applied sciences

MDPI

Article

ExoFIA: Deep Exogenous Assistance in the Prediction of the Influence of Fake News with Social Media Explainability

Pei-Xuan Li [1,†], Yu-Yun Huang [1,†], Chris Shei [2] and Hsun-Ping Hsieh [1,*]

[1] Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan; n26100618@gs.ncku.edu.tw (P.-X.L.); n26090546@gs.ncku.edu.tw (Y.-Y.H.)
[2] English Language, Tesol and Applied Linguistics, Swansea University, Swansea SA2 8PP, UK; c-c.shei@swansea.ac.uk
* Correspondence: hphsieh@mail.ncku.edu.tw
† These authors contributed equally to this work.

Abstract: The growth of social platforms has lowered the barrier of entry into the media sector, allowing for the spread of false information and putting democratic politics and social security at peril. Preliminary analysis shows that posts sharing real news and fake news are disseminated on social media. Moreover, posts pointing to fake news spread faster, so this paper aims to predict the impact of posts citing fake news on social platforms. In this study, we take into account that exogenous factors, in addition to endogenous factors, can potentially determine how influential a post is. For example, the occurrence of social events can generate public resonance and discussion, thereby increasing the impact of relevant posts. Given that Google Trends can obtain search trends that reflect social popularity, this work aims to use Google Trends as the source of our exogenous factors. We propose a deep learning model called the deep exogenous aid in fake news (ExoFIA) model, which combines multi-modal features and utilizes an attention mechanism to provide model interpretability and analyze the influencing factors. Applying the model to real-world data from Twitter demonstrates that our model outperforms existing diffusion models. Furthermore, further examination of the relevant aspects of true and fake news reveals that the two are influenced by distinct variables.

Keywords: fake news; popularity prediction; exogenous factors; multi-modal learning; model interpretability

Citation: Li, P.-X.; Huang, Y.-Y.; Shei, C.; Hsieh, H.-P. ExoFIA: Deep Exogenous Assistance in the Prediction of the Influence of Fake News with Social Media Explainability. *Appl. Sci.* **2023**, *13*, 6782. https://doi.org/10.3390/app13116782

Academic Editors: Andrea Prati, Muhammad Zubair Asghar, Asad Masood and Shakeel Ahmad

Received: 20 April 2023
Revised: 19 May 2023
Accepted: 30 May 2023
Published: 2 June 2023

1. Introduction

1.1. Introduction

Social media, e.g., Facebook, Twitter, and Sina Weibo, has changed the way we consume news because of its low cost, ease of access, and rapid dissemination [1]. Yet these factors have made social media a breeding ground for fake news. The term 'fake news' became popularized during Donald Trump's presidential election campaign in 2016 [2]. In recent years, in addition to the COVID-19 pandemic, we have also been battling an "infodemic" [3]. Fake news poses a direct threat to free speech, public awareness, and democratic societies. A significant amount of false information is disseminated on social media via hyperlinks in posts, as illustrated in Figure 1.

Fortunately, numerous works have paid attention to the detection of false news [4–6], aiding in the fight against online fake news. However, most works dealing with fake news focus on detection rather than gauging the influence of fake news on social media, which is also crucial for mitigating its impact on society.

Our work does not build a new fake news detection model. We aim to predict the future **popularity** of an online post linked to fake news, i.e., **the size of a cascade** (the term used to describe a posting's diffusion tree created by the original tweet that included the URL and all of its retweets). The cascade instances we defined are at the post level,

not the news level (one or multiple cascades with a singular origin). An example is shown in Figure 2. Fake news is spread via links in posts. The estimator should have the ability to gauge the influence of each post and sort all posts by their estimated influence, which executives can use as a measure of priority for reviewing posts. This can help organizations mitigate the spread of disinformation at an early stage. For example, online service providers such as Twitter or other forums can use the estimator to detect possible outbreaks of posts containing fake news, thereby safeguarding healthy conversations. Therefore, this work can be used in subsequent efforts to detect false news.

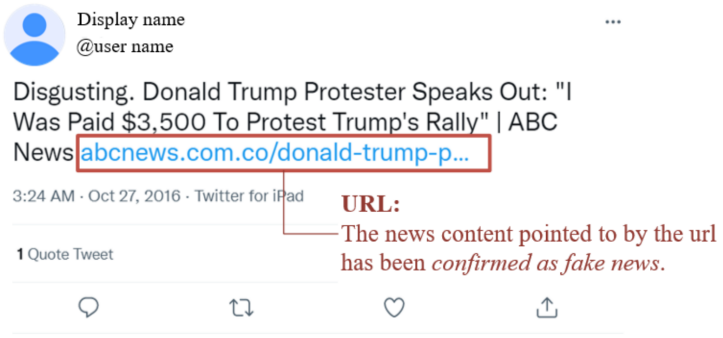


Figure 1. Modified examples (for anonymity) of tweets.

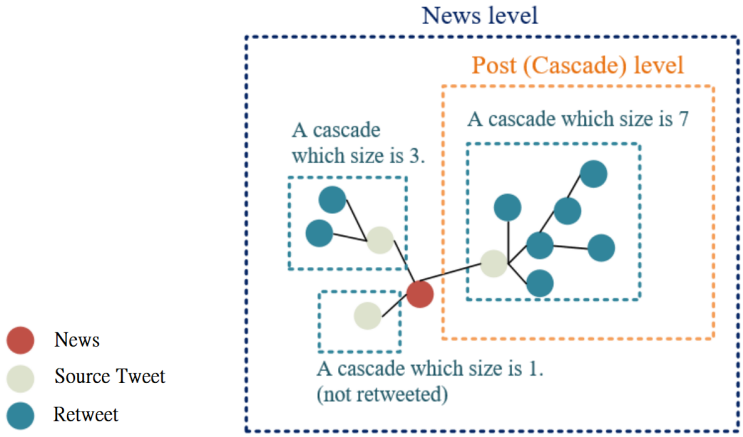


Figure 2. Illustration of fake news propagating on social media.

This work faces the following challenges. (1) Data collection and processing: the API limitations of Twitter make data acquisition time-consuming. Features used for forecasting are heterogeneous and are of multiple types, including social relationships, fake news content, news metadata, and user contexts. These types of features should be captured from different sources and require different retrieval processes and fusion methods to link the data together to reconstruct the whole picture. (2) Exogenous stimuli: the factors that cause large cascades are complex and are both endogenous (internal) and exogenous (external). Endogenous factors come from social platforms. However, exogenous factors are uncertain events outside the social platform that stimulate cascaded diffusion. To conclude, exogenous factors make modeling difficult. (3) Unbalanced distributions: the number of retweets of posts in real life exhibits a power-law distribution, with the bulk of posts not being retweeted and only a few posts receiving a large number of retweets. This characteristic makes the training process difficult.

Recently, several efforts have been devoted to popularity prediction on social media, which can be categorized into three main categories: generative approaches, feature-based approaches, and deep-learning-based approaches. Probabilistic statistical generative approaches, such as the Poisson process and Hawkes process, aim to model the

arrival/occurrence of event sequences or the participation time series, e.g., information retweeting [7–11].

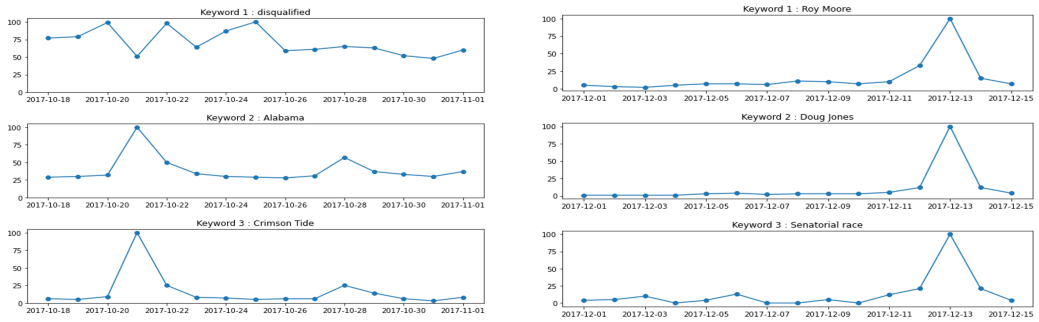
Because the intricate underlying mechanisms governing the success of a cascade are oversimplified, these studies cannot fully leverage the implicit information in cascade dynamics for effective prediction.

Feature-based approaches employ features from user characteristics [12,13], temporal information [14–17], content features [12,14,18], and the structures of propagation networks [14,15,19–21]. These methods heavily depend on domain knowledge and handcrafting, making the models hard to generalize. Deep-learning-based methods can automatically capture the dynamics of information dissemination [22–24] without requiring strong prior knowledge and feature engineering. Deep-learning-based models are powerful enough to extract representative features, yet they usually work as black-box models. The prediction results derived from deep-learning-based models for cascade popularity lack interpretability and are, as a result, of little value for decision making. Furthermore, most diffusion models of the above-mentioned work have never considered exogenous factors, such as burst events.

Endogenous and Exogenous Influences

Endogenous factors are the variables we can directly extract from the social platform of the target news item, such as when a news item is shared on Twitter. Endogenous factors, such as the number of likes or the news content of the post, can be further analyzed. On the other hand, exogenous factors can be extracted outside the post's social platform. For example, from the Google Trends service, we can know how many people are searching for news stating that U.S. President Joe Biden said he has decided to run for a second term.

As shown in Figure 3, both tweets come from the same person, but one has a smaller cascade size than the other. The search trend of Figure 3b peaked around the time before the source tweet was posted, which means that the news topic was quite popular at the time. In the case shown in Figure 3a, the first keyword, "disqualified", does not fluctuate much. In contrast, the second keyword, "Alabama", and the third keyword, "Crimson Tide", have a peak on the same day, but this was ten days before the tweet was posted, which shows that the discussion has declined. This result indicates that the search trend also plays a crucial role in predicting the influence of the tweets. Social media is inseparable from life, so information on social media is bound to be influenced by external platforms. Furthermore, the event will most likely occur outside of the social platform, implying that information will erupt on the social platform in the future. In conclusion, exogenous influences must also be considered.



(**a**) cascade sizes = 6 (**b**) cascade sizes = 134

Figure 3. The Google search trends of three keywords of two source tweets.

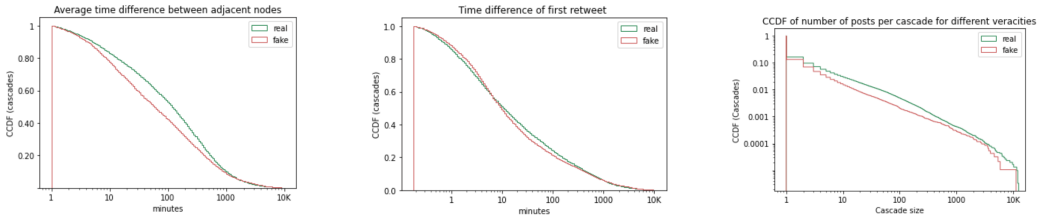
1.2. Present Work and Contributions

In this paper, we propose a multi-modal attention model called the ExoFIA model that predicts the influence of a post by forecasting the size of the cascade using both endogenous data (i.e., the post itself, the social network, and user information) and exogenous data (i.e., the trend of the news to which the post has linked). We adopt a graph convolution network

(GCN) [25] and gated recurrent unit (GRU) [26] to encode the dynamic propagation of the post, and we use Google Trends as exogenous sources and extract trends with 1D convolutional neural networks (1D-CNN) [27]. We further adopt attention mechanisms and extract the learned weights to enhance the interpretability of the model and explain why a post sharing fake news causes a large cascade. To evaluate our proposed model, we use large-scale real-world Twitter data.

Diffusion Patterns of Tweets Linked to Fake News vs. Posts Linked to Real News

First, we looked into the numerous ways that verified real news and fake news were being distributed on Twitter. Although many works have analyzed the dissemination differences between fake and real news [28–30], most of the studies performed their analysis at the news level (Figure 2). The cascade size of the tweet post related to real news is larger than fake news, as shown in Figure 4c. The average time difference between adjacent nodes (retweet nodes) indicates how fast a post is retweeted. Figure 4a shows that posts citing fake news are retweeted faster. On average, 40% of fake news cascades have a time difference exceeding 100 min, compared to 60% of real news cascades. In Figure 4b, it can be seen that the time difference between the source tweet and the first retweet is similar across different veracity sets.



(a) (b) (c)

Figure 4. CCDFs for different veracities. (**a**) Average time difference between adjacent nodes (retweet nodes). Cascades related to fake news have an average of 381.01, with a median of 64.44; cascades related to real news have an average of 390.99, with a median of 103.69. (**b**) Time difference between the source tweet and the first retweet. Cascades related to fake news have an average of 248.25, with a median of 9.58; cascades related to real news have an average of 229.30, with a median of 9.50. (**c**) Cascade size. Cascades related to fake news have an average of 2.92, with a median of 1; cascades related to real news have an average of 4.40, with a median of 1.

Due to the aforementioned differences, we also investigated the difference in variable importance between real and fake news posts by testing the framework on a dataset of 'real' news propagation.

To summarize, the main contributions of this paper are four-fold:

- This work studies a novel topic of predicting the influence of fake news on social media in an early stage, which is crucial for mitigating the impact of fake news.
- We propose a comprehensive framework named ExoFIA, which jointly models multi-modal features, including exogenous factors, such as public trends, and endogenous factors, such as the contents of posts, user characteristics, and the social network being posted on. ExoFIA is able to capture temporal and structural dynamics along the propagation of posts.
- Our proposed framework provides explainability with the aid of an attention mechanism for better understanding. We further examine the difference of feature importance between real and fake news.
- Extensive experiments are conducted on a real-world Twitter dataset, demonstrating the effectiveness of our proposed model. A comparison with existing prediction methods shows the superiority of ExoFIA.

2. Related Work

Many efforts have been made to anticipate the popularity of social media content based on information dissemination. This section briefly overviews popularity prediction and exogenous influences on social media.

2.1. Information Diffusion and Macroscopic Prediction

Previous research on information prediction can be divided into two categories based on the granularity of the tasks: the micro-level and the macro-level. Micro-level models focus on individual responses to information, whereas macro-level models predict how much attention information will receive in the future. Therefore, information prediction methods can be divided into the following categories.

2.1.1. Generative Process Approaches

We look at information retweeting as a series of events taking place within a continuous time period and model the impact of each event. The model observes every event and learns the parameters by maximizing the probability of events occurring during the observation time window [7–9,11,31]. Typically, the solutions fall into two categories. The first one is the Poisson process [8,10], which predicts the item's popularity by employing the reinforced Poisson process (RPP) and incorporating it into the Bayesian framework for external variable inference and parameter estimation. The second one is the Hawkes process adopted in [7,9,11], which constructs predictors that combine a self-exciting point process that regards the rate of events (e.g., retweets or citations) as a function of time and the previous history of events. The predictor leverages a feature-driven method to fit a memory kernel for estimating user influence, memory decay, and content virality [9].

These models are incapable of modeling important structural information that could aid in understanding the pathways of information diffusion.

2.1.2. Feature-Based Approaches

A variety of features are extracted from the raw data, which can be mainly divided into four types: temporal data [15–17,22], structural data [14,15,19–21], user information [12,13], and content features [12,14,18]. These features are used in a machine learning model to predict popularity. Temporal features are usually extracted using the peeking strategy, i.e., observing a small number of early participants and their active time. Temporal consideration in cascades has been identified as one of the most important factors in popularity prediction [14]. However, some studies claim that their advantages diminish over time [16]. Structural features extracted from graphs can be classified as [32]: (i) participants only, i.e., only cascade graphs are involved [20]; (ii) global graphs, i.e., both participants and non-participants are considered [21]; and (iii) r-reachable graphs, i.e., a compromise that extends the cascade graph within the scope of the global graph [14,15]. Furthermore, different platforms have unique diffusion mechanisms, which might result in dynamics that differ from well-studied social network scenarios [19]. Opinions of recent studies differ on the validity of content features. The study [14] found that content features became less important when more participants were observed, and ref. [12] found that their model did not improve its effectiveness by the addition of content features.

These hand-crafted features are difficult to build and rely on subject knowledge, and the conclusion of previous works may differ depending on the community platform.

2.1.3. Deep-Learning-Based Approaches

Inspired by the recent success of deep learning in many fields, cascade prediction has achieved significant performance gains using deep neural networks. DeepCas [23] is the first method based on graph representation learning to model and forecast the popularity of information cascades. It uses DeepWalk [33] concepts to sample cascaded graphs via random walks.

Unlike the DeepWalk concepts in DeepCas, Topo-LSTM [24] employs a directed acyclic graph(DAG)-structured recurrent neural network to model diffusion topologies. CasCN [22] leverages GCN and LSTM to extract both structural and temporal information from the cascade graph. By considering a cascade graph as a sequence of sub-cascade graphs, CasCN first learns each sub-cascade's local structure through graph convolutions and then adopts long short-term memory neural networks (LSTM) to model the evolving process of the cascade's structure.

Deep learning has good predictive power, but deep learning models lack model interpretability due to the "black box" nature of neural networks [34]. In addition, the computational cost of deep learning models is considerably greater than that of generative models and feature-based models.

2.2. Endogenous and Exogenous Influences

Exogenous (or external) factors are uncertain events that provide a stimulus for cascade diffusion. As shown in previous work [35], about one-third of tweets have been significantly affected and even manipulated by exogenous forces. Furthermore, burst events are more likely to first appear on newspapers and video-sharing sites and then spread to other microblogging platforms, such as Twitter and Weibo. This inspired later works to predict the popularity of a field via other information sources and dissemination platforms to model the external stimuli responsible for popularity. For example, the study [36] retrieved information from Twitter and YouTube to predict the "views" and "ratings" of movies in IMDB. With regard to social media discourse, the work [37] confirmed the superiority of models of chatter prediction that consider exogenous influences. In a recent study [38], Masud et al. became the first to propose a retweet prediction model to consider external influences. They used the news events as exogenous factors and modeled hate speech diffusion on Twitter. However, the use of "news events" as exogenous factors is relatively limited because news events are usually influenced by one-sided media or only cover topics that most people are interested in. In contrast, Google Trends is a more comprehensive collection of topics of interest to users across a region. Therefore, we believe Google Trends can be a strong representation of the popularity of people's engagement. A press release can only represent that an event occurred at that time, but its popularity is unknown. Modeling external stimuli can significantly enrich data diversity and improve model robustness.

3. Preliminaries

3.1. Dataset

3.1.1. Twitter Data

We conducted our experiments on the most well-known fake news data repository: FakeNewsNet [31]. The repository contains diverse features that can be categorized into three categories: news content, social context, and spatiotemporal information.

News content includes the meta-attributes of news (e.g., body text and title), collected from two reliable fact-checking platforms: GossipCop and PolitiFact. Each piece of news is reviewed by domain experts and annotated as real or fake news.

Social context includes the social engagements of news items. This includes, for instance, the posts that directly spread news pieces and the detailed information of the users, such as their Twitter profile description and the list of Twitter followers of each user.

As for spatiotemporal information, the spatial information indicates the location explicitly provided in user profiles or posts, and the temporal information indicates the timestamps of user engagements, which can be used to study how news pieces propagate on social media. The detailed statistics of the dataset are illustrated in Table 1.

Table 1. The statistics of PolitiFact dataset of different veracities.

	Fake	Real
# News	432	624
# News having related tweets	319	359
# Users	141,336	419,545
# Tweets	117,360	316,871
# Retweets	104,762	427,666

In the retweet data crawled from the official Twitter API, the retweet of a retweet points to the original tweet, as Figure 5 has shown. As a result, we cannot establish who discovered a tweet that a user later retweeted when reconstructing interactions. For the above reasons, we could not use the dataset directly to build the network, so we built networks by crawling the corresponding "follower network" from FakeNewsNet. The details are described in Section 3.2.

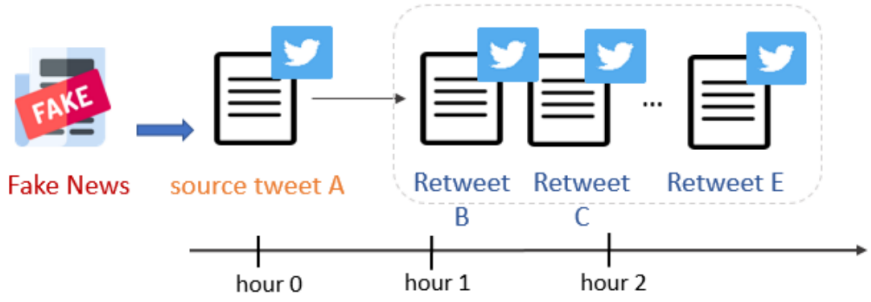
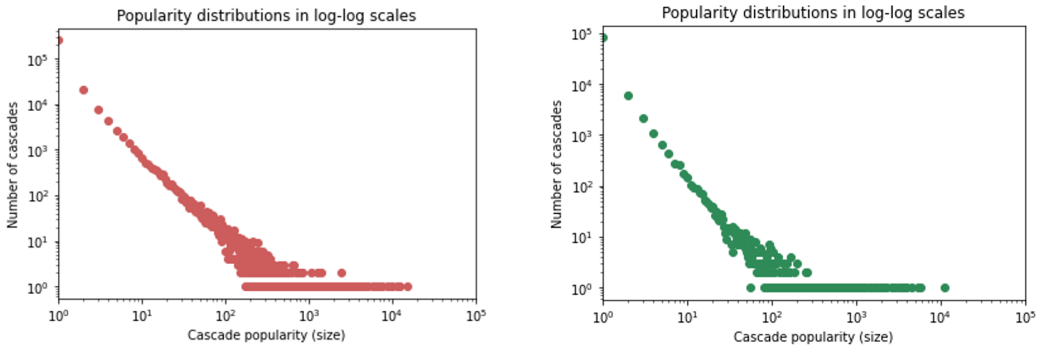


Figure 5. An example of retweets crawled from Twitter API.

As shown in Figure 6, the distribution of cascade sizes (popularity) is approximately a power-law distribution, implying that most source tweets do not spread at all, while a small fraction are reposted thousands of times. Source tweets sharing fake news have an average cascade size of 2.92, with a median of 1; tweets sharing real news had an average cascade size of 4.40, with a median of 1.



(**a**) Fake (**b**) Real

Figure 6. Distribution of cascade sizes.

3.1.2. Exogenous Source

As exogenous information, we queried Google Trends, which reflects real-world public concerns since it comprehensively collects the keywords searched by users across the region on Google's search engine. Google Trends is a valuable tool that displays trending search queries and the popularity of various keyword phrases over time. However, it only gives relative numbers, and there is no way to obtain absolute numbers. Users can select a certain

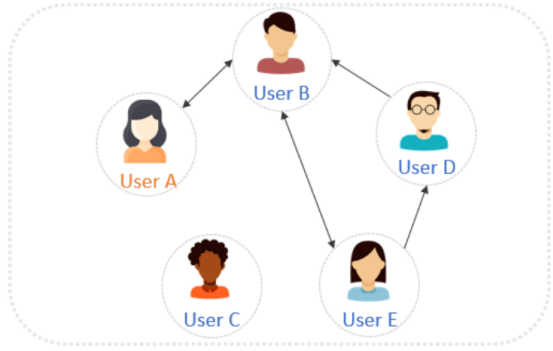
period and location, and the Google engine will show the trend according to the specific conditions selected. More information will be provided in Section 4.1.4.

3.2. Reconstructing Cascades

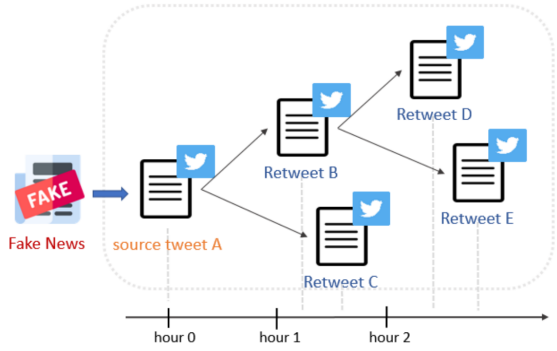
We can infer the source of a retweet and identify the possible user's friends who retweeted the tweet based on a retweet. We assume that if the user's retweet timestamp is later than the retweet timestamp of one of the user's friends (following), the user most likely saw the tweet from one of his/her friends and retweeted it.

When a tweet is retweeted by multiple friends, we consider the earliest retweet the source. As shown in Figure 7a, if user E has followed both user B and user D, and user B's retweet is earlier than user D, as shown in Figure 5, we believe that user E has received information from user B's retweet. Thus, we connect retweet B to E as a cascade in Figure 7b.

In the absence of an immediate retweet from a user's friend, we can assume that the retweet is from the original tweet rather than a retweet of another retweet. For instance, we can see that user C has no friends who have retweeted this tweet; thus, we link the source tweet to retweet C.



(**a**)



(**b**)

Figure 7. An example of a cascade constructed from (**a**) Follower network. A directed edge from user A to user B means that user A follows user B. (**b**) Post cascades with size = 5, depth = 3.

3.2.1. Diffusion Network and Social Network among Users

Figure 7b illustrates the process of a news post spreading in Twitter by retweeting the source tweet, which then shares the news link and can be further converted into a diffusion network and social network among users.

The diffusion path among users is shown in Figure 8, which is directly converted from the post's cascade. The social network among users can be converted from the post's cascade via the spreaders' follower lists. However, the direction is from the followed to the followers, which is in line with the information dissemination direction (information is spread from the followed to the followers).

A diffusion path is a tree structure with a single root node (i.e., the source tweet). In contrast, a social network built from the posts' cascade is not necessarily a tree structure.

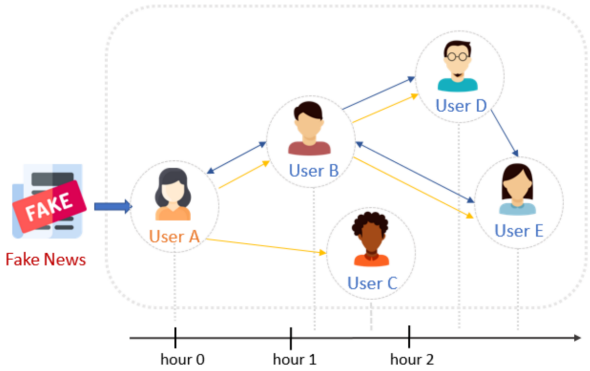


Figure 8. Diffusion network and social network among users. Yellow lines depict the diffusion paths among users. Blue lines depict the users' social networks. The direction of the arrow is the direction of information flow.

3.3. Problem Statement

Let us say we have S source tweets containing the fake news link, $\mathbb{S} = \{s_i\}(1 \leq i \leq S)$.

Fake News. Two key factors are used to define fake news: authenticity and intent [2]. First, fake news contains claims that can be verified by users as wrong information. Second, fake news is created with malicious intentions to mislead readers. Based on these two key features, the definition of fake news can be divided into two categories: narrow and broad. News needs to satisfy both key characteristics in order to meet the narrow definition of fake news. On the other hand, a broader definition of fake news focuses on the content's authenticity or intent. In this paper, we have adopted the broad definition to involve more data instances, such as inadvertently created false news content or biased news articles for political propaganda purposes.

Cascade. An information cascade can be viewed as a diffusion topology, which is depicted in the tree's structure. Each node in the tree represents one step of information propagation. For example, we refer to the post's diffusion tree generated by the source tweet s_i referencing the URL and all its retweets as the cascade.

Observed Cascade. For each source tweet s_i, the observed cascade is recorded as the set of early spreaders u within the observation time window T, i.e., $\boldsymbol{C}_{t=T}^{s_i} = \{u_1, u_2, \ldots, u_{n_T^{s_i}}\}$, in which $n_T^{s_i}$ is the number of users propagating the source tweet s_i within the observation time window T.

Popularity of Post. If the tweet is retweeted, then the retweet will become a child node of the source tweet. Therefore, we can define the size of a post cascade in social media as the number of users involved in the retweeting process, which is the post's total number (including source tweets and retweets). We quantify the popularity of a post using the cascade size, as defined in [8,14].

Future Popularity Prediction. Given an observation time T and a source tweet s with the news link, we have the observed cascade $\boldsymbol{C}_T^s$ and underlying network $\boldsymbol{G}_T^s = (\boldsymbol{V}_T^s, \boldsymbol{E}_T^s)$, in which $\boldsymbol{V}_T^s$ is the set of users associated with s within the observation time window T, and $\boldsymbol{E}_T^s \subset \boldsymbol{V}_T^s \times \boldsymbol{V}_T^s$ is the set of relationships between all users. For each user in the cascade, the profile and the historical tweets are retrieved as well. In this work, we aim to predict the natural logarithm of the final popularity $log(Y_{t_p=\infty}^s + 1)$ of the source tweet s . Figure 9 shows that when approaching seven days after publications, the cascade saturates. Thus, in this work, we consider seven days as a good approximation to the final popularity; i.e., we choose seven days as the prediction time, t_p.

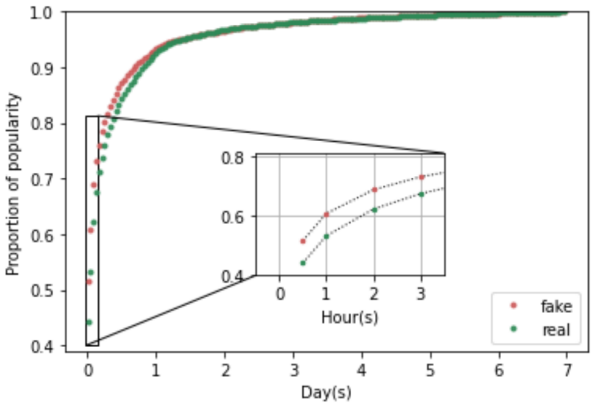


Figure 9. Percentage distribution.

4. Methodology

In this section, we provide the details of our proposed framework, ExoFIA. The architecture of ExoFIA is illustrated in Figure 10. Our method consists of two major components, uni-modal representation extraction and multi-modal attention fusion to learn the popularity. The features are categorized into endogenous and exogenous classes, which can be further divided into four categories: the network, the tweet post, the user's information, and the trend. The representation of each feature type is extracted through the uni-modal extraction module. Finally, the multi-modal fusion is in charge of performing our regression task.

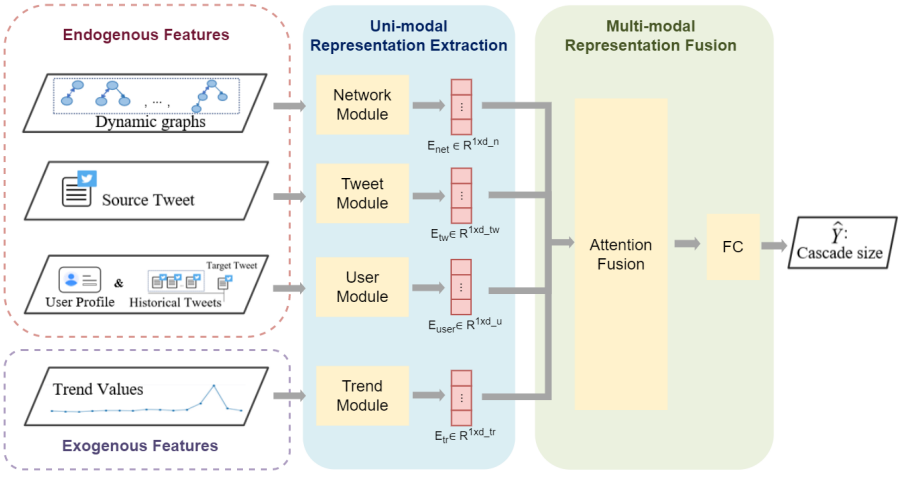


Figure 10. The architecture of our ExoFIA framework. We first enter information about a source tweet, including a dynamic graph constructed from the propagation of the tweets at the time of observation, the characteristics of the source tweets, information about the user who posted the source tweet, and Google Trends information. Then, the following embeddings are obtained by each representation extraction module: E_{net}, the dynamic graph embedding with size d_n; E_{tw}, the source tweet feature with size d_{tw}; E_{user}, a feature of the user who posted the source tweet, with size d_u; and E_{tr}, the Google Trends embedding with size d_{tr}. Finally, these four embeddings will be fed into the fusion module to predict the answers.

4.1. Uni-Modal Representation Extraction

4.1.1. Network Module

We intended to capitalize on the process of information propagation among users during the growth of the post's cascade over time. The architecture of Network Module is illustrated in Figure 11. And the input features of this module is shown in Table 2.

Dynamic graphs. Given the observation time $[0, T)$, we divided the observation time into n intervals, so the length of an interval was denoted as $\lfloor \frac{T}{n} \rfloor$. Then, we transformed the original graph into a sequence of sub-graphs, i.e., $\mathbb{G}_T^s = \{\boldsymbol{g}^s_{\lfloor \frac{T}{n} \rfloor * 1}, \boldsymbol{g}^s_{\lfloor \frac{T}{n} \rfloor * 2}, \cdots, \boldsymbol{g}^s_{\lfloor \frac{T}{n} \rfloor * k}\}$, $k \in [1, n]$, in which $\boldsymbol{g}^s_{\lfloor \frac{T}{n} \rfloor * k}$ is the snapshot of the original graph at the time-instant $\lfloor \frac{T}{n} \rfloor * k$. The dynamic graphs afford us a glimpse of the speed of spread and the scale of the early cascade.

We leveraged GCN [25] to exploit the local structure of each sub-graph and then used GRU [26] to capture the evolving process of the graph structure. We adopted a heterogeneous graph convolutional network to jointly learn the structural characteristics of the diffusion graph and social graph mentioned in Section 3.2.1. Diffusion graphs are important for understanding representations of structural and temporal patterns, while social networks provide a structure for understanding the relationships between users. Social graphs are the basic means of tweets dissemination, which can reveal community information.

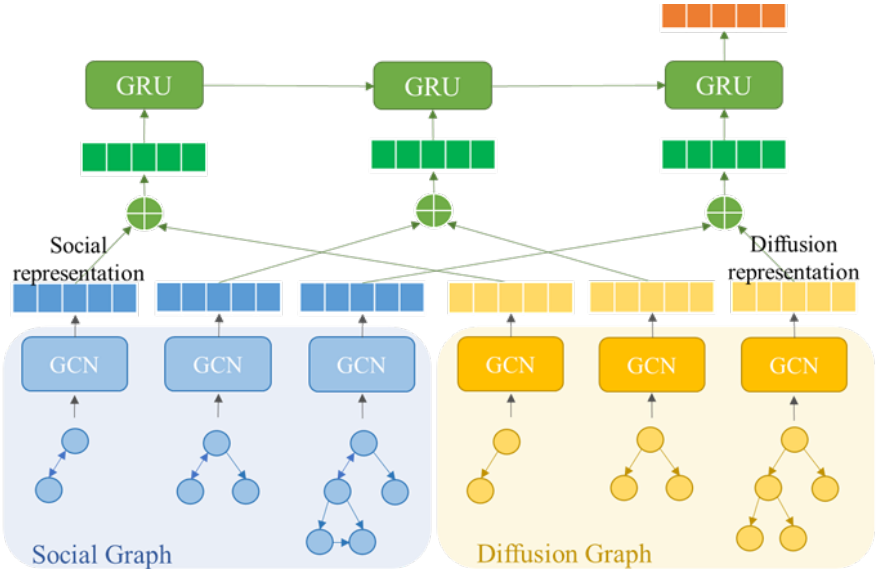


Figure 11. The architecture of the network component.

GCN is a convolutional neural network that can work on a non-Euclidean structure and take advantage of a graph's structural information. Given an adjacency matrix $\boldsymbol{A}$ and node feature matrix $\boldsymbol{X}$, the convolution layers capture spatial features between nodes by their first-order neighborhoods. The GCN model can be built by stacking multiple convolution layers, which can be expressed as

$$\boldsymbol{H}^{(\ell+1)} = \sigma(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}), \tag{1}$$

in which $\boldsymbol{H}^{(\ell)}$ is the output of the ℓ layer, $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ denotes the matrix with added self-connections, $\hat{\boldsymbol{D}}_{ii} = \sum_{j=0} \hat{\boldsymbol{A}}_{ij}$ is the diagonal degree matrix, $\boldsymbol{W}^{(\ell)}$ are learnable parameters of the ℓ layer, and $\sigma(\cdot)$ represents the ReLU function.

We stacked one convolution layer for each graph. $\boldsymbol{H}^{(0)}$ was set as a node feature matrix $\boldsymbol{X}$. $H^{(0)}$ was fed into the GCN layer to compute the output matrix of the next layer. Seven user traits are specified in Section 4.1.3, as well as a time gap between the retweet node and the source tweet node. The new node feature matrix was denoted by $\boldsymbol{H}^{(\ell+1)} \in \mathbb{R}^{n_t^s \times d}$, in which n_t^s is the node number of a graph and d is the dimensionality of the node embedding.

$$\boldsymbol{H}^{(\ell+1)}_{social} = \sigma(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}_{social}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}_{social}), \tag{2}$$

$$\boldsymbol{H}^{(\ell+1)}_{diff} = \sigma(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}_{diff}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(\ell)}\boldsymbol{W}^{(\ell)}_{diff}). \tag{3}$$

After convolution, average pooling was applied to generate the graph representation $\boldsymbol{o} \in \mathbb{R}^d$. To fuse two types of graph representations, we used element-wise addition for aggregation, as shown in Equation (4). Thus far, each time interval's network structure has been properly encoded.

$$\boldsymbol{o}_{network} = \boldsymbol{o}_{social} \oplus \boldsymbol{o}_{diff}. \tag{4}$$

GRU was then adopted to capture temporal dependence. The GRU hidden vector output was at step t, h_t (Equation (6)). For the input sequence $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_n\}$ is given by

$$\boldsymbol{u}_t = \sigma(\boldsymbol{W}_u \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{o}_t]), \tag{5a}$$
$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_r \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{o}_t]), \tag{5b}$$
$$\tilde{\boldsymbol{h}}_t = \sigma(\boldsymbol{W}_{\tilde{h}}[\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}, \boldsymbol{o}_t]), \tag{5c}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{u}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{u}_t \odot \tilde{\boldsymbol{h}}_t, \tag{6}$$

in which $\odot$ denotes the element-wise product; u_t functions as an update gate (Equation (5b)) and determines which portions of the previous hidden state must be transmitted to the future; $\boldsymbol{r}_t$ indicates the reset gate (Equation (5c)), which decides what parts of the previous hidden state to consider or ignore at the current step; $\tilde{\boldsymbol{h}}_t$ in Equation (5c) computes new hidden state content using the parts of the previous hidden state, as dictated by $\boldsymbol{r}_t$; and $\boldsymbol{W}_u$, $\boldsymbol{W}_r$, and $\boldsymbol{W}_{\tilde{h}}$ are learnable weights.

To further enhance the network representation, the last hidden state of GRU was further concatenated with structural features and temporal features as the auxiliary network representation.

Table 2. Network features.

	Feature	Description
Structural features	*max_deg*	The maximum out-degree of the social graph.
	avg_deg	The average out-degree of the social graph.
	min_deg	The minimum out-degree of the social graph.
	edge_number	The number of edges of the social graph.
	node_number	The number of nodes of the social graph.
	diff_path	The depth of the diffusion graph.
Temporal features	*min_timediff*	Time elapsed between the source tweet and the first retweet.
	max_timediff	Time elapsed between the source tweet and the last retweet.
	avg_timediff	The average time elapsed between the adjacent retweets.
	time_diff_at_max_deg	Time elapsed between the source tweet and the retweets with the maximum out-degree.

4.1.2. Tweet Module

It is worth using information from the post itself to predict the size of a post cascade. We fetched the tweet object through the Twitter API, including the tweet content and meta-information of the tweet. The metadata of the post have also been confirmed to affect the spread of the post. However, the effectiveness of the content is a point of contention. We did not use content features in the final tweet component for two reasons. First, we used DistilBERT [34] as an encoder to obtain semantic vectors, and the performance did not improve by including content features. The results of this are similar to the work [14,39]. Second, the amount of time spent training DistilBERT was significant.

We further classified features into three categories: statistical, temporal, and sentiment, as shown in Table 3. For the sentiment features, we use dValence Aware Dictionary and sEntiment Reasoner (VADER) [40], a dictionary and rule-based sentiment analysis tool that specializes in sentiments expressed in social media. A sentence's sentiment score was the total of the sentiment scores of each sentiment-bearing word. The average sentiment score

of every sentence in the source tweet was used as the source tweet's *sentiment score*, and the value ranged between −1 and 1, from most negative to most positive.

Then, we concatenated statistical, temporal, and sentiment features as the source tweet representation, as shown in Table 3.

Table 3. Tweet features.

	Feature	Description
Statistical features	*user_count*	The number of mentioned users in the source tweet.
	tag_count	The number of hashtags in the source tweet.
	symbol_count	The number of symbols in the source tweet.
	url_count	The number of URLs in the source tweet.
	sentence_count	Sentence count of the source tweet.
Temporal features	*hour*	The hour when the source tweet was created.
	weekday	Day of the week when the source tweet was created.
	is_holiday	Boolean. When true, the created time of the source tweet is on a holiday.
Statistical features	*sentiment_score*	The source tweet's sentiment score. The value is between −1 and 1, from the most negative to the most positive.
	pos_count	The number of positive sentences in the source tweet.
	neg_count	The number of negative sentences in the source tweet.
	sentiment_ratio	The ratio of positive sentences to negative sentences.

4.1.3. User Module

User profile. User behavior plays an important role in the propagation of the cascade, and the most obvious factor is the user's follower count. The greater the number of followers, the greater the user's influence, and the more widely the post spreads [13]. Therefore, we also fed users' information into the model, such as how active they are on social media, including their number of followers, their number of friends, whether their account is officially verified, and so on. This information can be seen in the user's attributes via Twitter API, as shown in Table 4 below.

Table 4. User profile features.

	Feature	Description
Statistical features	*geo_enabled*	Boolean. If true, then the user agrees to have location data posted in his tweets.
	verified	Boolean. If true, it indicates that the user has a verified account.
	followers_count	The number of followers this account currently has.
	friends_count	The number of users this account is following.
	listed_count	The number of public lists that this user is a member of.
	statuses_count	The number of tweets (including retweets) that a user has posted.
	favourites_count	The number of tweets this user has marked as favorite in the account's lifetime.

User timeline. The user's posting habits and followers' feedback to the user can also be observed from the user's historical posts. We used the Twitter API to collect the ten tweets preceding the target tweet and observed the user's posting habits from the historical timeline. We extracted two categories of features that shown in Table 5: statistical, such as *avg_time_diff*, which can provide a glimpse into the user's posting habits, and $avg\_favorite\_count$, which can be used to observe the response of followers. The sentiment feature was the average sentiment score of the timeline acquired through VADER, allowing one to witness whether the content leans towards extremes.

Table 5. User timeline features.

	Feature	Description
Statistical features	*avg_rt_count*	The average number of retweets of historical posts.
	avg_favorite_count	The average number of times the historical posts have been marked as favorites.
	avg_time_diff	The average time between posts of historical posts.
	avg_sentence_count	The average sentence count of historical posts.
	sensitive_tweet_ratio	The ratio of historical posts marked as sensitive.
Sentiment features	*avg_sentiment_score*	The average value of sentiment_score from historical posts.
	avg_pos_count	The average value of pos_count from historical posts.
	avg_neg_count	The average value of neg_count from historical posts.
	avg_sentiment_ratio	The average value of sentiment_ratio from historical posts.

In this module, we simply combined all features as a single feature representation, i.e., E_{net} in Figure 10. This representation was fed into the multi-modal attention fusion module together with the other representations obtained from the other modules.

4.1.4. Trend Module

The dissemination of information on social media is affected by internal signals, such as user engagement with posts, as well as endogenous signals, such as sudden occurrences outside the platform. Our aim is to measure public reactions to the news contained in a tweet at that very moment and use it as an exogenous signal. Utilizing Google Trends, we were able to accomplish our objectives. The architecture of Trend Module is illustrated in Figure 12.

Exogenous features. Using TF-IDF, we pulled out three relevant keywords from the title of the news. Not all of these were accurate, however, so we manually sorted through them to replace any unreasonable ones. To obtain their corresponding trending values, we used the Google Trends API to take data from 15 days prior to the date of the source tweet up until the day before. Finally, the geographical location was set to the United States area.

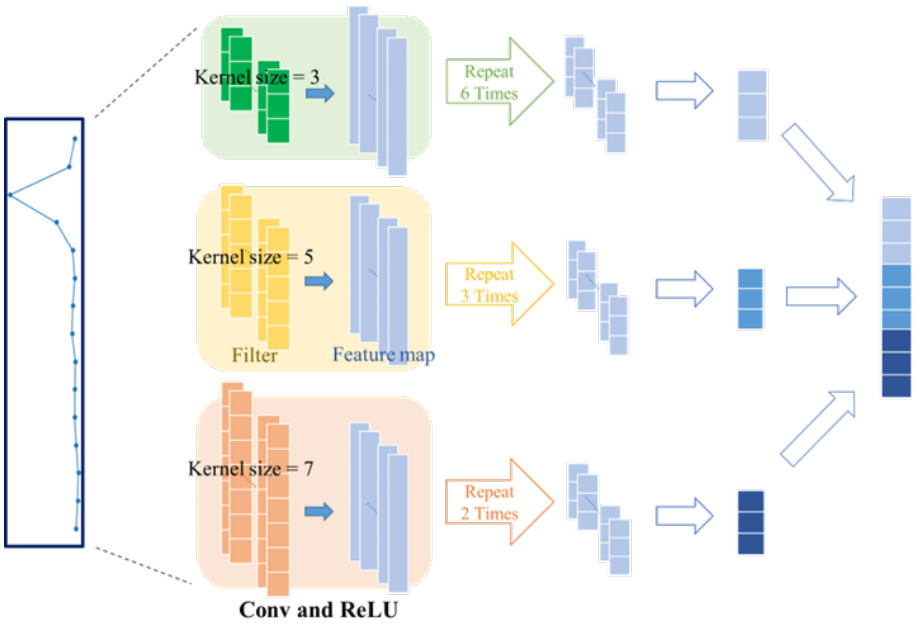


Figure 12. The architecture of the trend component.

One-dimensional convolutional neural network (1D-CNN) models can be used to encode trending values. 1D filters were employed in each convolutional layer to capture the local features of the input data, followed by a ReLU function for non-linear transformation of the data.

The mathematical representation of the convolutional operation is given by

$$\boldsymbol{c}_i = \sigma(\boldsymbol{x}_{i:i+h-1} * \boldsymbol{w}_i + \boldsymbol{b}), \boldsymbol{c}_i \in \mathbb{R}, \tag{7}$$

in which $\boldsymbol{w}_i \in \mathbb{R}^{h \times 1}$ denotes a filter that is applied to a window of h days' trending values to produce a new feature, i corresponds to the trending value index, and b is the bias. The convolution operator is represented by $*$, and $\sigma(\cdot)$ corresponds to the activation function. The feature $\boldsymbol{c}_i$ is generated from windows of trending values $\boldsymbol{x}_{i:i+h-1}$. This filter was applied to each possible window of trending values to produce a feature map, as shown in Equation (8).

$$\mathbf{c} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_{n-h+1}]. \tag{8}$$

The model employed multiple filters (with a diverse range of window sizes) to extract multiple features. To capture the public's attention at different intervals, we set filter windows of 3, 5, and 7. We set up distinct numbers of layers for each branch of the window sizes so that the output size of the final layer was consistent; i.e., it had the same proportion of features regardless of time scale. Then, we applied channel-wise mean and concatenated the features from different time scales.

The trend representation was obtained and further concatenated with statistical features and trend features, as shown in Table 6 below.

Table 6. Trend features.

	Feature	**Description**
Statistical features	*img_count*	Number of images in the news article.
Trend features	*RSI_Mean*	*Relative strength index*: A momentum indicator that measures the magnitude of recent trend changes. The average is the three keywords' RSI values. There are three variables for different time-scales.
	RSI_Max	We use the largest RSI value of the three keywords. There are three variables for different time-scales.
	MA_Mean	*MA*: moving average is a statistic that captures the average change in a data series over time. The average is the three keywords' *MA* values. There are three variables for different time-scales.
	MA_Max	We use the largest *MA* value of the three keywords. There are three variables for different time-scales.

4.2. Multi-Modal Attention Fusion

In order to incorporate the multi-modal information, we introduce an attention mechanism to focus on specific features, since each feature contributes differently to the model. The architecture is show in Figure 13. This is followed by multiple layers of the perceptron. Taking the concatenation of the representation encoded by different modules denoted as $\boldsymbol{Z} \in \mathbb{R}^{1 \times D}$ as input, it outputs one final unit y, i.e., a cascade size level ($y = log(Y^s_{t_p=\infty} + 1)$). D represents the sum of the dimensions of the four output embeddings of the uni-modal representation extraction, i.e., $D = d_n + d_{tw} + d_u + d_{tr}$.

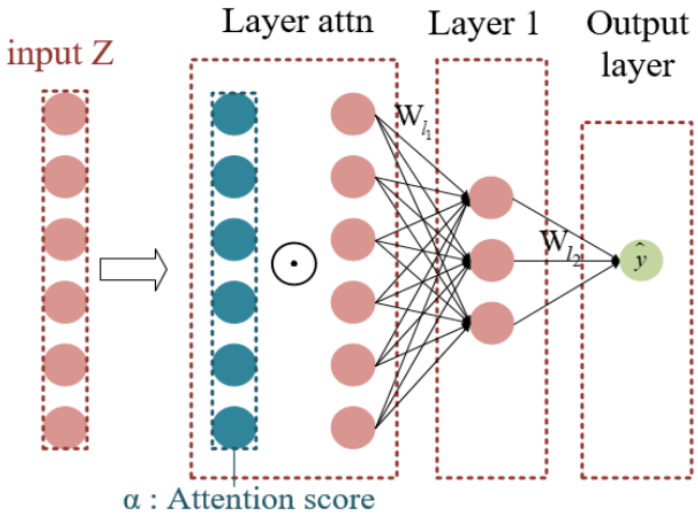


Figure 13. Thearchitecture of attention fusion.

The neural network architecture that implements the attention mechanism can be formulated as

$$y = \sigma(\boldsymbol{W}_{\ell_2}(\sigma(\boldsymbol{W}_{\ell_1}(\boldsymbol{Z} \odot \alpha) + \boldsymbol{b}_{\ell_1})) + \boldsymbol{b}_{\ell_2}), \tag{9}$$

in which σ corresponds to the activation function ReLU, $\odot$ refer to the element-wise product, and $\boldsymbol{W}_{\ell_1}, \boldsymbol{W}_{\ell_2}, \boldsymbol{b}_{\ell_1}, \boldsymbol{b}_{\ell_2}$ are learnable weights.

$$\alpha = softmax(W_{l_{attn}}\boldsymbol{Z} + b_{l_{attn}}), \tag{10}$$

where $\boldsymbol{W}_{l_{attn}} \in \mathbb{R}^{D\times D}$; $\boldsymbol{b}_{l_{attn}} \in \mathbb{R}^{1\times D}$; $\alpha \in \mathbb{R}^{1\times D}$ denotes the attention score vector, with the value of each dimension in the vector $\alpha_i \in \mathbb{R}, 1 < i < D$ representing the importance of the feature corresponding to this dimension. To obtain our target, the cascade size level $\hat{y}$, we applied a multilayer perceptron (MLP) with one hidden layer on $\boldsymbol{Z} \odot \alpha \in \mathbb{R}^{1\times D}$ to obtain the target.

Objection function. The objective function to minimize is defined as follows:

$$L(\hat{\boldsymbol{Y}}, \boldsymbol{Y}) = \frac{1}{S}\sum_{i}^{S}(log(\hat{y}_i + 1) - log(y_i + 1))^2, \tag{11}$$

in which S is the total number of cascades, $\hat{\boldsymbol{Y}}$ is the predicted value, and $\boldsymbol{Y}$ is the ground-truth label. The cascade size, or popularity of a tweet, is measured on a natural logarithmic scale. This is because tweet popularity can vary greatly, ranging from just one retweet to thousands. Most tweets have very few retweets, resulting in a heavy and long-tailed distribution, as illustrated in Figure 6. When using logarithmic scaling, the model is not impacted by extreme values, which ensures stability for the loss function.

5. Experiments

We first present the experimental setup in Section 5.1, including the dataset partition and evaluation metrics. Sections 5.2–5.4 contain various experiments that evaluate the effectiveness of our proposed framework, ExoFIA. Specifically, we aim to answer the following research questions:

- RQ 1. Can our proposed framework, ExoFIA, achieve robust effectiveness in fake news popularity prediction by taking into account multi-modal contents, such as networks, user characteristics, tweets, and trends?
- RQ 2. How critical are exogenous features to improving ExoFIA's prediction performance? Does the use of both diffusion and social networks improve performance?

- RQ 3. Can ExoFIA exploit the features extracted from endogenous and exogenous sources to explain why a tweet sharing fake news leads to a massive information cascade? Is there a discrepancy between real news and fake news?

To answer RQ 1 and RQ 2, we compared the performance of our model, ExoFIA, to several baselines in Section 5.2.1, followed by a few variants of ExoFIA in Section 5.2.2. In addition, we measured the performance of these models under different observation time windows, as detailed in Section 5.3. To address RQ 3, we explored the differences in feature importance between real and fake news propagation by testing our framework on a dataset of posts sharing 'real' news, as seen in Section 5.4.

5.1. Experimental Setup

We sampled 10% of the entire data as a testing set and used the remaining 90% to tune parameters by stratified 5-fold cross-validation on a parameter grid. This 90% was further split into a 9:1 ratio for training and validation, respectively. The performance is reported on the testing set.

5.1.1. Baselines

We selected methodologies from the following categories to be competitors: (1) statistics-based approaches, (2) feature-based approaches, and (3) deep-learning-based approaches.

Statistics-based approahches:

- Node_num_at_T: We counted the number of posts observed within the observation time and used the values as the predictive cascade size.

Feature-based approaches:

- Future-linear: We fed the hand-crafted features outlined in Sections 4.1.1–4.1.4 into a linear regression model.
- Future-deep: We fed the hand-crafted features into a MLP model consisting of three layers of nodes.

Deep-learning-based approaches:

- Topo-LSTM [24]: A DAG (directed acyclic graph)-structured recurrent neural network was used to model diffusion topologies. The original application of Topo-LSTM is tailored for tasks related to node activation, so we replaced the logistic classifier with a regressor to predict the cascade size. Additionally, Topo-LSTM only depends on the order of nodes in each cascade. To ensure fairness in comparison, richer node features were incorporated by taking into account user information.
- CasCN [22]: CasCN leverages GCN and LSTM to extract both structural and temporal information from the cascade graph. By considering the cascade graph as a sequence of sub-cascade graphs, it studies the local structure of each sub-cascade through graph convolutions. After that, it applies LSTM to model the development of the cascade structure.

5.1.2. Variants of ExoFIA

In addition to our comparison with existing baselines, we also derived a few variants of ExoFIA:

- ExoFIA-trend: In ExoFIA-trend, the exogenous signal of public search interest in news is not taken into consideration.
- ExoFIA-diff: In ExoFIA-diff, the diffusion network among users is disregarded.
- ExoFIA-social: In ExoFIA-social, the social network among users is disregarded.

5.1.3. Parameter Setting

In the following experiments, all neural network were optimized by the Adam optimizer [41] with a learning rate of 1^{-3}. The batch size of the training set was set to 4096, and early stopping was employed when the validation loss did not decline for three consecutive

epochs. Regarding the tuning of the hyper-parameters, we used the 5-fold validation method with the brute force method to find the best hyper-parameters. In other words, we set 5 different learning rates and 5 different batch sizes using 5 different training–validation sets to find the most suitable values.

The observation time T for all models was set to three hours and divided into three intervals, i.e., $n = 3$. For Topo-LSTM, CasCN, and ExoFIA, the node-embedding dimension was 8. All other hyper-parameters for each model were set to their default values. For ExoFIA, the dimensionality of node embedding, d_n, is 8 as shown in Figure 10, and the hidden vector dimension of GRU is 8. The size of the source tweet feature d_{tw} is 12. The size of the user features d_u is 16. For the 1D-CNN in the trend module, we used three different filters with sizes 3, 5, and 7, each with two feature maps, and output three embeddings of dimension 3, which we then concatenated. Thus, the dimension of the Google Trends embedding, d_{tr}, is 9, as seen in Figure 10.

5.1.4. Evaluation Metric

For the evaluation metric, we used the mean square error (MSE) as defined in Equation (12) [22,24,42].

$$MSE = \frac{1}{S}\sum_{i}^{S}(\hat{y}_i - y_i)^2, \tag{12}$$

in which S is the total number of cascades, $\hat{y}_i$ is the predicted value, and y_i is the ground-truth label.

In the real-life setting, our focus is more on identifying a relatively large cascade rather than the exact value of its size. That is, we should be more concerned about whether the model predicts a higher ranking of relative values when the source tweets have larger cascades. Therefore, in addition to regression metrics, we also use two metrics commonly employed for ranking: normalized discounted cumulative gain on the top k ($NDCG@k$) and hit ratio on the top k ($HR@k$). $HR@k$ is defined as follows:

$$HR@k = \frac{Number\ of\ Hits@k}{Total\ number@k}, \tag{13}$$

in which 'hits' means that if a tweet is predicted to be in the top k, it is also in the top k of the ground-truth label.

$NDCG@k$ is defined as follows:

$$NDCG@k = \frac{DCG@k}{IDCG@k}, \tag{14}$$

where $DCG@k$ and $IDCG@k$ are defined as

$$DCG@k = \sum_{i=1}^{k}\frac{rel(i)}{log_2(i+1)}, \tag{15a}$$

$$IDCG@k = \sum_{i=1}^{|REL|}\frac{rel(i)}{log_2(i+1)}, \tag{15b}$$

in which i is the index of the i-th highest predicted label. $rel(i)$ represents the corresponding true cascade size of the i-th cascade. $|REL|$ indicates that the results are sorted in descending order based on relevance, and the set of the first k results is selected, allowing for enhanced efficiency.

Both evaluation measures enabled us to examine our model from different perspectives. The MSE determines how accurate our model is in terms of cascade size (when comparing the true and predictive values), while NDCG is used to compare the level of similarity between true and predictive cascade orders in terms of cascade size.

We varied k in {1%, 5%, 10%, 15%} since only about 15% of cascades contain retweets. The distribution is shown in Figure 14.

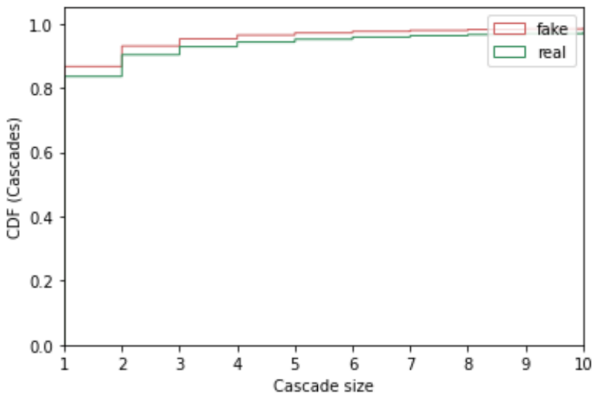


Figure 14. Cumulative distribution function (CDF) of number of posts per cascade for different veracities.

5.2. Performance Comparison

5.2.1. Baselines Performance

To address RQ 1, we conducted experiments on several baselines. From Table 7, we can summarize the following points:

1. Our proposed method, ExoFIA, outperformed the baselines according to the NDCG and MSE metrics. Additionally, our model had a higher hit rate than other models when considering HR@1P and HR@5P. It appears that our model was successful in capturing the outbreak cascades of interest. We also noticed that the simpler model measured better on HR@15P (statistics-based approaches > feature-based approaches > deep-learning-based approaches) because the distribution of this dataset is skewed. The first 15% of the cascades equate to all the cascades with propagation (i.e., the remaining 85% of cascades were not propagated, and their cascade size was 1).
2. When all metrics are taken into consideration, ExoFIA > the future-deep method > the future-linear method. The future-deep method employs multiple non-linear functions to model the relationship between the predicted values and the actual cascade size levels and performs better than the future-linear method. However, it overlooks the dynamic data implicitly stored in networks.
3. When all metrics are considered, ExoFIA > CasCN > Topo-LSTM. All of these models analyze the cascade structure. However, Topo-LSTM does not consider time information, so it performs poorly. ExoFIA performs better than CasCN because ExoFIA incorparates both endogenous factors, such as user history, as well as exogenous signals.
4. The performance of CasCN in terms of hit rate is superior to that of the future-deep method, yet it lags behind with respect to the NDCG metric. This demonstrates that while dynamic features play a crucial role in capturing burst tweets, other features are still needed to effectively accomplish the sorting task.

Table 7. Prediction performance of ExoFIA and baselines when observation time is set to 3 h. The best result in each column is in boldface.

Types	Model Name	HR@1P	HR@5P	HR@10P	HR@15P	NDCG@1P	NDCG@5P	NDCG@10P	NDCG@15P	MSE
Statistics	Node_num_at_T	0.677	0.791	**0.798**	**0.808**	0.712	0.766	0.787	0.784	0.077
Feature -based	Future-linear	0.645	0.778	0.791	0.758	0.711	0.780	0.802	0.845	0.069
	Future-deep	0.707	0.786	0.785	0.757	0.761	0.800	0.825	0.853	0.062
Deep-learning -based	Topo-LSTM	0.699	0.771	0.712	0.728	0.740	0.793	0.818	0.844	0.067
	CasCN	0.720	0.791	0.797	0.729	0.757	0.792	0.818	0.849	0.065
	ExoFIA	**0.753**	**0.799**	0.797	0.754	**0.825**	**0.853**	**0.868**	**0.882**	**0.049**

5.2.2. Ablation Study

In order to address RQ 2, we demonstrated the effectiveness of each module in the ExoFIA model. Table 8 shows the performance of different variants from Section 5.1.2.

Studies have shown that integrating multi-modal information (as in ExoFIA) yields the greatest performance among all variants. If we exclude exogenous factors(ExoFIA-Trend), the performance declines. Likewise, omission of either the social network (ExoFIA-Social) or diffusion network (ExoFIA-Diff) also reduces performance.

Table 8. Prediction performance of variants of ExoFIA when observation time is set to 3 h. The best result in each column is in boldface.

Variants	HR@1P	HR@5P	HR@10P	HR@15P	NDCG@1P	NDCG@5P	NDCG@10P	NDCG@15P	MSE
ExoFIA-Trend	0.724	0.787	0.791	0.753	0.797	0.831	0.849	0.869	0.056
ExoFIA-Social	0.729	0.792	0.791	0.748	0.765	0.807	0.830	0.857	0.061
ExoFIA-Diff	0.715	0.786	0.792	**0.756**	0.774	0.817	0.843	0.869	0.059
ExoFIA	**0.753**	**0.799**	**0.797**	0.754	**0.825**	**0.853**	**0.868**	**0.882**	**0.049**

5.3. Observation Time Window Study

To provide a more comprehensive answer to whether our model is more robust than other models regarding RQ 1, we conducted experiments using observation time windows of various sizes.

For the observation time window T, we made four settings, namely T = 0.5 h, 1 h, 2 h, and 3 h; these corresponded to the time when the popularity reaches roughly 50%, 60%, 65%, and 70% (Fake) (45%, 55%, 60%, and 65%, Real) of the final popularity, respectively, as depicted in Figure 9.

- Observation time window T = 3 h, time interval $\lfloor \frac{T}{n=3} \rfloor$ = 1 h;
- Observation time window T = 2 h, time interval $\lfloor \frac{T}{n=3} \rfloor$ = 40 min;
- Observation time window T = 1 h, time interval $\lfloor \frac{T}{n=3} \rfloor$ = 20 min;
- Observation time window T = 0.5 h, time interval $\lfloor \frac{T}{n=3} \rfloor$ = 10 min.

$\lfloor \cdot \rfloor$ means the floor operation.

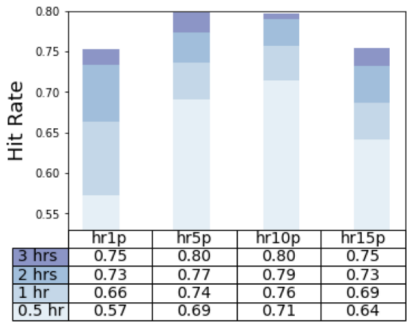
The evaluation results are shown in Table 9. We visualize this table for illustration.

Table 9. Prediction performance within different observation times (T). The best result in each row is in boldface.

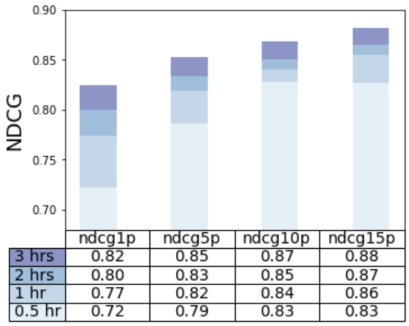
Types	Model Name	T (h)	HR@1P	HR@5P	HR@10P	HR@15P	NDCG@1P	NDCG@5P	NDCG@10P	NDCG@15P	MSE
Statistics -based	Node_ num_at_T	3	0.677	0.791	**0.798**	**0.808**	0.712	0.766	0.787	0.784	0.077
		2	0.656	**0.776**	0.777	**0.787**	0.700	0.748	0.773	0.769	0.083
		1	0.591	0.736	0.719	**0.731**	0.654	0.724	0.746	0.744	0.094
		0.5	0.484	0.586	0.554	0.570	0.613	0.674	0.701	0.703	0.109
Feature -based	Future-linear	3	0.645	0.778	0.791	0.758	0.711	0.780	0.802	0.845	0.069
		2	0.581	0.753	0.778	0.737	0.696	0.769	0.797	0.839	0.072
		1	0.559	0.723	0.754	0.695	0.669	0.754	0.797	0.820	0.078
		0.5	0.484	0.676	0.709	0.653	0.642	0.719	0.777	0.798	0.087
	Future-deep	3	0.707	0.786	0.785	0.757	0.761	0.800	0.825	0.853	0.062
		2	0.681	0.770	0.773	0.735	0.737	0.789	0.817	0.851	0.066
		1	0.640	0.738	0.747	0.689	0.742	0.789	0.837	0.849	0.069
		0.5	0.534	0.687	0.714	**0.654**	0.663	0.733	0.788	0.801	0.085
Deep Learning -based	Topo-LSTM [24]	3	0.699	0.771	0.712	0.728	0.740	0.793	0.818	0.844	0.067
		2	0.677	0.752	0.677	0.710	0.714	0.762	0.789	0.821	0.080
		1	0.613	**0.743**	0.617	0.668	0.692	0.757	0.809	0.820	0.081
		0.5	0.541	0.679	0.568	0.635	0.679	0.746	0.793	0.813	0.083
	CasCN [22]	3	0.720	0.791	0.797	0.729	0.757	0.792	0.818	0.849	0.065
		2	0.685	0.771	**0.793**	0.714	0.725	0.777	0.808	0.836	0.070
		1	0.629	0.738	0.741	0.670	0.689	0.758	0.804	0.814	0.077
		0.5	0.570	0.682	0.694	0.633	0.669	0.729	0.788	0.801	0.085
	ExoFIA	3	**0.753**	**0.799**	0.797	0.754	**0.825**	**0.853**	**0.868**	**0.882**	**0.049**
		2	**0.734**	0.774	0.791	0.732	**0.800**	**0.834**	**0.850**	**0.865**	**0.058**
		1	**0.664**	0.737	**0.758**	0.686	**0.774**	**0.818**	**0.841**	**0.855**	**0.066**
		0.5	**0.573**	**0.691**	**0.715**	0.641	**0.723**	**0.786**	**0.828**	**0.826**	**0.076**

Focusing on just ExoFIA, as visualized in Figure 15, we observe that the value of 1P (prediction on larger cascades) is most affected by observation time. This is because larger

cascades usually have a longer growth period, and thus, a longer observation time window can adequately capture the growth of its lifespan.



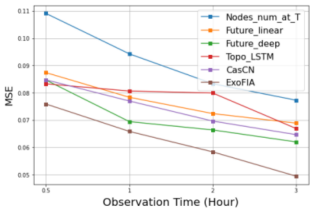
	hr1p	hr5p	hr10p	hr15p
3 hrs	0.75	0.80	0.80	0.75
2 hrs	0.73	0.77	0.79	0.73
1 hr	0.66	0.74	0.76	0.69
0.5 hr	0.57	0.69	0.71	0.64



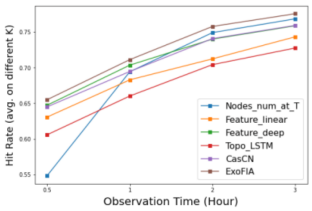
	ndcg1p	ndcg5p	ndcg10p	ndcg15p
3 hrs	0.82	0.85	0.87	0.88
2 hrs	0.80	0.83	0.85	0.87
1 hr	0.77	0.82	0.84	0.86
0.5 hr	0.72	0.79	0.83	0.83

Figure 15. Prediction performance of ExoFIA for top-K cascade sizes under different observation times.

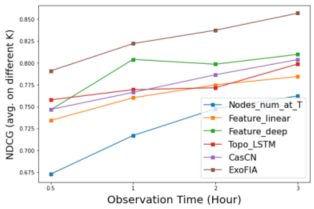
Moreover, as the size of the observation time window increases, the cascade becomes saturated; thus, the prediction errors decline and ranking metrics ascend for all models (see Figure 16). Furthermore, the proposed ExoFIA performs better than these baselines under a variety of circumstances related to different windows, further validating the robustness of our model.



(**a**) MSE



(**b**) Average HR



(**c**) Average NDCG

Figure 16. Comparisons for all competitors under different observation times.

We observe an interesting phenomenon in Figure 16b. Node_num_at_T has the worst performance when the observation time is 0.5 h, but its performance abruptly increases with the observation time and surpasses most models. For clarity, we only chose node_num_at_T and Feature-deep for illustration. In Table 10, we find that as the observation time increases, the *HR@k* (with k varying from small to large) of node_num_at_T gradually exceeds the value of Feature-deep. As the observation time increases, saturation of the cascade is likely to occur, and node_num_at_T will become closer and closer to the final size of the cascade. Most of the cascades are still spreading and growing when the observation period is short. Only relying on the node numbers at instant T is not enough to predict future predictions, so deep learning models must be used to extract representative information. However, when the observation period lasts longer, more cascades stop spreading, and only those with larger sizes continue to grow.).

Table 10. Partial of Table 9. The best result in each row is in boldface.

Model Name	T (h)	HR@1P	HR@5P	HR@10P	HR@15P
Node_ num_at_T	3	0.677	**0.791**	**0.798**	**0.808**
	2	0.656	**0.776**	**0.777**	**0.787**
	1	0.591	0.736	0.719	**0.731**
	0.5	0.484	0.586	0.554	0.570
Future-deep	3	**0.707**	0.786	0.785	0.757
	2	**0.681**	0.770	0.773	0.735
	1	**0.640**	**0.738**	**0.747**	0.689
	0.5	**0.534**	**0.687**	**0.714**	**0.654**

5.4. Interpretability and Case Study

To address RQ 3, we extracted the attention vector from the attention layer and used it to calculate feature importance. $\alpha(s^i)$ denotes the attention vector of the instance i.

$$\alpha = softmax(\boldsymbol{W}_{l_{attn}}\boldsymbol{Z} + \boldsymbol{b}_{l_{attn}}), \alpha \in \mathbb{R}^{1\times D}. \tag{16}$$

We aggregated attention vectors on the instance level [43] as

$$IMP = \frac{1}{S}\sum_{i}^{S}\alpha(s^i), \tag{17}$$

in which $IMP \in \mathbb{R}^{1\times D}$ denotes feature importance.

We evaluated the value of each feature in the construction of an attention network by summing their relevant feature importances. For example, $IMP_{i:i+n-1}$ denotes the n feature importance related to the tweet post, and the feature importance of the post is formulated as $\sum_{k=i}^{i+n-1} IMP_k$. From Figure 17, we can see the importance of each feature set. It is clear that in both real news and fake news datasets, the two most significant feature sets are network and user timeline. Tweet posts and user profiles are less influential, while the trend is the least impactful.

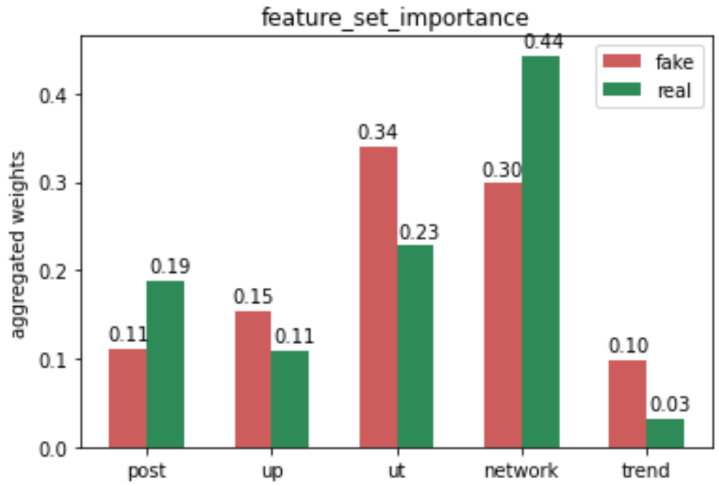


Figure 17. Feature importance for each component on different veracity.

Moreover, we divided the cascades arranged by size into five sequential intervals on the p-quantile ($p = [0, 0.87, 0.95, 0.997, 0.999, 1]$). In Figure 18, the x-axis tick indicates the size of a cascade on a scale of 0 to 4. We wanted to know which feature sets would become more important as the cascades size increased.

Figure 18. Attention weight for each component of different veracities.

In two datasets of different veracities, we found that the same pattern appeared in the changes in the importance of **networks**, **tweet posts, user timeline (history)**, and **trends**, whereas the **user profile** took an opposite route.

Generally, the **network** importance increases then decreases for both datasets because the more significant the post, the less important the dissemination structure of early hours. This implies that the initial structural features become less important as cascades grow, which is consistent with previous work [14].

Trends (public concern) tend to be more important on large-scale than on small-scale cascades. However, both **user timelines** (history) and **tweet posts** decrease as scale increases. The visualization of the chart corresponds with common sense; as the cascade size increases, more attention is paid to the trend rather than the post itself and the user timeline.

In the real dataset, the importance of **user profiles** steadily rises, but in fake datasets, it is just the reverse. We believe that even if the information posted by ordinary people is accurate, very few people will retweet it. As a result, when the cascade size increases, the user's data also becomes more valued.

6. Conclusions and Future Work

6.1. Conclusions

Existing research on online fake news mainly focuses on fake news detection, with few attempts to analyze the dissemination dynamics of fake news on large-scale information networks. This led us to combine rich features to predict the cascade level when sharing a post containing fake news. Our neural framework, ExoFIA, takes into account exogenous information, i.e., extra-Twitter information, which has been rarely considered in previous popularity prediction studies. It adopts attention mechanisms to explore social network characteristics, user characteristics, post content, and public interest. Additionally, we explore the differences in feature importance between the propagation of real and fake news. Comparisons with multiple state-of-the-art prediction models reveal the overall advantages of ExoFIA.

6.2. Future Work

This work presents a prediction model to predict the cascade size of a source tweet with fake news. The cascade size of the source tweet can be treated as the influence or the impact of fake news. However, there is still room for improvement. First, we should consider features that provide rich information on news influence on social media, such as pictures from news articles, the geographical correlations between events and online users, etc. Second, when considering exogenous factors, using TF-IDF to find news keywords is challenging, as they require manual intervention. If the keywords do not map aptly to the news, the trends derived from Google Trends will be inaccurate. In future work, we plan to employ deep learning methods to accurately extract these keywords. Finally, exogenous sources need to be diversified, such as other microblogging platforms (e.g., Facebook and Instagram), video-sharing sites (e.g., YouTube), or news channels. Various kinds of platforms can be used alternately to explore which platform is paramount. Does the combination of different types of platforms result in better prediction performance? Or is it more beneficial to use only one kind of platform?

Limitation

Dataset decay. The drawback is that fake news datasets can quickly become obsolete as the hyperlinks and social media accounts associated with the news at the time of its publication may no longer be accessible due to deletion or privacy issues. We look forward to conquering this limitation if the new version of Twitter API is released in the future.

Author Contributions: Supervision, H.-P.H.; methodology, H.-P.H., P.-X.L. and Y.-Y.H.; validation, Y.-Y.H.; investigation, H.-P.H., P.-X.L. and Y.-Y.H.; writing—original draft preparation, Y.-Y.H. and P.-X.L.; writing—review and editing, H.-P.H. and C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science and Technology Council of Taiwan under grants 111-2636-E-006 -026.

Data Availability Statement: Not applicable.

Acknowledgments: This work was partially supported by the National Science and Technology Council of Taiwan under grants 111-2636-E-006 -026—(NSTC Young Scholar Fellowship).

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

References

1. Gottfried, J.; Shearer, E. News use across social media platforms 2016. *Pew Research Center*, 26 May 2016
2. Allcott, H.; Gentzkow, M. Social media and fake news in the 2016 election. *J. Econ. Perspect.* **2017**, *31*, 211–236. [CrossRef]
3. Thomas, Z. WHO Says Fake Coronavirus Claims Causing 'Infodemic'. *BBC News*, 13 February 2020.
4. Nguyen, V.H.; Sugiyama, K.; Nakov, P.; Kan, M.Y. Fang: Leveraging social context for fake news detection using graph representation. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1165–1174.
5. Ruchansky, N.; Seo, S.; Liu, Y. Csi: A hybrid deep model for fake news detection. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 797–806.
6. Shu, K.; Wang, S.; Liu, H. Beyond news contents: The role of social context for fake news detection. In Proceedings of the twelfth ACM International Conference on Web Search and Data Mining, Melbourne, Australia, 11–15 November 2019; pp. 312–320.
7. Crane, R.; Sornette, D. Robust dynamic classes revealed by measuring the response function of a social system. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 15649–15653. [CrossRef] [PubMed]
8. Gao, S.; Ma, J.; Chen, Z. Modeling and predicting retweeting dynamics on microblogging platforms. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 107–116.
9. Mishra, S.; Rizoiu, M.A.; Xie, L. Feature driven and point process approaches for popularity prediction. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 24–28 October 2016; pp. 1069–1078.
10. Shen, H.; Wang, D.; Song, C.; Barabási, A.L. Modeling and predicting popularity dynamics via reinforced poisson processes. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28.
11. Zhao, Q.; Erdogdu, M.A.; He, H.Y.; Rajaraman, A.; Leskovec, J. Seismic: A self-exciting point process model for predicting tweet popularity. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2015; pp. 1513–1522.
12. Bakshy, E.; Hofman, J.M.; Mason, W.A.; Watts, D.J. Everyone's an influencer: Quantifying influence on twitter. In Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, Hong Kong, China, 9–12 February 2011; pp. 65–74.
13. Zaman, T.; Fox, E.B.; Bradlow, E.T. A bayesian approach for predicting the popularity of tweets. *arXiv* **2014**, arXiv:1304.6777.
14. Cheng, J.; Adamic, L.; Dow, P.A.; Kleinberg, J.M.; Leskovec, J. Can cascades be predicted? In Proceedings of the 23rd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014; pp. 925–936.
15. Gao, S.; Ma, J.; Chen, Z. Popularity prediction in microblogging network. In Proceedings of the Web Technologies and Applications: 16th Asia-Pacific Web Conference, APWeb 2014, Changsha, China, 5–7 September 2014; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2014; pp. 379–390.
16. Xie, D.; Xu, J.; Lu, T.C. What's trending tomorrow, today: Using early adopters to discover popular posts on Tumblr. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2168–2176.
17. Yang, J.; Leskovec, J. Modeling information diffusion in implicit networks. In Proceedings of the 2010 IEEE International Conference on Data Mining, Sydney, Australia, 13–17 December 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 599–608.
18. Naveed, N.; Gottron, T.; Kunegis, J.; Alhadi, A.C. Bad news travel fast: A content-based analysis of interestingness on twitter. In Proceedings of the 3rd International Web Science Conference, Koblenz, Germany, 15–17 July 2011; pp. 1–7.
19. Anderson, A.; Huttenlocher, D.; Kleinberg, J.; Leskovec, J.; Tiwari, M. Global diffusion via cascading invitations: Structure, growth, and homophily. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 66–76.
20. Galuba, W.; Aberer, K.; Chakraborty, D.; Despotovic, Z.; Kellerer, W. Outtweeting the twitterers-predicting information cascades in microblogs. *WOSN* **2010**, *10*, 3–11.
21. Huberman, B.A.; Romero, D.M.; Wu, F. Social networks that matter: Twitter under the microscope. *arXiv* **2008**, arXiv:0812.1045.

22. Chen, X.; Zhou, F.; Zhang, K.; Trajcevski, G.; Zhong, T.; Zhang, F. Information diffusion prediction via recurrent cascades convolution. In Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 8–11 April 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 770–781.
23. Li, C.; Ma, J.; Guo, X.; Mei, Q. Deepcas: An end-to-end predictor of information cascades. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 577–586.
24. Wang, J.; Zheng, V.W.; Liu, Z.; Chang, K.C.C. Topological recurrent neural network for diffusion prediction. In Proceedings of the 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, 18–21 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 475–484.
25. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
26. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
27. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [CrossRef]
28. Juul, J.L.; Ugander, J. Comparing information diffusion mechanisms by matching on cascade size. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2100786118. [CrossRef] [PubMed]
29. Shu, K.; Mahudeswaran, D.; Wang, S.; Liu, H. Hierarchical propagation networks for fake news detection: Investigation and exploitation. In Proceedings of the International AAAI Conference on Web and Social Media, Atlanta, GA, USA, 1–5 June 2020; Volume 14, pp. 626–637.
30. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. *Science* **2018**, *359*, 1146–1151. [CrossRef] [PubMed]
31. Shu, K.; Mahudeswaran, D.; Wang, S.; Lee, D.; Liu, H. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big Data* **2020**, *8*, 171–188. [CrossRef] [PubMed]
32. Zhou, F.; Xu, X.; Trajcevski, G.; Zhang, K. A survey of information cascade analysis: Models, predictions, and recent advances. *ACM Comput. Surv.* **2021**, *54*, 1–36. [CrossRef]
33. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
34. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
35. Myers, S.A.; Zhu, C.; Leskovec, J. Information diffusion and external influence in networks. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, London, UK, 19–23 August 2012; pp. 33–41.
36. Oghina, A.; Breuss, M.; Tsagkias, M.; De Rijke, M. Predicting IMDB Movie Ratings Using Social Media. In Proceedings of the 34th European Conference on IR Research, ECIR 2012, Barcelona, Spain, 1–5 April 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 503–507.
37. Dutta, S.; Masud, S.; Chakrabarti, S.; Chakraborty, T. Deep exogenous and endogenous influence combination for social chatter intensity prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 1999–2008.
38. Masud, S.; Dutta, S.; Makkar, S.; Jain, C.; Goyal, V.; Das, A.; Chakraborty, T. Hate is the new infodemic: A topic-aware modeling of hate speech diffusion on twitter. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 504–515.
39. Kupavskii, A.; Umnov, A.; Gusev, G.; Serdyukov, P. Predicting the audience size of a tweet. In Proceedings of the International AAAI Conference on Web and Social Media, Cambridge, MA, USA, 8–11 July 2013; Volume 7, pp. 693–696.
40. Hutto, C.; Gilbert, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Proceedings of the International AAAI Conference on Web and Social Media, Ann Arbor, MI, USA, 1–4 June 2014; Volume 8, pp. 216–225.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
42. Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; Cheng, X. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1149–1158.
43. Škrlj, B.; Džeroski, S.; Lavrač, N.; Petkovič, M. Feature importance estimation with self-attention networks. *arXiv* **2020**, arXiv:2002.04464.