6, we adopt five *MFF* variants, each of which takes out or replaces one component from the complete method *MFF*. Specifically, *MFF-NP* refers to the *MFF* without pre-training stage so the parameters in BERT are randomly initialized (Figure 2). *MFF-AH* replaces the LSTM for modeling the dynamic user short-term interest effect in Equation (5) with an attention layer as Reference [64] does. *MFF-AP* replaces the tag-aware attention mechanism in news encoder in Equation (1) with an average pooling operation. *MFF-RU* removes the user's prior knowledge, which means the user embedding in Equation (3) is only assigned with a random initialized vector (i.e., $\boldsymbol{u}_l = \boldsymbol{u}$). The last *MFF-GM* modifies the click probability p_{click} in Equation (13) with geometric mean of p_1, p_2, p_3 (i.e., Equation (12)).

From the results in Table 6, we observe *MFF-NP* performs the worst, which means that the pre-training stage is critical, since it benefits a lot from a large volume of unlabeled news data to learn comprehensive news content semantics. Second, *MFF-AH* also decreases the performance compared our *MFF*. This indicates that modeling user-dependent short-term factor with a dynamic architecture is essential for news recommendation, which can better integrate user recent reading behaviors into one vector in our model. Then, *MFF* outperforms *MFF-AP*, demonstrating our model benefits from the proposed news encoder with the novel tag-aware attention mechanism. This also indicates that considering news category tag for learning news semantics could better help the model choose related words of news content, which benefits to establishing the relationship between news in semantic space. Similarly, *MFF* outperforms *MFF-RU*, which also demonstrates our tag-aware user embedding can bring improvement. Last but not least, compared with *MFF* and *MFF-GM*, we find that combing three factors under the assumption in *MFF* that a user would click the candidate news due to any one of them would produce improvement, where our assumption is valid for news recommendation.

6 CONCLUSION AND FUTURE WORK

In this article, we conducted a focused study on the personalized news recommendation. We proposed a novel **Multi-Factors Fusion (*MFF*)** model for news recommendation by integrating both user-dependent preference effect and user-independent timeliness effect together. Specifically, we decomposed the user-dependent preference into two factors, including the long-term habit and the short-term interest. Then, we succeeded in exploring the user-independent timeliness effect with the sophisticated survival analysis technique, where the short lifecycle of news could be well modeled in our model. Our experimental results demonstrated such component was one of the most significant one, especially for alleviating the cold-start problem. By combining three factors, our *MFF* could recommend not only news articles followed by user interests but also the breaking news that could satisfy users' demands for a wide range of information. We evaluated the performance of *MFF* using two real datasets, where the extensive experimental results fully validated the effectiveness of our proposed model. In the future, there are some potential study directions. First, we would like to model the user preference followed by groups for news recommendations. Second, we would further explore the news timeliness effect in more detail and design more sophisticated survival analysis models for tracking the news lifecycle. Third, we are also willing to perform more pre-training natural language processing modes for the news content semantics learning, which might benefit the performance further. We hope this work can lead to more studies.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long-and short-term user representations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 336–345.

[2] Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li, Courtney Page, Benjamin Goldstein, Lawrence Carin Duke, and Ricardo Henao. 2018. Adversarial time-to-event modeling. In *Proceedings of the International Conference on Machine Learning*. PMLR, 735–744.

[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 335–344.

[4] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 891–900.

[5] Tong Chen, Hongzhi Yin, Yujia Zheng, Zi Huang, Yang Wang, and Meng Wang. 2021. Learning elastic embeddings for customizing on-device recommenders. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 138–147.

[6] Tianqi Chen, Weinan Zhang, Qiuxia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. 2012. SVDFeature: A toolkit for feature-based collaborative filtering. *J. Mach. Learn. Res.w* 13, Dec. (2012), 3619–3622.

[7] Yifan Chen, Yang Wang, Xiang Zhao, Jie Zou, and Maarten De Rijke. 2020. Block-aware item similarity models for top-n recommendation. *ACM Trans. Inf. Syst.* 38, 4 (2020), 1–26.

[8] David R. Cox. 1992. Regression models and life-tables. In *Breakthroughs in Statistics*. Springer, 527–541.

[9] Asghar Darvishy, Hamidah Ibrahim, Fatimah Sidi, and Aida Mustapha. 2020. HYPNER: A hybrid approach for personalized news recommendation. *IEEE Access* 8 (2020), 46877–46894.

[10] Abhinandan S. Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: Scalable online collaborative filtering. In *Proceedings of the 16th International Conference on World Wide Web*. 271–280.

[11] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: Harnessing the real-time web for personalized news recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining.* 153–162.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT.* Association for Computational Linguistics, 4171–4186.

[13] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems.* 3492–3500.

[14] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Trans. Inf. Syst.* 39, 1 (2020), 1–42.

[15] Alex Graves, Abdel-Rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 6645–6649.

[16] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa dataset for news recommendation. In *Proceedings of the International Conference on Web Intelligence.* ACM, 1042–1048.

[17] Jinquan Hang, Zheng Dong, Hongke Zhao, Xin Song, Peng Wang, and Hengshu Zhu. 2022. Outside in: Market-aware heterogeneous graph neural network for employee turnover prediction. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining.* 353–362.

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Wweb.* 173–182.

[19] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR).*

[20] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the International Conference on Information Management.* ACM, 2333–2338.

[21] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. 2020. Learning or forgetting? A dynamic approach for tracking the knowledge proficiency of students. *ACM Trans. Inf. Syst.* 38, 2 (2020), 1–33.

[22] Zhenya Huang, Qi Liu, Chengxiang Zhai, Yu Yin, Enhong Chen, Weibo Gao, and Guoping Hu. 2019. Exploring multi-objective exercise recommendations in online education systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management.* 1261–1270.

[23] Qinglin Jia, Jingjie Li, Qi Zhang, Xiuqiang He, and Jieming Zhu. 2021. RMBERT: News recommendation via recurrent reasoning memory network over BERT. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1773–1777.

[24] Tingting Jiang, Qian Guo, Yaping Xu, Yang Zhao, and Shiting Fu. 2019. What prompts users to click on news headlines? A clickstream data analysis of the effects of news recency and popularity. In *Proceedings of the International Conference on Information.* Springer, 539–546.

[25] Binbin Jin, Hongke Zhao, Enhong Chen, Qi Liu, and Yong Ge. 2019. Estimating the days to success of campaigns in crowdfunding: A deep survival perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 4023–4030.

[26] Nirmal Jonnalagedda, Susan Gauch, Kevin Labille, and Sultan Alfarhood. 2016. Incorporating popularity in a personalized news recommender system. *PeerJ Comput. Sci.* 2 (2016), e63.

[27] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. 2018. News recommender systems–Survey and roads ahead. *Inf. Process. Manag.* 54, 6 (2018), 1203–1227.

[28] Faisal M. Khan and Valentina Bayer Zubek. 2008. Support vector regression for censored data (SVRc): A novel tool for survival analysis. In *Proceedings of the 8th IEEE International Conference on Data Mining.* IEEE, 863–868.

[29] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* 1746–1751.

[30] Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. 2014. Modeling dwell time to predict click-level satisfaction. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining.* 193–202.

[31] Thomas B. Ksiazek, Limor Peer, and Kevin Lessard. 2016. User engagement with online news: Conceptualizing interactivity and exploring the relationship between online news videos and user comments. *New Media Societ.* 18, 3 (2016), 502–520.

[32] Talia Lavie, Michal Sela, Ilit Oppenheim, Ohad Inbar, and Joachim Meyer. 2010. User attitudes towards news content personalization. *Int. J. Hum.-computer Stud.* 68, 8 (2010), 483–495.

[33] Changhee Lee, William R. Zame, Jinsung Yoon, and Mihaela van der Schaar. 2018. DeepHit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 2314–2321.

[34] Dongho Lee, Byungkook Oh, Seungmin Seo, and Kyong-Ho Lee. 2020. News recommendation with topic-enriched knowledge graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 695–704.

[35] Yu Lei, Hongbin Pei, Hanqi Yan, and Wenjie Li. 2020. Reinforcement learning based recommendation with graph convolutional Q-network. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1757–1760.

[36] Yu Lei, Zhitao Wang, Wenjie Li, and Hongbin Pei. 2019. Social attentive deep Q-network for recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1189–1192.

[37] Kristina Lerman and Tad Hogg. 2010. Using a model of social dynamics to predict popularity of news. In *Proceedings of the 19th International Conference on World Wide Web.* 621–630.

[38] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web.* 661–670.

[39] Yan Li, Vineeth Rakesh, and Chandan K. Reddy. 2016. Project success prediction in crowdfunding environments. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining.* ACM, 247–256.

[40] Yan Li, Jie Wang, Jieping Ye, and Chandan K. Reddy. 2016. A multi-task learning formulation for survival analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM, 1715–1724.

[41] Zhi Li, Zhao Hongke, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* ACM, 1734–1743.

[42] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2018. Towards better representation learning for personalized news recommendation: A multi-channel deep fusion approach. In *Proceedings of the International Joint Conference on Artificial Intelligence.* 3805–3811.

[43] Dongliang Liao, Jin Xu, Gongfu Li, Weijie Huang, Weiqing Liu, and Jing Li. 2019. Popularity prediction on online articles with deep fusion of temporal process and content features. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 200–207.

[44] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* 33, 1 (2019), 100–115.

[45] Yiqun Liu, Xiaohui Xie, Chao Wang, Jian-Yun Nie, Min Zhang, and Shaoping Ma. 2016. Time-aware click model. *ACM Trans. Inf. Syst.* 35, 3 (2016), 1–24.

[46] Hongyu Lu, Min Zhang, Weizhi Ma, Ce Wang, Feng Xia, Yiqun Liu, Leyu Lin, and Shaoping Ma. 2019. Effects of user negative experience in mobile news streaming. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 705–714.

[47] Pengtao Lv, Xiangwu Meng, and Yujie Zhang. 2019. BoRe: Adapting to reader consumption behavior instability for news recommendation. *ACM Trans. Inf. Syst.* 38, 1 (2019), 1–33.

[48] Tao Mei, Bo Yang, Xian-Sheng Hua, and Shipeng Li. 2011. Contextual video recommendation by multimodal relevance and user feedback. *ACM Trans. Inf. Syst.* 29, 2 (2011), 1–24.

[49] Grégoire Montavon, Geneviève Orr, and Klaus-Robert Müller. 2012. *Neural Networks: Tricks of the Trade.* Vol. 7700. Springer.

[50] Mohammad Naseri and Hamed Zamani. 2019. Analyzing and predicting news popularity in an instant messaging service. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1053–1056.

[51] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 1933–1942.

[52] Keunchan Park, Jisoo Lee, and Jaeho Choi. 2017. Deep neural networks for news recommendations. In *Proceedings of the ACM on Conference on Information and Knowledge Management.* 2255–2258.

[53] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Trans. Inf. Syst.* 37, 2 (2019), 1–24.

[54] Priyanka Rathord, Anurag Jain, and Chetan Agrawal. 2019. A comprehensive review on online news popularity prediction using machine learning approach. *Trees* 10, 20 (2019), 50.

[55] Shaina Raza and Chen Ding. 2020. A survey on news recommender system–Dealing with timeliness, dynamic user interest and content quality, and effects of recommendation on news readers. *arXiv e-prints* (2020), arXiv–2009.

[56] Steffen Rendle. 2012. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.* 3, 3 (2012), 57.

[57] Julio Rieis, Fabrício de Souza, Pedro Vaz de Melo, Raquel Prates, Haewoon Kwak, and Jisun An. 2015. Breaking the news: First impressions matter on online news. In *Proceedings of the International AAAI Conference on Web and Social Media*. 357–366.
[58] T. Y. S. S. Santosh, Avirup Saha, and Niloy Ganguly. 2020. MVL: Multi-view learning for news recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1873–1876.
[59] Chuan Shi, Xiaotian Han, Li Song, Xiao Wang, Senzhang Wang, Junping Du, and S. Yu Philip. 2019. Deep collaborative filtering with multi-aspect information in heterogeneous networks. *IEEE Trans. Knowl. Data Eng.* 33, 4 (2019), 1413–1425.
[60] Alexandru Tatar, Jérémie Leguay, Panayotis Antoniadis, Arnaud Limbourg, Marcelo Dias de Amorim, and Serge Fdida. 2011. Predicting the popularity of online articles based on user comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. 1–8.
[61] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint knowledge pruning and recurrent graph convolution for news recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 51–60.
[62] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. 2009. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. 1765–1768.
[63] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2020. Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation. *ACM Trans. Inf. Syst.* 39, 2 (2020), 1–33.
[64] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the World Wide Web Conference*. 1835–1844.
[65] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.
[66] Yang Wang. 2021. Survey on deep multi-modal data analytics: Collaboration, rivalry, and fusion. *ACM Trans. Multim. Comput., Commun. Applic.* 17, 1s (2021), 1–25.
[67] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2576–2584.
[68] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2023. Personalized news recommendation: Methods and challenges. *ACM Trans. Inf. Syst.* 41, 1 (2023), 1–50.
[69] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2020. User modeling with click preference and reading satisfaction for news recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 302–3029.
[70] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.
[71] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Trans. Knowl. Data Eng.* (2022), 1–1. DOI: 10.1109/TKDE.2022.3145690
[72] Guolei Yang, Ying Cai, and Chandan K. Reddy. 2018. Spatio-temporal check-in time prediction with recurrent neural network based survival analysis. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*.
[73] Yiping Yuan, Jing Zhang, Shaunak Chatterjee, Shipeng Yu, and Romer Rosales. 2019. A state transition model for mobile notifications via survival analysis. In *Proceedings of the ACM Conference on Web Search and Data Mining*. ACM, 123–131.
[74] Chengyuan Zhang, Yang Wang, Lei Zhu, Jiayu Song, and Hongzhi Yin. 2021. Multi-graph heterogeneous interaction fusion for social recommendation. *ACM Trans. Inf. Syst.* 40, 2 (2021), 1–26.
[75] Hui Zhang, Xu Chen, and Shuai Ma. 2019. Dynamic news recommendation with hierarchical attention network. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*. IEEE, 1456–1461.
[76] Qi Zhang, Qinglin Jia, Chuyuan Wang, Jingjie Li, Zhaowei Wang, and Xiuqiang He. 2021. AMM: Attentive multi-field matching for news recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1588–1592.
[77] Yanan Zhang, Guanfeng Liu, An Liu, Yifan Zhang, Zhixu Li, Xiangliang Zhang, and Qing Li. 2020. Personalized geographical influence modeling for POI recommendation. *IEEE Intell. Syst.* 35, 5 (2020), 18–27.
[78] Hongke Zhao, Binbin Jin, Qi Liu, Yong Ge, Enhong Chen, Xi Zhang, and Tong Xu. 2019. Voice of charity: Prospecting the donation recurrence & donor retention in crowdfunding. *IEEE Trans. Knowl. Data Eng.* 32, 8 (2019), 1652–1665.

[79] Hongke Zhao, Qi Liu, Hengshu Zhu, Yong Ge, Enhong Chen, Yan Zhu, and Junping Du. 2017. A sequential approach to market state modeling and analysis in online P2P lending. *IEEE Trans. Syst., Man, Cyber.: Syst.* 48, 1 (2017), 21–33.

[80] Shiwei Zhao, Runze Wu, Jianrong Tao, Manhu Qu, Minghao Zhao, Changjie Fan, and Hongke Zhao. 2023. perCLTV: A general system for personalized customer lifetime value prediction in online games. *ACM Trans. Inf. Syst.* 41, 1 (2023), 1–29.

[81] Wei Zhao, Benyou Wang, Min Yang, Jianbo Ye, Zhou Zhao, Xiaojun Chen, and Ying Shen. 2019. Leveraging long and short-term information in content-aware movie recommendation via adversarial training. *IEEE Trans. Cyber.* 50, 11 (2019), 4680–4693.

[82] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the World Wide Web Conference.* 167–176.

[83] Panpan Zheng, Shuhan Yuan, and Xintao Wu. 2019. Safe: A neural survival analysis model for fraud early detection. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 1278–1285.

[84] Qiqi Zheng, Guanfeng Liu, An Liu, Zhixu Li, Kai Zheng, Lei Zhao, and Xiaofang Zhou. 2021. Implicit relation-aware social recommendation with variational auto-encoder. *World Wide Web* 24, 5 (2021), 1395–1410.

[85] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 5941–5948.

[86] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1059–1068.

[87] Tengfei Zhou, Hui Qian, Zebang Shen, Chao Zhang, Chengwei Wang, Shichen Liu, and Wenwu Ou. 2018. JUMP: A joint predictor for user click and dwell time. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence.* AAAI Press. 3704–3710.

Received 13 December 2021; revised 5 June 2022; accepted 16 July 2022