Prediction of Reposting on X

Ziming Xu, Ingemar J. Cox, *Fellow, IEEE,* Shi Zhou, *Senior Member, IEEE*

***Abstract*—There have been considerable efforts to predict a user’s reposting behaviour on X (formerly Twitter) using machine learning models. The problem is previously cast as a supervised classification task, where Tweets are randomly assigned to a test or training set. The random assignment helps to ensure that the test and training sets are drawn from the same distribution. In practice, we would like to predict users’ reposting behaviour for a set of messages related to a new, previously unseen, topic (defined by a hashtag). In this case, the problem becomes an out-of-distribution generalisation classification task.**

**Experimental results reveal that while existing algorithms, which predominantly use features derived from the content of Tweet messages, perform well when the training and test distributions are the same, these algorithms perform much worse when the test set is out of distribution. We then show that if the message features are supplemented or replaced with features derived from users’ profile and past behaviour, the out-of-distribution prediction is greatly improved, with the F1 score increasing from 0.26 to 0.70. Our experimental results suggest that a significant component of reposting behaviour can be predicted based on users’ profile and past behaviour, and is independent of the content of messages.**

***Index Terms*—X (Twitter), Reposting/Retweet, Prediction, Ma- chine Learning, Out-of-Distribution Generalisation, Feature Im- portance, Online Social Media.**

1. Introduction

**U**

NDERSTANDING and predicting user reposting be- haviour on social media platforms is an area of interest across disciplines such as computer science, social science, po- litical science, and marketing. This interest stems from the in- fluence that individual behaviours have on broader phenomena like information diffusion [1]–[9], including how information is spread during the marketing [4], [5] and political campaigns

[6], [7] or how misinformation and rumours spread [8], [9]. The prediction of reposting has been formulated as a su-

pervised classification task. The objective is to train a binary classifier to estimate the likelihood of a recipient reposting a message from a sender. In this setting, the dataset is typically divided into training and test sets through random assignment, ensuring that both sets are drawn from the same underlying data distribution for fair and stable evaluation.

On social media platforms new topics appear regularly, e.g. those related to breaking news. Thus, in addition to the typical in-distribution classification, out-of-distribution prediction is also needed, where a model is trained on existing data related to previous topics (usually defined by hashtags) and is then

Ziming Xu is with Department of Computer Science, University College London, London WC1E 6BT, U.K. (e-mail: [ziming.xu.22@ucl.ac.uk).](mailto:ziming.xu.22@ucl.ac.uk)

Ingemar J. Cox is with Department of Computer Science, University College London, London WC1E 6BT, U.K., and also with the Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen, Denmark (e-mail: [ingemar@ieee.org).](mailto:ingemar@ieee.org)

Shi Zhou is with AI Centre, Department of Computer Science, University College London, London WC1E 6BT, U.K. (e-mail: [s.zhou@ucl.ac.uk).](mailto:s.zhou@ucl.ac.uk)

used to predict a user’s reaction to a message related to a new, previously unseen, topic. Such a model can be useful for many applications, and it is more challenging, as it involves handling a distributional shift between known and unknown topics.

In this paper, we first review previous work on predicting repostings. Prior work can be categorised by their input features and the classification algorithm used. We identified a total of 305 features used in previous work to predict reposting behaviour on Twitter. We categorised these features into 4 sets:

(i) the set *M* of features derived from the message/post under consideration for reposting, (ii) the set *HM* of features derived from the previous (historical) messages sent by a user, (iii) the set *U* of features derived from the user’s profile, and (iv) the set *HM* of features derived from a user’s history of actions. We note that the majority of prior work only uses features from *M* and *HM* . We consider two classification models: (1) a decision tree model based on the work reported in [10] and

(2) a neural network model based on the work reported in [11].

\*\*\*Why these two? \*\*\*

Our experimental results are based on 2.5 million tweets from 50 thousand users collected in 2022 from the Twitter platform. Our results show that while the prediction mod- els perform well for in-distribution prediction, they perform much worse for out-of-distribution prediction when only using message-related features. When the message-related features are supplemented or replaced with user-related features (*U* and *HU* ) the out-of-distribution performance greatly improves,

e.g. from an F1 score of 0.26 to 0.70

Our work highlights the importance of incorporating user- related features to predict repostings, a set of features that are currently seldom used. Interestingly, our experimental results suggest that a significant component of reposting behaviour is independent of the message content.

1. Background
2. *Reposting on X*

X (formerly Twitter) is one of the most popular social media networks. It allows users to post short messages (also called Tweets), follow other users to curate their content feed, and interact with other users through actions, such as ‘like’ and ‘repost’. When a user posts a message, it is shown to the user’s followers as well as other users who might be interested (as determined by X’s recommendation algorithm).

As shown in Figure 1a, a user can *repost* a message received from another user. A reposting action involves three elements: a message, a sender (or author) of the message, and a recipient that receives and reposts the message. The act of reposting is often considered as an endorsement of a message and its sender. Reposting plays an important role in

User u1 **posted** message m1 containing hashtag #A

u3 received m1 and did not repost it

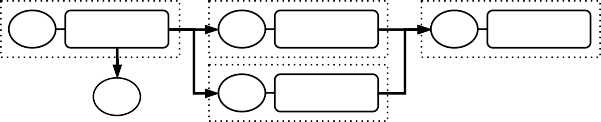
u2 received m1 and **reposted** it

u4 received m1 and **reposted** it

u5 received m2 and m4 and chose to repost m2

research has focused on X due to its widespread use and, until recently, its availability. Additionally, Instagram has a ‘share’ function, and TikTok offers a ‘repost’ feature, but these differ from the repost action discussed here, as they represent privately forwarding messages or posts to specific users or group chats.

1. *Input Data for Reposting Prediction*



u1 m1(#A)

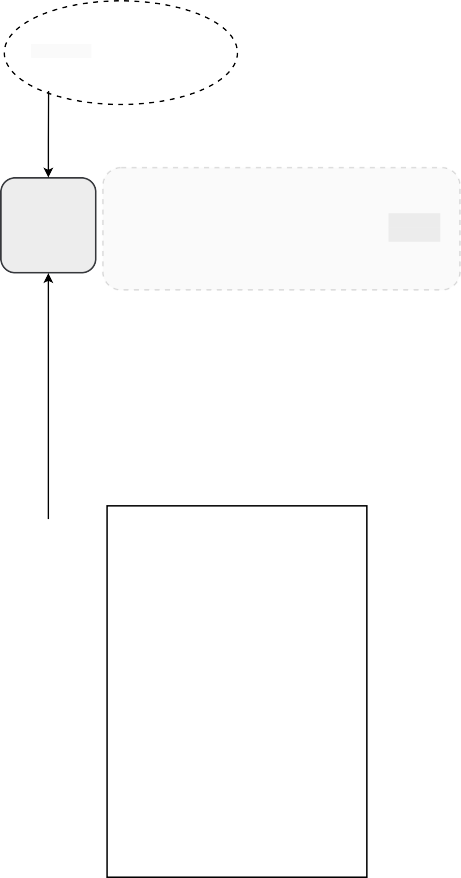
u2 m2(m1)

u5 m5(m2)

u3

u4 m4(m1)

* 1. Reposting on X



**Sender (S)**

User data

**Data(U)**

**Original message**

**Historical messages**

Historical User data

**Data(HU)**

**Reposted message**

Historical Message data

**Data(HM)**

Message data

**Historical messages**

**Data(M)**

**Recipient (R)**

* 1. Input Data from X

Fig. 1. Reposting on X. (a) A flowchart of information spreading via reposting. (b) A real example of reposting, where a user ‘Ziming XU’ (the recipient) reposted a message from ‘BBC Breaking News’ (the sender). Four types of input data for reposting prediction, including the message data, Data(M); the user data, Data(U); the historical message data, Data(HM); and the historical user data, Data(HU), are derived from this reposting example.

information diffusion on social media. For example, almost 40% of messages we collected were repostings. Note that a repost on X was formerly known as ‘retweet’ on Twitter. In this paper, the term repost also includes quote and reply \*\*\* Is this is single action? \*\*\* which are equivalent.

Facebook and Sina Weibo also have similar repost functions, called ‘share’ on Facebook and ‘repost’ on Sina Weibo. While there are studies using datasets from these platforms, most

As shown in Figure 1b and Table I, the following types of input data can be collected from X for reposting prediction.

* Message-related data:
  + *M* – represents the data available in the message currently being considered from reposting by user

U. It includes the textual content of the message, as well as the sender’s and recipient’s (User’s) IDs.

* + *HM* – represents the data available in previous up to 50 (historical) messages sent by user U;
* User-related data:
  + *U* – represents the data available in the User’s profile, which includes the list of the user’s followers;
  + *HU* – represents the data available in public metrics (described shortly \*\*\*Ziming: visible engagement data such as likes and retweets) of historical mes- sages of a user.

*M* and *HM* contain only the textual content of messages. Following previous work, any multimedia content, such as pictures, audio and video, is discarded.

*HU* contains public metrics of historical messages of a user. These metrics include (1) the type of the message, i.e. post or repost (including quote and reply), (2) the number of retweets/quotes/replies/likes the message has received, and (3) any username mentioned in the message. *HU* are considered user-related data because the public metrics disclose no in- formation about message content and only quantify a user’s previous behaviour and interactions with other users.

1. *Reposting Prediction*

The task of reposting prediction is to predict whether a recipient will repost a message received from a sender. Specifically, we want to learn a binary classification model *f* , which is able to predict the label *y* using,

*f* : {*Data*(*U, M, HM, HU* )} → *y* ∈ {0*,* 1} (1)

where *y* = 1 means the recipient will repost the message received from the sender, and *y* = 0 otherwise. The model *f* uses all or some of the above reposting-related data, or features derived from the data. \*\*\* what is the difference between data and features? \*\*\*

1. *Prediction By Machine Learning*

Several machine learning models have been used for re- posting prediction based on features extracted from reposting- related data. These models include Support Vector Machines (SVM) [12], [13], decision trees [10], [12], [14], [15], lo-

gistic regression [16], [17], and factorization models [18]–

[21]. Comparative studies for reposting prediction [10], [12] have shown that decision tree models often outperform other machine learning models.

1. *Prediction by Neural Networks*

Neural network (NN) models can learn directly from raw data and bypass the need for feature extraction. NN models have been used for a series of prediction tasks related to reposting. For example, [22]–[26] predict the next user who is going to repost a given message based on the sequence of previous reposters of the message and the social relations among these reposters; [27], [28] predicts the increment of the size of cascades \*\*\* What’s a cascade? \*\*\* over time;

[29] predicts whether a recipient will repost a given rumour regardless of its sender; [30] predicts the time gap between reposts of a given message; [31] predicts whether a given message from a sender will be reposted (by any user). These prediction tasks consider only one or two of the three elements of reposting, namely the recipient, the message, and the sender.

Reposting prediction predicts whether a recipient will repost a message from a sender, and as such considers all three elements, i.e. the sender, the message, and the recipient. NN models used for this prediction task include [11] which introduced an Attention-based Convolutional Neural Network model, called SUA-ACNN; and more recently [32], which introduced a Bidirectional Long Short-Term Memory (Bi- LSTM). \*\*\* Why do we focus on SUA-ACNN? \*\*\*

1. *Limitations of Existing Works*

Despite the substantial body of existing research on repost- ing prediction, there are limitations. Decision tree models typ- ically used a limited set of primarily message-based features. Similarly, neural network models primarily used message- related data and largely ignored user-related data. Prior work, however, suggests that the profile and past behaviour of users can have a significant influence on their engagement, including repost decisions [12], [13] and comment actions [33]. For example, [13] suggested that those recipients who follow the sender, and have frequent prior interactions through reposts or mentions with the sender, and share common interests with the sender, are more likely to repost a message from the sender. Furthermore, [12] suggested that although the influence of a sender is an important factor in the prediction of the global cascade of a message, it is far less important for individual repost behaviour, where the relation between a sender and a recipient is the key; in other words, ‘*@justinbieber* may not be influential to those users who like information technology a lot’ [12]. \*\*\*IJC: I don’t understand this quotation \*\*\*

A further limitation of prior work is the evaluation method- ology. \*\*\* Ziming is there any prior work on predicting reposts that uses out-of-distribution evalauion? \*\*\* \*\*\* IJC: Need to edit from here \*\*\* Some works [2], [14], [18] incorporated different topics or hashtags as features to predict message popularity or reposting likelihood, but they treated messages on different topics as they were from the same context. By mixing all messages before dividing them into training and test sets, these models create an in-distribution scenario, which means the training and test sets originate from the same distribution [34], [35]. In practice, we would like to predict users’ reposting behaviour for a set of messages related to a new, previously unseen topic. This will cause a distribution

shift from training to test distribution due to variations in topics (defined by hashtags), leading to an out-of-distribution generalisation scenario. The evaluation experiments should include both in-distribution and out-of-distribution predictions. Therefore, we argue that it is crucial to consider all three el- ements related to reposting, and respect the constraints among them. Specifically, we will examine whether the performance of reposting prediction can be improved by supplementing or replacing message-related data and features with user-related data and features, which include user profiles, user activities,

and interactions between a recipient and a sender.

In addition, we will conduct both in-distribution and out- of-distribution predictions. By evaluating how well our mod- els generalise to new, previously unseen topics, we address a critical gap in existing research, where message-focused models often struggle with such distributional shifts. This generalisation scenario tests the robustness of models across different topic contexts. More importantly, it is useful for real- world applications where emerging, unknown topics are very common.

1. Features

X provides a source of data, from which a wide range of features can be extracted for reposting prediction. Previous work on reposting prediction has always considered a rela- tively small subset of the 305 features we identified from the literature [1]–[3], [10], [12]–[14], [17], [18], [21], [36]–[42].

The 305 features identified, consist of 30 user (U) features, 80 Message (M) features, 157 Historical Message (HM) features, and 38 Historical User (HU) features, as summarised in Table I.

1. *User (U) Features*

A user refers to either the Sender or Recipient. User features are extracted from the profile and following list of the user. User features are either numeric or binary. Examples of numeric features include the number of followers/followees of a user [40], [43], the number of lists that a user is included in [12], [13], and the number of messages a user has posted since registration [13], [40], [44]. All examples are normalised by the *accountage*, which is the number of days since a user registered [40].

Examples of binary features include whether a user is verified [12], [18], and whether a URL is provided in a user’s profile [39].

User features extracted from their following lists in- clude indegree [10] and LeaderRank [10], and whether a Sender/Recipient is following a Recipient/Sender [12].

1. *Message (M) Features*

Message features are extracted from the textual content of the message under consideration. These features relate to topic [10], [14], [17], [21], [40], [41], [45], language style

[18], [42], [46], readability [42], sentiment [3], [14], [37],

[47], emotion [1], [3], [14], [36], [48], and nature of hate- speech\*\*\*IJC: what does this mean?\*\*\* [48], [49]. Message

TABLE I

Input Data and Features for Reposting Prediction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input Data | X Data Sources | Feature Type  (# of features) | Feature Sub-type  (# of features) | Example feature | Note |
| Data(U)  – User data | User profile of senders (S)  and recipients (R) | U (30)  – User  features | Profile (24)  Network (6) | U R FollowerNum U S LeaderRank | The number of followers of the Recipient. A measure of the Sender’s influence. |
| Data(M)  – Message data | Text content  of the message being reposted | M (80)  – Message features | Topic (39)  Language (10)  Readability (11)  Sentiment (5)  Emotion (8)  Hate-Speech (4)  Other (3) | M TopicM5 M Grammer1  M Readability1 M Sentiment1 M Emotion1  M Hate2  M Hashtag | Likelihood of a message related to ‘family’ topic. Grammatical correctness of a message.  A measure of difficulty in understanding a message. Negative sentiment score of a message.  The probability of a message expressing anger. A measure of hatefulness of a message.  If a specific hashtag is contained in a message. |
| Data(HM) – Historical  Message data | Text content of the last 50 messages of S and R | HM (157)  – Historical Message features | Topic (78)  Language (20)  ...  Hate-Speech (8)  Similarity (3) | (same as above)  ...  ...  (same as above) HM SR TopicSim | (averaged over last 50 messages of S and R, respectively)  ...  ...  (averaged over last 50 messages of S and R, respectively) Similarity between S’s and R’s topic preference. |
| Data(HU)  – Historical User  data | Public metrics of the last 50 messages of  S and R | HU (38)  – Historical User features | Popularity (8)  Activity (30) | HU R LikedRate HU R MentionS | Average # of ‘Likes’ received by R’s last 50 messages. The number of messages in which R mentioned S. |

features are calculated by various publicly available language models and tools 1 2 3 4 5 6 7 8. A message’s Hashtag is recorded as a categorical label [2], [14], [18]. \*\*\* Need to move elsewhere \*\*\* and each user is anonymised as a random, unique ID number [18].

1. *Historical Message (HM) Features*

HM features are message features calculated and averaged over the last 50 historical messages of a user. Note that some users may have fewer than 50 historical messages in which case all are used. \*\*\* This is not a precise definition \*\*\* Different from the message features of the reposted message, the HM features describe the interest and character of a user, such as the user’s recent interest in a given topic, or the language style that commonly appeared in their recent messages. HM features are calculated for both the sender and the recipient. In addition, 3 HM features measure pairwise topic similarities among the sender, the recipient, and the reposted message [10], [17], [38].

1. *Historical User Activity (HU) Features*

HU features are extracted from the publicly available met- rics associated with a user’s historical messages. Some HU

1Twitter-roBERTa-base: https://huggingface.co/cardiffnlp

2Latent Dirichlet Allocation model from scikit-learn

3language-tool-python: https://github.com/jxmorris12/language tool python

4TextBlob: https://github.com/sloria/textblob

5A Logistic Regression model trained on this public data source: [https://www.kaggle.com/datasets/crowdflower/twitter-user-gender-](http://www.kaggle.com/datasets/crowdflower/twitter-user-gender-) classification

6Readability: https://github.com/andreasvc/readability/

7NLTK-VADER: [https://www.nltk.org/index.html](http://www.nltk.org/index.html)

8A Transformer-based library for SocialNLP tasks to measure the emotion and hate-speech: https://github.com/pysentimiento/pysentimiento

features measure how popular a user is, e.g. the average number of likes/retweets/quotes/replies that the user has re- ceived. Some HU features can describe the pattern of a user’s behaviour. For example, the percentage of a user’s historical messages that are reposts [10], [17], [18] and the average time interval between a user’s posts [18]. Other HU features quantify the interactions between a Sender and a Recipient,

e.g. the ratio of a sender’s historical messages that have mentioned a recipient [12], [13], and vice versa.

Details of the 305 features are provided in the Appendices.

1. Reposting Prediction Models

We considered two models for predicting reposts; a decision tree model and a neural network model.

Recent work [10] has shown that decision trees remain a very competitive choice for determining whether a message will be reposted. In [10] a C5.0 decision tree was used for reposting prediction based on 37 features. We implemented this method using both C5.0 and XGBoost decision trees. There is little difference in performance \*\*\*XGBoost better?

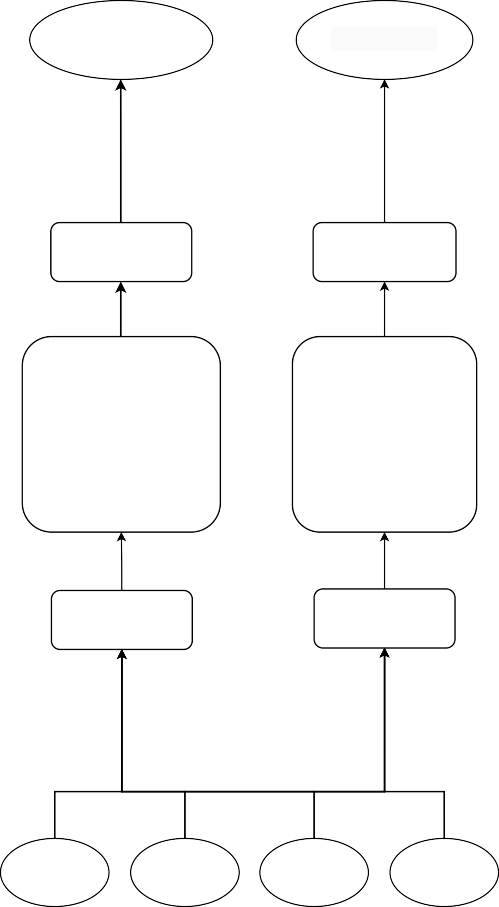
\*\*\* and we therefore only report experimental results using XGBoost. The original work in [10] is referred to as TORS

\*\*\* what does this stand for \*\*\*. Our experimental results report results for TORS and XGBoost decision trees using various subsets of the 305 features we previously described.

We also implemented a neural network architecture based on SUA-ACNN \*\*\* why? \*\*\*

=== begin delete?

1. *Decision Tree Models*
   1. *Baseline: the TORS model [10]:* This decision tree model is the most recent feature-based machine learning model for reposting prediction. The model focused on exploring the



**Prediction**

**Prediction**

**FR-DT**

**TORS**

Decision Tree

Decision Tree

**305 features**

including: 30 U-features,

38 HU-features,

157 HM-features,

80 M-features.

**37 features**

including: 2 U-features,

13 HU-features,

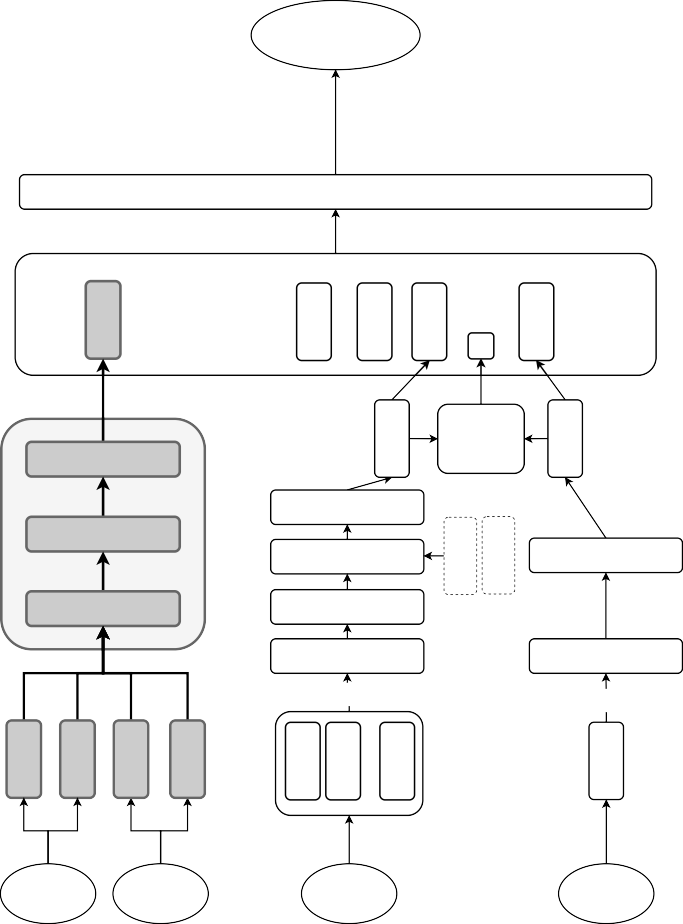
12 HM-features,

10 M-features.

Feature extraction

Feature extraction

Data(U) Data(HU) Data(HM) Data(M)



**Prediction**

**DR-NN**

**SUA-ACNN**

Dense layer

Concatenation layer

**vu**

vs vr vi

vm

s

**MLP**

vi

Similarity matrix

vm

**Dense layer**

Folding layer

**Dense layer**

Attention layer vs

vr

Pooling layer

**Dense layer**

Pooling layer

Convolution layer word2vec

Convolution layer

word2vec

**us ur hs hr**

m1 m2 m.. n

m

Data(U) Data(HU)

Data(HM)

Data(M)

(a) FR-DT and TORS (b) DR-NN and SUA-ACNN.

Fig. 2. Diagrams of the four models under study for reposting prediction. (a) Two decision tree models, including our Feature-Rich Decision Tree (FR-DT) model and the baseline model TORS. The FR-DT model uses 305 features, whereas the TORS model uses only 37 features. (b) Two neural network models, including our DR-NN model and the baseline model SUA-ACNN. The SUA-ACNN model uses only message-related data, i.e. Data(M, HM), whereas our DR-NN model uses user-related data, i.e. Data(U, HU) as well.

‘Topic-Oriented Relationship Strength’ between users. The model used 37 features. Apart from the 2 user features, all other features are related to topics, including 10 message features on topics of reposted messages; 12 historical message features on user’s topic preference and similarity of topic preference between users; and 13 historical user features on users’ interactions on different topics. In this paper, we will use TORS as a baseline model for comparison with our feature- rich decision tree model.

* 1. *Our Feature-Rich Decision Tree (FR-DT) model:* Figure 2a shows our feature-rich decision tree (FR-DT) model using a comprehensive list of 305 features. The main difference between our model and the baseline TORS model is that we use eight times more features and, more importantly, encompass a wider range of dimensions beyond message

topics. 9

1. *Neural Network Models*
   1. *Baseline: the SUA-ACNN model [11]:* This Attention- based Convolutional Neural Network model generates not only the embedding of reposted messages and historical messages but also the embedding of users. The model only uses message-related data, i.e. message data, Data(M), and historical message data, Data(HM). In this paper, we will use SUA-ACNN as a baseline model for comparison with our data- rich neural network model.
   2. *Our Data-Rich Neural Network (DR-NN) Model:* As shown in Figure 2b, our data-rich model is an extension of the SUA-ACNN model, where we use a Multilayer Perception (MLP) network to include user-related data, i.e. user data,

9A minor difference is that TORS used the C5.0 decision tree, whereas our model used the XGBoost decision tree which is faster. Our additional experiments showed that all observations and conclusions in this paper would be the same if both models use C5.0 or XGBoost.

Data(U), and historical user data, Data(HU). The fact that the SUA-ACNN is a component of our DR-NN model allows us to investigate the effect of adding user-related data for reposting prediction.

=== end delete?

1. Reposting Prediction

TABLE II

Decision Tree Hyperparameters of the FR-DT model

1. *Our X Data*

We collected data from X using the former Twitter API with the Academic Access between July 27th and August 14th in 2022. We only collected messages on particular topics, determined by corresponding hashtags. The topics/hashtags were chosen \*\*\* how \*\*\* This resulted in 14 topics listed in Table VI. We obtained 111,401 Tweets containing one of these 14 hashtags during the collection period (see Table VI). posted by 79,707 unique users around the world.

From these messages, we identified 44,014 reposted mes- sages where the reposting occurred within 24 hours or receiv- ing the original post. \*\*\* Need to move this early where we define what a reposting is. Does anyone else require a 24hour time period? \*\*\* A repost within 24 hours is more likely to reflect the immediate, direct impact of the original message on the recipient than a long-delayed repost which might be influenced by many other external factors, such as exposure to the same information via other media or platforms.

For each of the 79,707 unique users, we collected their profile, following list, and their 50 latest historical messages. If a user had sent less than 50 historical messages, we collected all that they sent. A total of 3,875,312 historical messages were collected.

1. *Datasets for Reposting Prediction*

For reposting prediction, a positive instance {*mh, s, r*} is a reposted message in our X data, where a message *m* (containing a hashtag *h*) was posted by a sender *s* and was then reposted by a recipient *r* (within 24 hours).

A corresponding negative instance is {*m*∗ *, s*∗*,* !*r*}, where a message *m*∗ (containing the same hashtag *h*) was posted by a sender *s*∗ but was *not* reposted by the recipient *r*. Note that a recipient may receive two identical messages from two

*h*

*h*

senders, *S*1 and *S*2. If the recipient reposts the message from

*S*1, sender *S*2 receives no credit.

We created three datasets with different ratios of positive to negative instances.

* + 1:1 dataset: For each positive instance, we selected one negative instance from our data, whose feature vector [*Mm*∗ *, Us*∗ ], containing all features of the message and the sender, has the smallest cosine distance from that of the positive instance, [*Mmh , Us*].

*h*

* + 1:5 dataset: Similar to the 1:1 dataset, for each positive

instance, we select *five* negative instances that are most similar to the positive instance in terms of the cosine distance between their feature vectors.

* + 1:10 dataset: In addition to the 1:5 dataset, for each positive instance, we add 5 randomly chosen negative instances (that are not already in the dataset). We chose not to select the 10 closest negative examples in order to

R = #(negative instances) / #(positive instances) Values marked with an \* are most frequently used.

|  |  |
| --- | --- |
| Hyperparameter | Searched values |
| *max depth* | 6, 7, 8\*, 9, 10 |
| *learning rate* | 0.3\*, 0.35, 0.4 |
| *n estimators* | 100\*, 150, 200 |
| *min child weight* | 1\*, 2, 3 |
| *subsample* | 0.8, 0.9, 1.0\* |
| *scale pos weight* | 1, 0.3R\*, 0.5R, 0.7R, 0.9R |

intriduce some diversity and to better reflect real-world scenarios where negative instances might not always be highly similar to positive instances.

In total, these datasets for reposting prediction consist of 44,014 positive instances along with their corresponding negative instances, involving 70,004 messages, 49,703 users, and 2,462,449 historical messages. \*\*\*IJC: this is not useful. We need the details for each daats set 1:1, 1:5 ... \*\*\*

1. *Experiment Settings for In-Distribution Predictions*

We conducted in-distribution prediction with the four mod- els using the above 3 datasets.

For each dataset, we performed Monte Carlo cross vali- datation 10 time, each randomly splitting the dataset 70/30 to the training and test sets. we selected decision tree hyperpa- rameters for the FR-DT model using 5-fold cross-validation (see Table II, parameter values marked with an \* are most frequently used). The TORS decision tree, we used the same settings as in the original work [10]. For the full feature set, the hyperparameters of the decision tree were determined by *k*-fold cross validation on the training data. \*\*\*How were the final parameters chosen? \*\*\* The SUA-ACNN model, which itself is a component of the DR-NN model, was configured in the original work [11]. For the additional MLP component in the DR-NN model, the number of units of the three dense layers are set to 128, 128 and 64, respectively. The activation functions are set to *ReLU*.

Each of the 10 runs is evaluated by the F1 score =

*TP/*( *TP* + 1*/*2(*FP* + *FN* ) ), where TP, FP and FN are the true positive, false positive, and false negative, respectively. We report the average and standard deviation of F1 scores from the 10 runs.

1. *Results for In-Distribution Predictions*

In-distribution prediction results of the four models are shown in Table III.

For the two Decision Tree (DT) models our FR-DT model shows significantly improved performance than the baseline TORS model. The performance of both decision tree models degrades as the class imbalance of the training set increase, but the FR-DT model always exhibits better performance (9.6% improvement for 1:1 to 23% for 1:10).

The performance of the two Neural Network (NN) models is slightly worse than for the decision tree models. Once again he performance of both models degrades as the class imbalance

TABLE III

Results for In-Distribution Predictions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Decision Tree models | | Neural Network models | |
|  | Our model | Baseline | Our model | Baseline |
| Model name | **FR-DT** | **TORS** | **DR-NN** | **SUA-ACNN** |
| Input data | 305 features | 37 features | Data(U,HU,M,HM) | Data(M,HM) |
| Dataset(1) | F1 score(2) | | | |

**Loose evaluation**: 70% vs 30% random split cross-validation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1:1 Dataset | **0.932**±0.002 | 0.850±0.003 | **0.914**±0.002 | 0.867±0.003 |
| 1:5 Dataset | **0.883**±0.002 | 0.733±0.004 | **0.851**±0.004 | 0.743±0.007 |
| 1:10 Dataset | **0.834**±0.003 | 0.677±0.004 | **0.801**±0.006 | 0.646±0.011 |

**Strict evaluation**: Walk-forward timewindows cross-validation without overlapping user pairs.

(1) Each dataset is named by the ratio of positive instances to negative instances (see Section V-B).

(2) The average F1 score and standard deviation are calculated over 10

runs of predictions, where each run randomly splits data for training and testing in a ratio of 7:3.

of the training set increase, but the DR-NN model always exhibits better performance (5.4% improvement for 1:1 to 24%

for 1:10).

as expected, the F1 scores of all the models decline with an increased proportion of negative instances in the training and testing data.

To analyse the relative importance of the 305 features used in the FR-DT model, we obtained the importance value

\*\*\*defined as ?? \*\*\* of each feature. The feature importance values are normalised such that the sum of values of the 305 features equals to one, and are listed in Table IV for the 1:5 dataset.

Table IV shows the top 30 features of the FR-DT model. While the 30 features are almost evenly split between 16 user- related (5 user (U), 11 historical user (HU)) and 14 message- related (3 message (M) and 11 historical message (HM)), we note that the top 9 features are all user-related, and the top-2 features account for almost 33% of the normalised importance. We note that all message-related features have small impor-

tance values ≤ 0*.*008.

To further investigate the relative importance of user and messages-related features, we constructed 2 additional de- cision trees. The first was only provided with user-related features (30 user features and 38 historical user features) and the second only message-related features (80 message features and 157 historical message features). Table V shows the performance for in-distribution prediction by the two models on the 1:5 dataset.

While the AllDT model, using all of the 305 features, achieves an F1 score of 0.883, the UHUDT model achieves an F1 score of 0.850, equal to the TORSDT, and 8.3% better that the MHMDT that only uses message-related features

(F1=0.785).

This was further refined by constructed models based only on User features (UDT), historical user features (HUDT), message features (MDT) and historical message features (HMDT). We observe that the performance of UHUDT is greater than HUDT which is greater than UDT. Similar relative performance is observed for message-related features. In all cases, the user-related model outperforms teh corre- sponding message-related model.

Despite the emphasis on message text in previous studies, our results indicate that user-related features are more relevant than the message-related features for reposting prediction for the in-distribution setting. Nevertheless, the combination of user *and* message-related features outperforms either.

\*\*\* For me, I’m still confused as to how to interpret this. The two sets of data are complementary yet either one does ok. \*\*\*

\*\*\* I’m not sure whether to delete this analysis \*\*\*

In this work, we studied a comprehensive list of 305 features and used them for reposting prediction. This large number of features is a computational burden; and moreover, it is unlikely that all of these features are relevant or necessary for reposting prediction.

Table V and Figure 3 show the performance of in- distribution prediction by the FR-DT model using only the top *n* features. In the beginning, the F1 score improves rapidly when *n* increases from 10 to 15 and 20; then the improvement slows down significantly when more and more features are used; and finally when *n* ≥ 90, the F1 score decreases slightly and becomes stable. This means the top 90 features contribute positively to the in-distribution prediction, features ranked between 90th and 100th may contain noisy or even disruptive

TABLE IV

Top 30 Features of the FR-DT model

TABLE V

Feature Analysis \*\*\* This table should become 2 tables \*\*\*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Number of features | Sum of import. value | Average import. value | F1  Score\* |
| FR-DT model | 305 | 1.0000 | 0.0033 | 0.883 |

FR-DT model using selected types of features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| U+HU | 68 | 0.5434 | 0.0080 | 0.850 |
| HU | 38 | 0.2480 | 0.0065 | 0.818 |
| U | 30 | 0.2955 | 0.0098 | 0.799 |
| M+HM | 237 | 0.4566 | 0.0019 | 0.785 |
| HM | 157 | 0.2830 | 0.0018 | 0.767 |
| M | 80 | 0.1736 | 0.0022 | 0.777 |

FR-DT model using top *n* features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Top 10 | 10 | 0.4466 | 0.0447 | 0.813 |
| Top 15 | 15 | 0.4810 | 0.0321 | 0.840 |
| Top 20 | 20 | 0.5096 | 0.0255 | 0.881 |
| Top 30 | 30 | 0.5542 | 0.0185 | 0.883 |
| Top 40 | 40 | 0.5905 | 0.0148 | 0.884 |
| Top 50 | 50 | 0.6233 | 0.0125 | 0.886 |
| Top 60 | 60 | 0.6537 | 0.0109 | 0.886 |
| Top 70 | 70 | 0.6815 | 0.0097 | 0.888 |
| Top 80 | 80 | 0.7075 | 0.0088 | 0.889 |
| Top 90 | 90 | 0.7321 | 0.0081 | 0.889 |
| Top 100 | 100 | 0.7556 | 0.0076 | 0.883 |

\* F1 score is averaged over 10 runs of in-distribution predictions, where each run randomly splits data for training and testing in a ratio of 7:3.

prediction.

It is notable that the decision tree model using only the top 30 features can obtain the same prediction performance (with F1=0.883) as achieved by the FR-DT model using all the 305 features. Such a simplified model has the advantage of easier data collection and more efficient model training.

0.900

0.883

0.880

0.860

F1 Score

0.840

0.820

0.800

10 15 20 30 40 50 60 70 80 90 100 305

n: the number of top features used

1. Out-of-Distribution Prediction

Our X data is comprised of messages containing 1 of the 14 trending hashtags that were popular during the collection period. As noted earlier, previous work on predicting repost- ing has constructed training sets in which all hashtags are represented. We have referred to this as in-distribution. In practice, predicting repostings is likely to be more valuable at the commencement of a new, previously unseen topic, i.e. a topic defined by a hashtag that is not included in the training set. We refer to this as out-of-distribution prediction. The out-of-distribution prediction allows us to assess a model’s generalisation ability, i.e. its capability to predict reposts on a previously unseen topic.

F1 score

Standard deviation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature name | | Rank | Feature  Type | Feature  Sub-Type | Import.  value | TORS |
| U | R FollowS | 1 | U | network | 0.225 |  |
| HU | S LikedRate | 2 | HU | popularity | 0.104 |  |
| HU | S RetweetedRate | 3 | HU | popularity | 0.022 |  |
| HU | S InteractivePer | 4 | HU | activity | 0.021 |  |
| HU | R MentionSPer | 5 | HU | activity | 0.016 |  |
| HU | S TweetNum | 6 | HU | activity | 0.016 |  |
| HU | S RepliedRate | 7 | HU | popularity | 0.013 |  |
| HU | S QuotedRate | 8 | HU | popularity | 0.012 |  |
| U | S Indegree | 9 | U | network | 0.009 |  |
| HM | S Readability7 | 10 | HM | readability | 0.008 |  |
| U | S FollowR | 11 | U | network | 0.008 |  |
| M | TopicLDA7 | 12 | M | topic | 0.007 | ✓ |
| HM | S MasculinityPer | 13 | HM | language | 0.007 |  |
| HM | S TopicM7 | 14 | HM | topic | 0.006 |  |
| HM | S TopicM18 | 15 | HM | topic | 0.006 |  |
| HM | S TopicM19 | 16 | HM | topic | 0.006 |  |
| HU | R MentionS | 17 | HU | activity | 0.006 |  |
| HU | S TweetPercent | 18 | HU | activity | 0.006 |  |
| U | S FolloweeNumDay | 19 | U | profile | 0.005 |  |
| HU | R RepostLatency | 20 | HU | activity | 0.005 |  |
| HU | S RetweetPercent | 21 | HU | activity | 0.005 |  |
| M | TopicLDA4 | 22 | M | topic | 0.005 | ✓ |
| HM | S Irony | 23 | HM | language | 0.005 |  |
| HM | R TopicLDA5 | 24 | HM | topic | 0.005 | ✓ |
| U | R Indegree | 25 | U | network | 0.004 | ✓ |
| HM | S TopicM10 | 26 | HM | topic | 0.004 |  |
| M | Readability11 | 27 | M | readability | 0.004 |  |
| HM | S Hate2 | 28 | HM | hate-speech | 0.004 |  |
| HM | S TopicM11 | 29 | HM | topic | 0.004 |  |
| HM | S TopicGNum | 30 | HM | topic | 0.004 |  |

We conducted out-of-distribution prediction for each of the 14 hashtags in our X data. For each given hashtag, we trained a model with data of other 13 hashtags (i.e. known topics), and then tested the model with data of the given hashtag (the new, unseen topic).

Fig. 3. F1 score of in-distribution prediction by the FR-DT model using the top *n* features only. The F1 score and standard deviation are averaged over 10 runs of in-distribution predictions. The dashed line is F1 score = 0.883, which is achieved by the FR-DT model using all the 305 features. \*\*\*This figure is misleading because the x-axis is NOT linear \*\*\*

information; and the remaining features are irrelevant to the

For each out-of-distribution prediction for a given hashtag, we carried out the Leave-One-Out cross-validation, which allowed us to assess the robustness of a model’s prediction for this hashtag. We randomly divided the training data (i.e./,data of other 13 hashtags) into 10 subsets, and left one subset out and used the other 9 subsets to train the model and obtain prediction results. Then we repeated the process 10 times

TABLE VI

Results for Out-of-Distribution Predictions (Generalisation Test)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Decision Tree models | | Neural Network models |
| Model | **FR-DT** | **TORS** | **DR-NN** |
| Number of Features (Types)  Input Data | 305 (ALL) 68 (U+HU) 237 (M+HM)  – – – | 37  – | – – –  D(U,HU,M,HM) D(U,HU) D(M,HM)(3) |
| For each hashtag, training data:  Testing data: | data of all other hashtags excluding the hashtag under study;  data of the hashtag under study. | | |
| Hashtag | **F1 Score**(1)(2) | | |
| #BLM  #Covid #NHS  #Brexit #Monkeypox  #IOS16  #Climatechange  #Supercup #Olivianewtonjohn #CostofLivingCrisis  #Avengers #UkraineWar #EnergyPrices  #LoveIsland | **0.821**∗±0.010 0.797±0.021 0.262±0.101  **0.807**∗±0.009 0.793±0.007 0.327±0.034  **0.798**±0.042 0.719±0.073 0.416±0.086  **0.749**∗±0.017 0.715±0.017 0.173±0.022  **0.729**∗±0.009 0.723±0.005 0.322±0.022  0.726±0.022 **0.763**∗±0.007 0.168±0.035  0.705±0.065 **0.762**±0.008 0.245±0.041  0.702±0.024 **0.734**±0.012 0.231±0.081  0.679±0.058 **0.748**∗±0.014 0.312±0.027  **0.654**∗±0.011 0.626±0.022 0.207±0.061  0.635±0.031 **0.668**±0.017 0.187±0.022  **0.625**∗±0.028 0.534±0.028 0.313±0.071  **0.609**∗±0.021 0.584±0.015 0.204±0.035  **0.589**∗±0.018 0.583±0.009 0.233±0.074 | 0.316±0.076  0.286±0.014  0.118±0.023  0.266±0.040  0.221±0.021  0.159±0.032  0.208±0.025  0.159±0.050  0.276±0.031  0.113±0.013  0.169±0.025  0.294±0.074  0.186±0.033  0.097±0.014 | **0.756**±0.027 0.753±0.027 0.251±0.029  0.691±0.018 **0.755**±0.009 0.139±0.028  0.696±0.093 **0.804**∗±0.007 0.386±0.232  0.640±0.020 **0.700**±0.018 0.176±0.076  0.679±0.020 **0.688**±0.010 0.163±0.040  0.679±0.018 **0.726**±0.024 0.092±0.025  0.717±0.014 **0.768**∗±0.006 0.143±0.063  **0.751**∗±0.012 0.707±0.018 0.329±0.121  0.613±0.031 **0.726**±0.017 0.226±0.063  0.594±0.033 **0.644**±0.017 0.089±0.028  0.535±0.016 **0.704**∗±0.015 0.153±0.089  0.441±0.057 **0.505**±0.007 0.091±0.018  0.472±0.042 **0.513**±0.028 0.096±0.049  0.465±0.044 **0.582**±0.013 0.221±0.024 |
| F1 score averaged over hashtags | **0.702** 0.696 0.257 | 0.205 | 0.623 **0.684** 0.183 |

(1) The F1 score and standard deviation are obtained using a ‘Leave-One-Out’ approach, where for each hashtag, data from all other hashtags are divided into ten subsets, and training was repeated ten times, each time excluding a different subset.

(2) When random guessing, i.e. randomly assign positive and negative labels with a ratio of 1:5, the F1 score would be 0.167. Details of random

guessing can be found in Appendices.

(3) The DR-NN model using message-related input data, i.e. D(M, HM), is exactly the SUA-ACNN model.

by leaving a different subset each time. We evaluated the 10 prediction results using the same testing set (i.e./,data of the given hashtag), and obtained the average F1 score and the standard deviation.

We conducted the out-of-distribution prediction for each of the 14 hashtags using each of the four models on the 1:5 dataset. Results are shown in Table VI.

We observe that the TORS and SUA-ACNN models show notably poor performance in the out-of-distribution prediction with average F1 scores of 0.205 and 0.183, which are only slightly better than random guessing (0.167) 10. In contrast, the FR-DT model and the DR-NN model perform much better with average F1 scores of 0.70 and 0.62. As expected, the out-

1.0

0.8

0.6

F1 Score

0.4

0.2

0.0

U+HU

In-Distribution

Out-of-Distribution

+M+HM

U+HU HU U M+HM HM M D(U,HU, D(U,HU) D(M,HM)

M,HM)

b) DR-NN with

of-distribution performance is worse that the in-distribution

a) FR-DT with selected type(s) of features

selected input data

performance (0.?? and 0.??).

We also explored variations of FR-DT and DR-NN models by using selected types of features/data, results are shown in Table VI and Figure 4. Models using only message-related fea- tures/data perform poorly as discussed. For the FR-DT model, supplementing user-related features with message-related fea- tures has negligible impact on prediction performance, the difference in F1 score is less than 0.01. \*\*\* This is NOT true if you look at individual results (per tag) \*\*\* Interestingly, DR- NN demonstrates higher predictive performance when using

10When random guessing, i.e. randomly assign positive and negative labels with a ratio of 1:5, the F1 score would be 0.167. Details of random guessing can be found in Appendices.

Fig. 4. Comparison of in-distribution and out-of-distribution predictions for

a) the FR-DT decision tree model with selected types of features and b) the DR-NN neural network model with selected input data.

only user-related data compared to when all data are used, with the F1 score increasing from 0.623 to 0.684.

Our results for out-of-distribution generalisation predictions indicate that supplementing or replacing message-related fea- tures with user-related features derived from users’ profiles and past behaviour can significantly enhance a model’s ability to generalise across different topics on X.

1. Conclusion

Reposting prediction is a fundamental problem in modelling information diffusion. We identified 305 features from prior work, though most papers only used a small subset of such. We characterized these features into four categories, namely User, Historical User, Message, and Historical Messages. We then investigated the prediction capability of models trained on subsets of these features using both decision tree (XGBoost) and neural network architectures.

Previous work on predicting repostings has been based on a training set that includes messages relating to (on the same topic) as the message under consideration for reposting. We refer to this as in-distribution testing. Our experimental results based on ... demonstrated that best performance (F1 score ??) is achieved with XGBoost using all 305 features. This is ??% better than the state-of-the-art predictor of [**?**] compares to an F This problem is previously formulated as a supervised in-distribution classification task. While existing message- focused methods perform well in in-distribution scenarios, they show poor performance in out-of-distribution tasks. To ensure comprehensive evaluation while addressing practical needs, we argue that reposting predictions should include both in-distribution and out-of-distribution assessments.

We did a comprehensive study on all three elements re- lated to reposting, namely the message, the sender, and the recipient, leading to the development of feature-rich and data- rich models, which utilize both message-related and user- related data and features. Our experimental results on a set of 49,703 unique users, consisting of approximately 2.5 million messages, demonstrate that our feature-rich and data-rich mod- els outperform message-focused baselines, especially when generalising to unseen topics of reposts. Our findings suggest that a significant component of users’ reposting decisions can be predicted based on users’ past behaviour and is independent of message content.

It is interesting to further explore the role of message and user in influencing reposting decisions. The future work includes investigating the specific motivations behind user reposting behaviour and assessing how these factors vary over time. The prediction of the spreading path is also interesting and valuable for real-world applications. These focused studies will enrich the understanding of the broader dynamics of information spreading.

Acknowledgments

The authors thank Ziwen Li, Berkem Billuroglu and Pei Lo for their contribution in the X data collection and prelim- inary analysis during their study at Department of Computer Science, UCL.

References

1. S. B. Paletz, M. A. Johns, E. E. Murauskaite, E. M. Golonka, N. B. Pandzˇa, C. A. Rytting, C. Buntain, and D. Ellis, “Emotional content and sharing on facebook: A theory cage match,” *Sci. Advances*, vol. 9, no. 39, p. eade9231, 2023.
2. X. Chen, X. Zhou, J. Chan, L. Chen, T. Sellis, and Y. Zhang, “Event popularity prediction using influential hashtags from social media,” *IEEE Trans. Knowl. and Data Eng.*, vol. 34, no. 10, pp. 4797–4811, 2020.
3. C. E. Robertson, N. Pro¨llochs, K. Schwarzenegger, P. Pa¨rnamets, J. J. Van Bavel, and S. Feuerriegel, “Negativity drives online news consump- tion,” *Nature Human Behaviour*, vol. 7, no. 5, pp. 812–822, 2023.
4. S. Tu and S. Neumann, “A viral marketing-based model for opinion dynamics in online social networks,” in *Proc. ACM Web Conf.*, 2022,

pp. 1570–1578.

1. E. Turkel, “Regulating online political advertising,” in *Proc. ACM Web Conf.*, 2022, pp. 3584–3593.
2. A. Haque and M. P. Singh, “Newsslant: Analyzing political news and its influence through a moral lens,” *IEEE Trans. Comput. Social Syst.*, 2024.
3. O. Papakyriakopoulos and E. Goodman, “The impact of twitter labels on misinformation spread and user engagement: Lessons from trump’s election tweets,” in *Proc. ACM Web Conf.*, 2022, pp. 2541–2551.
4. Y. Bai, Y. Liu, and Y. Li, “Learning frequency-aware cross-modal interaction for multimodal fake news detection,” *IEEE Trans. Comput. Social Syst.*, 2024.
5. Z. Jiang, X. Chen, J. Ma, and S. Y. Philip, “Rumordecay: rumor dissemination interruption for target recipients in social networks,” *IEEE Trans. Syst., Man, and Cybern.: Syst.*, vol. 52, no. 10, pp. 6383–6395, 2022.
6. H. Zhu, X. Yang, and J. Wei, “Path prediction of information diffusion based on a topic-oriented relationship strength network,” *Inf. Sciences*, vol. 631, pp. 108–119, 2023.
7. Q. Zhang, Y. Gong, J. Wu, H. Huang, and X. Huang, “Retweet prediction with attention-based deep neural network,” in *Proc. 25th ACM Int. Conf. Inf. and Knowl. Manage.*, 2016, pp. 75–84.
8. Z. Xu and Q. Yang, “Analyzing user retweet behavior on twitter,” in *2012 IEEE/ACM Int. Conf. Advances in Social Networks Anal. and Mining*, 2012, pp. 46–50.
9. Z. Luo, M. Osborne, J. Tang, and T. Wang, “Who will retweet me? finding retweeters in twitter,” in *Proc. 36th Int. ACM SIGIR Conf. Res. and Develop. in Inf. retrieval*, 2013, pp. 869–872.
10. S. N. Firdaus, C. Ding, and A. Sadeghian, “Retweet prediction based on topic, emotion and personality,” *Online Social Networks and Media*, vol. 25, p. 100165, 2021.
11. J. Yang, Z. Wang, F. Di, L. Chen, C. Yi, Y. Xue, and J. Li, “Propagator or influencer? a data-driven approach for evaluating emotional effect in online information diffusion,” in *Proc. 2017 IEEE/ACM Int. Conf. Advances in Social Networks Anal. and Mining*, 2017, pp. 836–843.
12. J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, “Social influence locality for modeling retweeting behaviors,” in *Proc. 27rd Int. Joint Conf. Artif. Intell.*, 2013.
13. L. Chen and H. Deng, “Predicting user retweeting behavior in social networks with a novel ensemble learning approach,” *IEEE Access*, vol. 8,

pp. 148 250–148 263, 2020.

1. W. Feng and J. Wang, “Retweet or not? personalized tweet re-ranking,” in *Proc. Sixth ACM Int. Conf. Web Search and Data Mining*, 2013, pp. 577–586.
2. B. Jiang, Z. Lu, N. Li, J. Wu, and Z. Jiang, “Retweet prediction using social-aware probabilistic matrix factorization,” in *Comput. Sci.–ICCS 2018*. Springer, 2018, pp. 316–327.
3. B. Jiang, Z. Lu, N. Li, J. Wu, F. Yi, and D. Han, “Retweeting prediction using matrix factorization with binomial distribution and contextual information,” in *Database Syst. for Adv. Appl.: DASFAA 2019*. Springer, 2019, pp. 121–138.
4. T.-A. Hoang and E.-P. Lim, “Microblogging content propagation mod- eling using topic-specific behavioral factors,” *IEEE Trans. Knowl. and Data Eng.*, vol. 28, no. 9, pp. 2407–2422, 2016.
5. J. Wang, V. W. Zheng, Z. Liu, and K. C.-C. Chang, “Topological recurrent neural network for diffusion prediction,” in *2017 IEEE Int. Conf. Data Mining (ICDM)*, 2017, pp. 475–484.
6. M. R. Islam, S. Muthiah, B. Adhikari, B. A. Prakash, and N. Ramakrish- nan, “Deepdiffuse: Predicting the’who’and’when’in cascades,” in *2018 IEEE Int. Conf. Data Mining (ICDM)*, 2018, pp. 1055–1060.
7. Z. Cao, K. Han, and J. Zhu, “Information diffusion prediction via dynamic graph neural networks,” in *2021 IEEE 24th Int. Conf. Comput. Supported Cooperative Work in Design (CSCWD)*, 2021, pp. 1099–1104.
8. C. Yuan, J. Li, W. Zhou, Y. Lu, X. Zhang, and S. Hu, “Dyhgcn: A dynamic heterogeneous graph convolutional network to learn users’ dynamic preferences for information diffusion prediction,” in *Mach. Learn. and Knowl. Discovery in Databases: Eur. Conf., ECML PKDD 2020*. Springer, 2021, pp. 347–363.
9. C. Yang, H. Wang, J. Tang, C. Shi, M. Sun, G. Cui, and Z. Liu, “Full-scale information diffusion prediction with reinforced recurrent networks,” *IEEE Trans. Neural Networks and Learn. Syst.*, vol. 34, no. 5,

pp. 2271–2283, 2021.

1. C. Li, J. Ma, X. Guo, and Q. Mei, “Deepcas: An end-to-end predictor of information cascades,” in *Proc. 26th Int. Conf. World Wide Web*, 2017,

pp. 577–586.

1. X. Xu, F. Zhou, K. Zhang, S. Liu, and G. Trajcevski, “Casflow: Exploring hierarchical structures and propagation uncertainty for cascade prediction,” *IEEE Trans. Knowl. and Data Eng.*, vol. 35, no. 4, pp. 3484–

3499, 2021.

1. Y. Tian, R. Fan, X. Ding, X. Zhang, and T. Gan, “Predicting rumor retweeting behavior of social media users in public emergencies,” *IEEE Access*, vol. 8, pp. 87 121–87 132, 2020.
2. H. Yin, S. Yang, X. Song, W. Liu, and J. Li, “Deep fusion of multimodal features for social media retweet time prediction,” *World Wide Web*, vol. 24, no. 4, pp. 1027–1044, 2021.
3. J. Wang and Y. Yang, “Tweet retweet prediction based on deep multitask learning,” *Neural Process. Lett.*, pp. 1–14, 2022.
4. R. Ma, X. Hu, Q. Zhang, X. Huang, and Y.-G. Jiang, “Hot topic-aware retweet prediction with masked self-attentive model,” in *Proc. 42nd Int. ACM SIGIR Conf. Res. and Develop. in Inf. retrieval*, 2019, pp. 525–534.
5. L. He, C. Shen, A. Mukherjee, S. Vucetic, and E. Dragut, “Cannot predict comment volume of a news article before (a few) users read it,” in *Proc. Int. AAAI Conf. Web and Social Media*, vol. 15, 2021, pp. 173–184.
6. M. Arjovsky, “Out of distribution generalization in machine learning,” Ph.D. dissertation, New York University, 2020.
7. J. Liu, Z. Shen, Y. He, X. Zhang, R. Xu, H. Yu, and P. Cui, “Towards out-of-distribution generalization: A survey,” *arXiv preprint arXiv:2108.13624*, 2021.
8. S. Wang, S. Lightman, and N. Cristianini, “Effect of the lockdown on diurnal patterns of emotion expression in twitter,” *Chronobiology Int.*, vol. 38, no. 11, pp. 1591–1610, 2021.
9. Y. Zhao, C. Wang, C.-H. Chi, K.-Y. Lam, and S. Wang, “A comparative study of transactional and semantic approaches for predicting cascades on twitter,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 1212– 1218.
10. Y. Quan, Y. Jia, B. Zhou, W. Han, and S. Li, “Repost prediction incorporating time-sensitive mutual influence in social networks,” *J. of Comput. Sci.*, vol. 28, pp. 217–227, 2018.
11. E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, “Everyone’s an influencer: quantifying influence on twitter,” in *Proc. Fourth ACM Int. Conf. Web Search and Data Mining*, 2011, pp. 65–74.
12. T. Martin, J. M. Hofman, A. Sharma, A. Anderson, and D. J. Watts, “Exploring limits to prediction in complex social systems,” in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 683–694.
13. L. Hong, O. Dan, and B. D. Davison, “Predicting popular messages in twitter,” in *Proc. 20th Int. Conf. Companion World Wide Web*, 2011, pp. 57–58.
14. E. Khabiri, C.-F. Hsu, and J. Caverlee, “Analyzing and predicting community preference of socially generated metadata: A case study on comments in the digg community,” in *Proc. Int. AAAI Conf. Web and Social Media*, vol. 3, no. 1, 2009, pp. 238–241.
15. B. Zhou, S. Pei, L. Muchnik, X. Meng, X. Xu, A. Sela, S. Havlin, and

H. E. Stanley, “Realistic modelling of information spread using peer-to- peer diffusion patterns,” *Nature Human Behaviour*, vol. 4, no. 11, pp. 1198–1207, 2020.

1. B. A. Huberman, D. M. Romero, and F. Wu, “Social networks that matter: Twitter under the microscope,” *arXiv preprint arXiv:0812.1045*, 2008.
2. D. Antypas, A. Ushio, J. Camacho-Collados, V. Silva, L. Neves, and

F. Barbieri, “Twitter topic classification,” in *Proc. 29th Int. Conf. Comput. Linguistics*, 2022, pp. 3386–3400.

1. F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves, “Tweeteval: Unified benchmark and comparative evaluation for tweet classification,” in *Findings Assoc. for Comput. Linguistics: EMNLP 2020*, 2020, pp. 1644–1650.
2. C. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in *Proc. Int. AAAI Conf. Web and Social Media*, vol. 8, no. 1, 2014, pp. 216–225.
3. J. M. Pe´rez, J. C. Giudici, and F. Luque, “pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks,” *arXiv preprint arXiv:2106.09462*, 2021.
4. M. Mousavi, H. Davulcu, M. Ahmadi, R. Axelrod, R. Davis, and

S. Atran, “Effective messaging on social media: What makes online content go viral?” in *Proc. ACM Web Conf.*, 2022, pp. 2957–2966.