Appendix

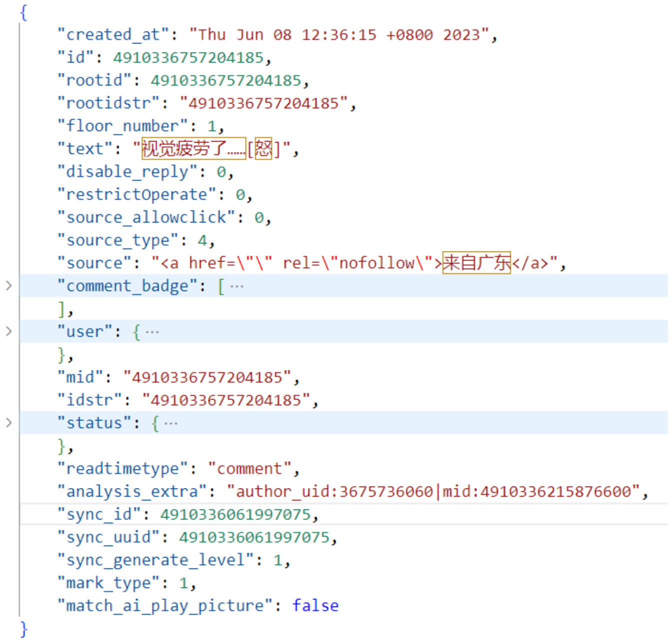
Appendix A
Supplement for Reproducibility

Dataset: Two TWITTER datasets (i.e., Individual-Behavior [1] and Multi-Behavior [2]) are already publicly available. Therefore, to maximize the availability of supplementary materials, the two original datasets are no longer uploaded. This work presents two WEIBO datasets (i.e., Individual-Behavior 14GB and Multi-Behavior 150GB). However, the original dataset consumes more than 164GB of memory, which exceeds the maximum value for supplementary materials on the TMM submission website. Therefore, we only uploaded a partial sample of the original dataset, i.e., three items of original data for each type. The original data were all obtained using the official Weibo API[1]. Therefore, they contain a large amount of data that is irrelevant to the current study (the specific acquired original data are shown in Fig. 1). Subsequently, the graph dataset obtained from the original dataset, i.e., the version of the dataset capable of running G2I (Graph-to-Image) directly, is provided. In particular, to fulfill the upload requirement of the supplementary materials, the original and graph datasets are partially sampled. Moreover, both types of datasets are saved in the Dataset_Case folder. Finally, the full dataset Dataset_2D is provided after being structurally encoded by the G2I algorithm, including the four datasets for Twitter and Weibo. All of these datasets can be found in the supplementary materials (see Fig. 2). In the future, the unprocessed datasets will be made publicly available on GitHub. Additionally, the address of the publicly available datasets is linked in the article.

Python library	Version	Python library	Version
python	3.10.6	tensorboard	2.10.1
node2vec	0.3.0	networkx	2.8.8
transformers	4.37.2	numpy	1.23.2
keras	2.10.0	scikit-learn	1.1.2
requests	2.28.1	jieba	0.42.1

TABLE I
IMPORTANT PYTHON LIBRARY NAMES AND VERSIONS FOR THE BDSI MODEL REPLICATION AND DATASET COLLECTION PROCESS PARTS

BDSI Model Source Code: The model source code is submitted along with four datasets that can be run directly, namely ”Datasets→Dataset_2D”. The graph data from “Datasets→Dataset_Case→Graph” to “Dataset_2D” goes through two stages. Firstly, the node features are computed using the pre-trained LSTM network in the ”Code→Datasets→emotion” folder. The three emotion feature values in the last layer of LSTM are considered as the RGB of the corresponding pixel point of the node. Subsequently, the graph structure is encoded using ”structure_encoding.py” in the ”Code→Datasets→image_like” folder. After that, the graph data will be converted to the form in the “Dataset_2D” folder.



```
{
    "created_at": "Thu Jun 08 12:36:15 +0800 2023",
    "id": 4910336757204185,
    "rootid": 4910336757204185,
    "rootidstr": "4910336757204185",
    "floor_number": 1,
    "text": "视觉疲劳了......[怒]",
    "disable_reply": 0,
    "restrictOperate": 0,
    "source_allowclick": 0,
    "source_type": 4,
    "source": "<a href=\"\" rel=\"nofollow\">来自广东</a>",
>   "comment_badge": [ ···
    ],
>   "user": { ···
    },
    "mid": "4910336757204185",
    "idstr": "4910336757204185",
>   "status": { ···
    },
    "readtimetype": "comment",
    "analysis_extra": "author_uid:3675736060|mid:4910336215876600",
    "sync_id": 4910336061997075,
    "sync_uuid": 4910336061997075,
    "sync_generate_level": 1,
    "mark_type": 1,
    "match_ai_play_picture": false
}
```

Fig. 1. The original data obtained by the Weibo API. The figure shows a single comment data. In particular, ”comment_badge”, ”user”, and ”status” are secondary dictionaries. For detailed API return information, please check the original dataset part sample in the Dataset_Case folder of the supplementary materials.

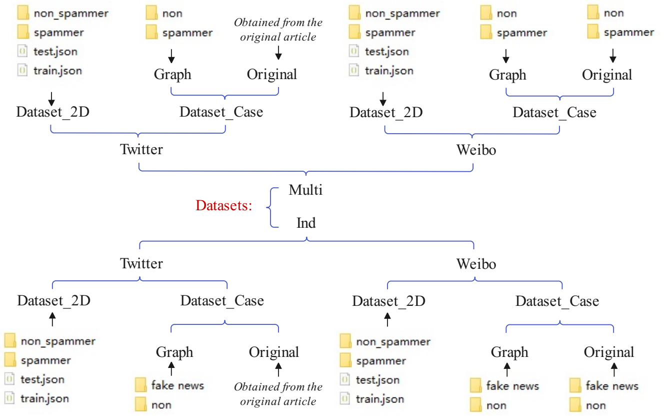


Fig. 2. Four dataset indexes in the Supplementary Materials

Subsequently, run the ”run.py” program in the ”Code→ind_detection” and ”Code→multi_detection” folders to run the model training and prediction algorithms. config.py” is the hyper-parameter control file for the model training or testing process. datasets library is the data processing library, and util library contains the tool algorithms for model prediction or training. The versions of Python libraries used for dataset construction and modeling code are shown in

[1] https://open.weibo.com/wiki/

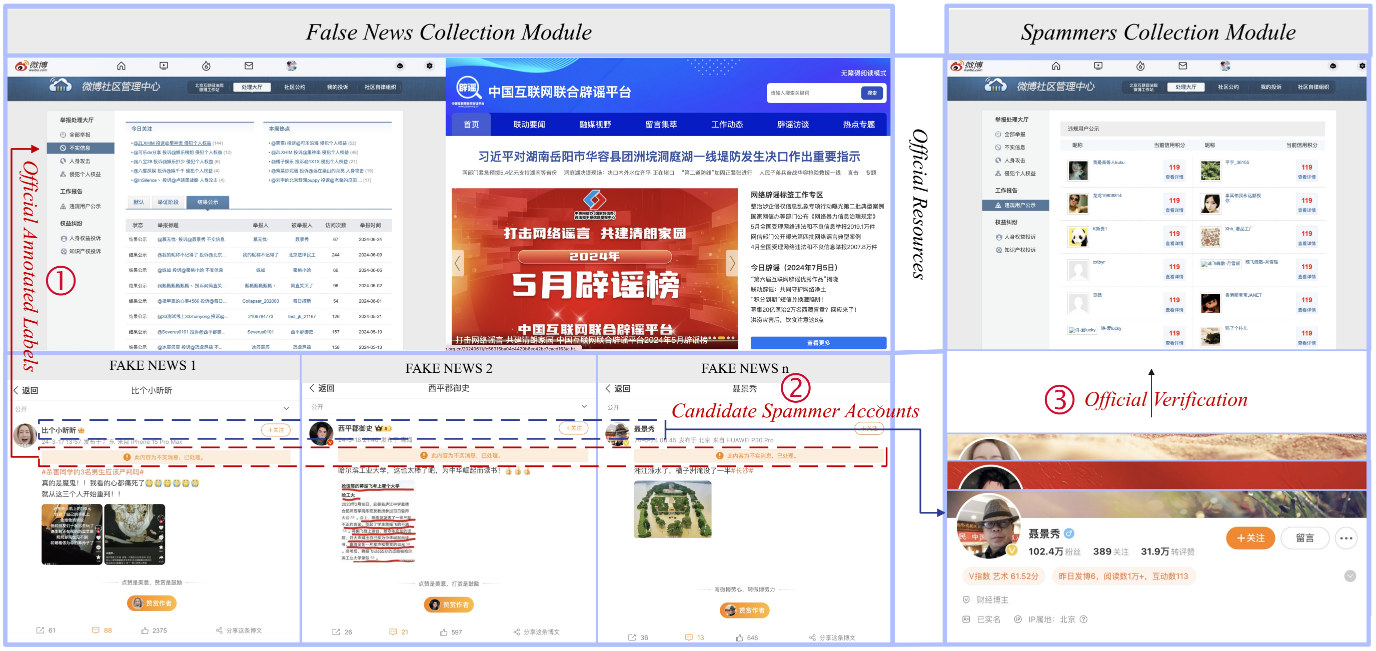


Fig. 3. Collection process for spammer accounts

Source	Description	URL
Weibo Community Management Center	The official display website for WEIBO. Displayed content is officially reviewed fake news, user dispute judgements and spammer accounts.	https://service.account.weibo.com/
Spammers Display Platform	The official website of WEIBO for displaying spammers' accounts. All displayed accounts are determined based on user submissions and official manual review.	https://service.account.weibo.com/toppunish
Spam Display Platform	The official website of WEIBO for displaying spam (fake news).	https://service.account.weibo.com/?type=5\&status=4
China Fact Check	A platform to fact-check Chinese international news.	https://chinafactcheck.com/
China Joint Internet Rumor-Busting Platform	A joint rumor (fake news) display platform launched by the Chinese government. It contains fake news from all social media platforms in China.	https://www.piyao.org.cn/

TABLE II
RESOURCES USED IN THE COLLECTION PROCESS

Table I. In the future, we will likewise make the source code publicly available on the GitHub website. In addition, the address of the public source code is linked in the article.

APPENDIX B
DATASETS CONSTRUCTION

Collection: The TWITTER dataset (i.e., Individual-Behavior [1] and Multi-Behavior [2]) uses publicly available datasets. Subsequently, the WEIBO dataset is constructed. The data collection process is shown in Fig. 3.

Spammer Account Collection: In contrast to the TWITTER dataset, WEIBO officially identifies accounts as spammers by restricting access to other users and deleting accounts. The official measures are taken to prevent spammers from continuing to cause harm. Therefore, it is not possible to collect users from the official WEIBO spammer account display platform [3]. However, as WEIBO is a mainstream social platform in China, it is important to identify spammers. To continue the research, we used other strategies to collect spammers' information. Because spammers often send fake news, information about users who send fake news is collected in advance. Subsequently, the candidate users are compared with the WEIBO spammer display platform to construct a spammer dataset. The specific process contains three main steps:

1. *Collecting fake news:* The Chinese government has constructed a rumor display platform to reduce the impact of rumors (fake news) (see Table II Button). The platform contains information on rumor topics across platforms, including WEIBO, TikTok (Douyin) and Xiaohongshu, etc. Meanwhile, WEIBO has also built its official rumor (fake news) display platform. Therefore, a fake news dataset (ind-Weibo) is collected based on the official platform (see Fig.3 Left).
2. *Determine the order of spammer candidate accounts:* The fake news dataset contains a large number of candidate accounts. When a user is identified as a spammer by WEIBO officials, the officials take action immediately. Meanwhile, the user's historical behavior information contains a large amount of data. For instance,

a user with the nickname ”Zhuge Lao Tiezhu” has more than 20,000 historical behaviors. In addition, the average number of user behaviors in the spreading space is more than 370,000, of which 199,753 are comments and 174,994 are retweets. The user’s data consumes more than 470 MB. Therefore, it is necessary to determine the user’s level of importance to target the best candidates more quickly. Considering that the more users send fake news, the potential to become spammers is higher. Therefore, we prioritize the collection of user history behavior information by weighting the number of sent fake news (see Fig. 3 Left).

3. *Identify the spammer account:* Compare candidate accounts with the Weibo spammer display platform (see Table II Top) to determine the user is a spammer (see Fig. 3 Right). Simultaneously, information about spammers’ historical behavior is put into our publicly available Weibo dataset (multi-Weibo). Due to the high real-time nature of this task, we have only correctly collected 342 items of data after one year. In the future, we will continue the data collection.

***Normal Account Collection*:** Firstly, the account addresses are collected randomly. Subsequently, comparisons with the spammer display platform and the fake news dataset filtered for non-existent accounts. After that, normal accounts are screened using manual review. In particular, we collected a large number of normal accounts. Because TWITTER is a typical unbalanced dataset, our initial intention is to construct a balanced dataset to validate the model performance. Therefore, the 343 normal accounts are randomly selected to be put into our publicly available datasets.

APPENDIX C
SUPPLEMENT FOR BASELINE ALGORITHMS

***Fake News (individual-behavior) Identification*:** To validate the individual behavior-driven BDSI models, the following baseline methods are chosen for this work. A more detailed description of the baseline algorithm replication could not be provided due to the page limitations of the manuscript. Therefore, this section provides a detailed baseline algorithm reproduction process:

- **SVM-TS** [4]: Time-series Support Vector Machine (SVM-TS) is used to model the individual-behavior-driven spreading space and identify fake news. In this case, SVM-TS uses the sklearn library for python, which provides the full-fledged model i.e. ”from sklearn.svm import SVC”. Finally, SVM classifier using Radial Basis Function (RBF) as kernel function i.e. ”SVC(kernel=’RBF’)”.
- **PPC_RNN** [5]: Combined Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) represent the spreading space. Firstly, all the response data in the spreading space are sorted based on the publication time. Secondly, each response is transformed into a word embedding vector using the Word2Vec algorithm. To ensure semantic integrity, the length of the statements is set to 29 words and the word embedding dimension is 100. Subsequently, “tensorflow.keras.layers.Conv2D()” and “tensorflow.keras. layers.MaxPooling2D()” are used to mine each response data. Finally, two layers of “tensorflow.keras.layers.SimpleRNN()” are introduced for final representation. Subsequently, two linear layers are connected to mine the hidden features and recognize the fake news.
- **TD-RvNN** [6]: A top-down recurrent neural network model. Similar to PPC_RNN, TD-RvNN essentially uses an RNN network to model the spreading space. However, there is a huge difference between the two in terms of sequence modeling. In the former, the spreading sequence is constructed based on the publication time. In the latter, recursive ideas are used to combine the structural features of the spreading part. Specifically, the TD-RvNN starts from the root node and traverses the entire spreading tree from top to bottom using a depth-first order. Subsequently, the traversed response order is corresponded to the response sequence. Finally, the authors use a Long Short-Term Memory Neural Network (LSTM) instead of a traditional RNN. Other than these, the model components are the same as PPC_RNN.
- **GAN_TRANS** [7]: A fake news recognition model based on Generative Adversarial Network (GAN) structure. In this case, the generator component uses the Transformer model to mine the spreading space. Compared with the RNN series models, the Transformer model performs better. Firstly, the Transformer model adopts the attention mechanism, so it avoids the problem that the RNN series models cannot be trained in parallel. Secondly, the attention mechanism is more advantageous in modeling long text sequences. Subsequently, the discriminator adopts the RNN model. The model training objective is to enhance the discriminator’s discriminatory ability. Finally, the discriminator is used to recognize fake news. In particular, the original article has open-sourced the model code[2]. Therefore, the model is not reproduced in this work.
- **FGCN** [8]: A GCN-based (graph convolutional neural network) fake news recognition model. The comment retweets and friend relationships in the spreading space are combined as display user-user relationships. Subsequently, implicit user-user relationships are established based on the textual similarity between response data. The graph relationship matrix of spreading space has a combination of display and implicit relationships. Moreover, the authors are inspired by fuzzy theory to construct the fuzzy module of graph edge structure, so as to enhance the generalization ability of the graph model. In the reproduction process, a graph sampling approach is used to construct the edge structure fuzzy module in the article. Finally, the response data is quantified as user node features using the pre-trained Bert model. Furthermore, two FGCN layers are used to mine graph

[2] https://github.com/majingCUHK/Rumor_GAN

structure features and two linear layers are used to predict whether the current topic is fake news.

In the task of fake news recognition, deep learning based baseline algorithms can be categorized into two types. 1) The spreading space is regarded as a time-series sequence and each response is considered as a word. Subsequently, the spreading response sequence is constructed using different curated sequences and the fake news is recognized with the help of RNN family of models, i.e., models such as PPC_RNN, TD-RvNN, GAN_TRANS, etc. 2) GNN-based fake news recognition models. This type of model is more suitable for graph data modeling. The first sequence-based modeling strategy of spreading space is not suitable for spammer identification task. The reason is as follows, spammers generate multi-behaviors for a short period of time. The multi-behavior-driven spreading space contains huge amount of response data. Moreover, RNN sequence models produce feature forgetting in terms of long text representation. Therefore, no matter any strategy to construct multi-behavior-driven spreading space response sequences can be directly represented using RNN models. Therefore, in the spammer identification task, we removed any baseline algorithms based on RNN structures. Only the graph convolutional neural network (FGCN) is used as the baseline algorithm for the new task.

***Spammer (multi-behavior) Identification*:** Due to the construction of an entirely new dataset, we reproduce the baseline algorithm. The process of reproducing the supplement is shown as follows:

- **MDGCN** [9]: The model source code has been made publicly available in the original paper[3]. Subsequently, the model is applied to the TWITTER and WEIBO datasets for performance comparison.
- **GCNwithMRF** [10]: A two-layer graph convolutional neural network (GCN) is constructed to represent the features of the user relationship graph. Subsequently, the features are represented non-linearly using the Softmax function. Finally, the MRF (Markov Random Field) layer proposed in the linked original paper is used for the final representation.

All of the above deep learning based models are trained by adding Dropout layer for parameter maximization at the time of replication. Moreover, the parameter forgetting rate is set to 0.5. Subsequently, the loss functions of all baseline models are set to the average cross-entropy loss. Meanwhile, the parameter p is introduced to adjust the weights of different samples in the batch data during the unbalanced dataset training. As a result, p is automatically calculated for each batch training, i.e., p is the complement of the few samples to the current batch data size.

References

[1] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *IJCAI*, 2016, pp. 3818–3824.

[2] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter," in *WWW*, 2012, pp. 71–80.

[3] Z. Yang, Y. Pang, Q. Li, S. Wei, R. Wang, and Y. Xiao, "A model for early rumor detection base on topic-derived domain compensation and multi-user association," *Expert Systems with Applications*, vol. 250, p. 123951, 2024.

[4] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in *CIKM*, 2015, pp. 1751–1754.

[5] Y. Liu and Y.-F. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *AAAI*, vol. 32, no. 1, 2018.

[6] J. Ma, W. Gao, S. Joty, and K.-F. Wong, "An attention-based rumor detection model with tree-structured recursive neural networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 4, pp. 1–28, 2020.

[7] J. Ma, J. Li, W. Gao, Y. Yang, and K.-F. Wong, "Improving rumor detection by promoting information campaigns with transformer-based generative adversarial learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2657–2670, 2023.

[8] L. Wei, D. Hu, W. Zhou, X. Wang, and S. Hu, "Modeling the uncertainty of information propagation for rumor detection: A neuro-fuzzy approach," *IEEE Transactions on Neural Networks and Learning Systems*, 2022. [Online]. Available: https://doi.org/10.1109/TNNLS.2022.3190348

[9] L. Deng, C. Wu, D. Lian, Y. Wu, and E. Chen, "Markov-driven graph convolutional networks for social spammer detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 310–12 322, 2023.

[10] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen, "Graph convolutional networks with markov random field reasoning for social spammer detection," in *AAAI*, 2020, pp. 1054–1061.

[3] https://github.com/dleyan/MDGCN