What Will Make Misinformation Spread: An XAI Perspective

Hongbo Bo[1,*], Yiwen Wu[2], Zinuo You[1],
Ryan McConville[2], Jun Hong[3], Weiru Liu[2]

[1] Department of Computer Science, University of Bristol, Bristol, UK
[2] Department of Engineering Mathematics, University of Bristol, Bristol, UK
[3] School of Computing and Creative Technologies, University of the West of England, Bristol, UK

Abstract. Explainable Artificial Intelligence (XAI) techniques can provide explanations of how AI systems or models make decisions, or what factors AI considers when making the decisions. Online social networks have a problem with misinformation which is known to have negative effects. In this paper, we propose to utilize XAI techniques to study what factors lead to misinformation spreading by explaining a trained graph neural network that predicts misinformation spread. However, it is difficult to achieve this with the existing XAI methods for homogeneous social networks, since the spread of misinformation is often associated with heterogeneous social networks which contain different types of nodes and relationships. This paper presents, MisInfoExplainer, an XAI pipeline for explaining the factors contributing to misinformation spread in heterogeneous social networks. Firstly, a prediction module is proposed for predicting misinformation spread by leveraging GraphSAGE with heterogeneous graph convolution. Secondly, we propose an explanation module that uses gradient-based and perturbation-based methods, to identify what makes misinformation spread by explaining the trained prediction module. Experimentally we demonstrate the superiority of MisinfoExplainer in predicting misinformation spread, and also reveal the key factors that make misinformation spread by generating a global explanation for the prediction module. Finally, we conclude that the perturbation-based approach is superior to the gradient-based approach, both in terms of qualitative analysis and quantitative measurements.

Keywords: Misinformation Spread, Graph Neural Networks, Explainable Artificial Intelligence

1 Introduction

Explainable Artificial Intelligence (XAI) [2] is a set of techniques used to make AI more explainable and understandable to humans. By using XAI techniques, developers and users of AI can understand how AI makes decisions or produces

* Corresponding author: hongbo.bo@bristol.ac.uk

outputs, including the factors considered when making the decisions. XAI has become popular because AI techniques are now prevalent in people's daily lives [30, 36], and it is important to know how AI makes decisions that can increase trust and confidence in AI systems by making AI more understandable to humans which can lead to better acceptance of and improvements in AI methods [12]. XAI methods can be divided into local explanation methods and global explanation methods. The local explanation methods [29, 20] provide the explanation for a specific decision or output of the system, while the global XAI methods [2] explain the behavior of the system as a whole.

Misinformation, which can cause negative effects, is pervasive on social media. A research question of interest to us is to understand the factors, for example, the content of the misinformation or the relationships between users, that enable the spread of misinformation on online social networks. Previous studies [33, 24] on this topic have largely cooperated with social scientists, relying on specialized knowledge for subjective analysis, which is not efficient when social media data is huge. However, a global explanation may be able to identify which factors enable misinformation spread, but this relies on an accurate underlying machine learning model. Graph Neural Networks (GNNs) have seen increasing use in many applications, including social network analysis [27, 6, 7], and have been demonstrated success at classifying misinformation on social networks [4, 21]. Several explainable approaches for GNNs have been explored, such as GNNExplainer [37], GraphLIME [16], and GraphSHAP [25].

However, these existing methods are insufficient to explain the misinformation spread. Social networks are often studied as homogenous networks between users [39, 5], but it can be argued that they are better modeled as heterogeneous networks of different types of nodes [23]. Some of these methods, such as GraphLIME [16], can only generate explanations for a homogeneous graph that contains the same types of nodes and edges. Some other explanation methods are limited to classification tasks and may not be suitable for explaining the spread of misinformation, such as PGM-Explainer [34] which is designed for node and graph classification tasks. To address the limitations of the existing XAI methods, this paper explores two research challenges. Firstly, how to train an effective graph neural network that can accurately predict the spread of misinformation on large complex heterogeneous social networks. Secondly, given this model, how to explain the factors contributing to misinformation spread.

To address these two challenges, this paper presents MisInfoExplainer, an XAI pipeline designed to explore the factors contributing to the spread of misinformation. The key contributions of this paper are as follows: First, we provide a new formulation of the spread of misinformation problem where the objective is to predict the spread value of each source of misinformation quantitatively. Second, we introduce a misinformation spread prediction approach that leverages the GraphSAGE model with the heterogeneous graph convolution (HeteroGraphConv) to accurately predict the spread of misinformation on heterogeneous social networks. Third, we propose a GNN-based explanation approach that uses both gradient-based and perturbation-based methods to identify what

node feature types and edge types contribute to the spread of misinformation. Furthermore, we apply MisInfoExplainer to a large social network dataset to demonstrate how it can be used to identify the node feature types and edge types that contribute to the spread of misinformation. Finally, we conclude that the explanations generated by the perturbation-based approach are superior to those produced by the gradient-based approach by conducting both qualitative analysis and quantitative measurements.

2 Related Work

Our study closely relates to two distinct topics of interest. The first topic centers around the analysis of misinformation spread, aiming to gain insights into its dynamics and effects. The second topic explores the domain of GNN-based Explainable AI (XAI), with a focus on interpreting and providing transparent insights into the decision-making process of Graph Neural Networks.

2.1 Misinformation Spread

Misinformation is false or inaccurate information by concealing the correct facts, also called 'fake news' or 'rumor'. Misinformation has the potential to spread rapidly through social media due to users' behaviors, leading to various negative effects. Consequently, the detection of misinformation has emerged as an important research topic. One category of studies involves using Natural Language Processing (NLP) technology to determine whether a post contains misinformation [17, 9] and the explanations are also involved during detection, such as dEFEND [31] which is to capture the features from the comments on a message to explain why a message is considered as fake. Other studies have used information propagation models for graph structures or GNNs to detect the spread of false information [4, 21].

Our study, however, focuses on the spread of known misinformation rather than whether a message is misinformation. Some research studies the spread of misinformation by using propagation models [33, 22], while few have used GNN models. However, the spread of known misinformation can be framed as an information propagation problem and GNNs are currently the most commonly used approach for modeling the relationships between users in information spread prediction models for social networks. Examples of such models include CasCN [11], MUCas [10], and coupledGNN [8], which all focus on homogeneous graphs rather than heterogeneous graphs.

There are also studies that aim at explaining the misinformation spread. For instance, [33] examined why fake information spreads faster than true information, and [24] provided a psychological framework for understanding the spread of misinformation. However, none of them used the XAI method to explain a prediction model. To the best of our knowledge, we are the first to explore the prediction and explanation of misinformation spread with the model-based XAI method.

2.2 GNN-based XAI

Graph Neural Networks (GNNs) have demonstrated their effectiveness in numerous graph machine learning tasks, as many real-world problems can be naturally represented as graphs [14]. The XAI approaches to explaining GNNs are broadly categorized into the following groups. Gradient-based methods leverage the input gradient, representing the rate of change of input features in a deep learning model, to quantify the importance values of the input features. Initially proposed for image explanation, these methods have been successfully extended to graphs, exemplified by techniques like Grad-CAM and Guided BP [26]. Perturbation-based methods assess the significance of input features by introducing perturbations to the inputs and observing the subsequent changes in model predictions. Several examples of perturbation-based Graph Neural Networks (GNNs) for Explainable AI (XAI) are GNNExplainer [37], GraphSHAP [25], and GraphMask [28]. Surrogate-based methods involve employing a simple surrogate model to approximate the outputs of a complex GNN model, and the feature importance in the surrogate model is utilized to explain the original model. Examples of surrogate-based Graph Neural Networks (GNNs) for Explainable AI (XAI) include GraphLIME [16] and PGM-Explainer [34]. These GNN-based XAI methods are designed for GNNs with homogeneous graphs, if the explanations are required for heterogeneous GNNs, extensions to these methods would be needed.

3 Problem Formulation

The social network with misinformation is represented as a heterogeneous graph that consists of multiple types of nodes, such as *users*, *misinformation*, *claims*, etc. and different types of relationships between nodes. For example, a user *following* another user, a user *posting* a misinformation tweet, a reply tweet *replying* to a misinformation tweet, a misinformation tweet *belonging* to a particular claim, etc. where *following*, *posting*, *replying* are edge types.

Definition 1 ***Heterogeneous Social Network.*** *A heterogeneous social network is defined as a heterogeneous graph $G = (V, E)$, consisting of a node set V and an edge set E. A heterogeneous graph is also associated with a node type mapping function $\xi : V \rightarrow R_V$ and an edge type mapping function $\psi : E \rightarrow R_E$. R_V and R_E denote the predefined sets of node types and edge types, respectively, with $|R_V| + |R_E| > 2$.*

A heterogeneous graph can also be represented as $G = (X, A)$, where $A = \{A_1, A_2, .., A_{|R_E|}\}$ is the set of adjacency matrices corresponding to the edge types R_E and $X = \{x_1, ..., x_v, ..., \}$ denotes the node feature vectors of nodes $v \in V$. A heterogeneous graph is also associated with a node feature type mapping function $\zeta : X \rightarrow R_X$, where R_X denotes the predefined set of node feature types and $|\zeta(x_v)| >= 1$. In a heterogeneous graph representing a social network, the misinformation (i.e., misinformation tweets) can be represented as a type of nodes $M \subset V$.

The first challenge this paper solves is to quantitatively predict the spread value, y_i, of each misinformation tweet, $m_i \in M$, on a social network G, which functionally depends on the number of reply tweets rp_i, the number of retweets rt_i, and the number of quote tweets qt_i for m_i:

$$y_i = log(rp_i + rt_i + qt_i + 1), \tag{1}$$

where y_i is the spread value of a source of misinformation $m_i \in M$.

Research Challenge 1 ***Misinformation Spread Prediction.*** *The objective of misinformation spread prediction is to use a learned misinformation spread prediction model ϕ to predict the spread value of a misinformation node $m_i \in M$ on a social network G. The model predicts the spread value of m_i on G which is represented as $\overline{y}_i = \phi(m_i, G)$ approximating the true spread value y_i.*

The second research challenge this paper solves is to analyze what causes a misinformation tweet to spread by explaining ϕ. The explanation focuses on the node feature types R_X and edge types R_E, specifically which node feature types in R_X and which edge types in R_E contribute to the misinformation spread.

Research Challenge 2 ***Misinformation Spread Explanation*** *Given the social network G and the trained misinformation spread prediction model ϕ, the objective of the misinformation spread explanation is to calculate a set of important values $Im_i \in [0, 1]$ for $i = 1, ..., |R_X| + |R_E|$ with each Im_i representing the contribution of an $Input_i \in \{R_X \cup R_E\}$, which is an input node feature or edge type to ϕ.*

4 Methodolodgy

In this section, we describe MinInfoExplainer, our proposed GNN-based explanation pipeline for predicting and explaining the spread of misinformation on social networks. The pipeline begins with training a misinformation spread prediction model ϕ to solve the problem of misinformation spread prediction (Research Challenge 1) using a heterogenous convolutional graph neural network (see Section 4.1). Then two XAI methods, a gradient-based method and a perturbation-based method, are used to explain the misinformation spread (Research Challenge 2), which is predicted by the model ϕ (see Section 4.2).

4.1 Misinformation Spread Prediction Module

We have implemented an extended version of GraphSAGE [13] to solve the misinformation spread prediction in Research Challenge 1, which is to predict the spread values $\overline{y}_i$ of the misinformation node m_i, which approximates the corresponding ground truth y_i. GraphSAGE is a GNN for node representation

learning by aggregating information from each node's neighborhood. For a homogeneous graph, a GraphSAGE layer updates the hidden representation for each node v based on the features of its neighbors $\mathcal{N}(v)$:

$$h_{\mathcal{N}(v)}^{(l+1)} = aggregate(\{h_u^l, \forall u \in \mathcal{N}(v)\}), \tag{2}$$

$$h_v^{(l+1)} = \sigma(W \cdot concat(h_v^l, h_{\mathcal{N}(v)}^{(l+1)})), \tag{3}$$

where l represents the l-th layer and W is the weight matrix. When $l = 0$, we have the $h_v^0 = x_v$, where $x_v \in X$ representing the features of v. The *aggregate* process in Eq. 2 determines how to combine the representations of v's neighbors and we use the LSTM (Long Short-Term Memory) [15] function as the *aggregate* function. Then the aggregated representation of $\mathcal{N}(v)$ and the representation of v are concatenated to generate a new representation for v (as shown in Eq. 3).

However, when the social network G used to predict the misinformation spread is heterogeneous, hence the different types of nodes and edges need to be taken into consideration. Each node is connected to its neighbor nodes by different types of edges and a heterogeneous graph convolution (HeteroGraphConv) provided by the Deep Graph Library [35] is used to initiate the GraphSAGE layer for each edge type $r \in R_E$. The different GraphSAGE layers in the same HeteroGraphConv module do not share the parameters and the HeteroGraphConv module passes the message from a source node to a target node based on the GraphSAGE layer given for the corresponding edge type. HeteroGraphConv updates the hidden representations for the nodes that are connected by the same type of edges and then a function *conv_agg* aggregates the representations for each node v that is connected by the different types of edges:

$$h_{\mathcal{N}_r(v)}^{(l+1)} = aggregate(\{h_u^l, \forall u \in \mathcal{N}_r(v)\}), \tag{4}$$

$$h_{v,r}^{(l+1)} = \sigma(W_r \cdot concat(h_v^l, h_{\mathcal{N}_r(v)}^{(l+1)})), \tag{5}$$

$$h_v^{(l+1)} = conv\_agg(\sum_{r \in R_E} h_{v,r}^{(l+1)}), \tag{6}$$

where Eq. 4 and 5 are the GraphSAGE layer for the the edge type $r \in R_E$ and $\mathcal{N}_r(v)$ represents the set of neighbors of node v with edge type r. We use a sum function as the *conv_agg* function in this work.

The entire prediction module is called HeteroGraphSAGE which outputs the prediction on the spread value, $\overline{y}_i = \phi(m_i, G), \overline{y}_i \in \overline{Y}$, for each misinformation node $m_i \in M$, with the MSE loss between $\overline{y}_i$ and y_i calculated as the feedback for the optimisation process. The prediction module is formally described in Algorithm 1.

Algorithm 1 HeteroGraphSAGE

Input: Social network G; Misinformation Nodes M; Spread Values Y
Output: The trained ϕ for predicting the spread values of M.

```
Initial φ;
while Training do
    for Each HeteroGraphConv layer in φ do
        for Each relation type in R_E do
            Initiate a GraphSAGE layer;
            Calculate the hidden representation for each node based on Eq. 4 and 5;
        end for
        Aggregate multiple relations to nodes by conv_agg (Eq. 6);
    end for
    Update weights in φ based on the loss between Ȳ and Y.
end while
```

4.2 GNN-based Explanation Module

With the prediction model ϕ trained, we propose a GNN-based explanation module that incorporates treating both node feature types R_X and edge types R_E together as the input to the model to identify the factors that contribute to the prediction on the spread of misinformation by the model. Gradient-based and perturbation-based methods are the two most common methods for explaining deep learning models. We extend these two methods to heterogeneous GNNs to explain the prediction model ϕ. Gradient-based methods use the gradients of the inputs in the deep learning model to measure the importance of the inputs, while perturbation-based methods perturb the inputs to measure the importance of the inputs. Both gradient-based and perturbation-based methods can output the importance values $Im_i \in [0, 1]$ that represents the contribution of the input feature or edge type $Input_i \in \{R_X \cup R_E\}$ to the model ϕ.

Gradient-based Method We use a widely used gradient-based attribution method, called Integrated Gradient (IG) [32], to help us understand which features are more important in making predictions. As we need to explain a heterogeneous graph model with different types of node features and edges, the IG method needs to be extended to compute the importance value of each node feature type and edge type. Given a trained model ϕ and the node feature set X, IG takes as input k different versions of the modified $\{\hat{X}_1, .., \hat{X}_k\}$ which only modified the node features values of the type that needs to be calculated. For each type of node feature, IG calculates the change in the output of the model as each feature $x^i \in X$ in the input is gradually changed. Then IG output the attribution score for each x^i by integrating the gradients of the model output with respect to x^i:

$$IG_i = (x^i - \hat{x}^i) \sum_{j=1}^{k} \left(\frac{\partial \phi(M, (A, \hat{X}_j + j/k(X - \hat{X}_j)))}{\partial x^i}\right) \tag{7}$$

where $\hat{X}_j + j/k(X - \hat{X}_j)$ is the combined modified node feature input and $\partial\phi(M, (A, \hat{X}_j + j/k(X - \hat{X}_j)))/\partial x^i$ is the gradient of output with respect to feature x^i, where M is the misinformation nodes set and A is the adjcency matrices set.

The explanation of edge types is based on a general principle of GraphSAGE, that training a model without edge weights is equivalent to training the model with all edge weights $w_e = 1, w_e \in W_e$ equal to 1, which is $\phi(M, (A, X, W_e))$. For each edge type, we first need to generate an edge weight vector w_e with values set to 1 for each type of edge and then use a similar equation to calculate the IG value for each edge type:

$$IG_e = (w_e - \hat{w_e}) \sum_{j=1}^{k} \left(\frac{\partial\phi(M, (A, X, \hat{w}_{ej} + j/k(w_e - \hat{w}_{ej}))}{\partial w_e}\right) \tag{8}$$

Since the explanation of misinformation spread in our proposed pipeline needs to be meaningful, we integrate the IG_i and IG_e absolute values into Im_i which corresponds to $Input_i$. This is done by mapping the node features corresponding to IG_i to the node feature types R_X and the edges corresponding to IG_e to edge types R_E, using the mapping functions ζ and ψ respectively. After this integration, Im_i is normalized so that $\sum_{i=1}^{N}(\mathrm{Im_i}) = 1$.

Perturbation-based Method We use a similar idea in GNNExplainer [37], to iteratively mask the node features and edges to identify the impact on the output of a GNN model. Given the trained ϕ, we use the node feature mask $Xm \in [0, 1]$ and edge mask $Am \in [0, 1]$ to perturb the node feature X and the set of adjacency matrix A, by $\hat{X} = X \odot Xm$ and $\hat{A} = A \odot Am$, where $\odot$ denotes element-wise multiplication. The intuition is that if a node feature or edge is not important to the model ϕ (with a low Im_i), even with a large perturbation (with small values in the masks), the model output $\hat{Y} = \phi(M, (\hat{A}, \hat{X})$ will not change much from the original output $\overline{Y} = \phi(M, (A, X))$. We want to obtain Xm and Am that can perturb the unimportant node feature or edge as much as possible that makes little change to the model output, then the elements in Xm and Am can indicate the importance of the node feature or edge types $Input_i$, based on the mapping functions ζ or ψ.

To generate an explanation module, the Xm and Am are trained by optimizing the following objective function:

$$\mathcal{L}_{all} = \mathcal{L}(\hat{Y}, \overline{Y}) + \alpha_1||Xm||_1 + \beta_1||Am||_1 + \alpha_2 H(Xm) + \beta_2 H(Am), \tag{9}$$

where $\mathcal{L}(\hat{Y}, \overline{Y})$ is to calculate the MSE loss of output changing after perturbation, $||Xm||_1$ and $||Am||_1$ is to make as many elements in two masks change as possible, $H(\cdot)$ is the entropy function which can make the masks as stable as possible, and α_1, α_2, β_1, β_2 are hyper-parameters.

For each node feature type or edge type, the Im_i is integrated using Xm and Am, which is the same operation used in the gradient-based method for integrating IG_i and IG_e.

5 Experimental Results

This section shows the experimental results of predicting misinformation spread and exploring factors contributing to the spread using our proposed MisinfoExplainer on a misinformation-labeled social network dataset. We also perform the evaluation of the two proposed XAI methods described in the previous section on this dataset.

5.1 Dataset

We perform our experiments on a large-scale misinformation social network dataset, MuMiN [23], to quantitatively evaluate the proposed MisinfoExplainer. The MuMiN dataset is a public misinformation graph dataset with three different versions that contain multimodal information from Twitter. Specifically, MuMiN associates multitopic and multilingual tweets with fact-checked claims, and it also includes textual and visual content from tweets. We only keep the data that are fact-checked tweets discussing misinformation and filter out the tweets discussing factual claims. The statistics of the different node types R_V in the MuMiN dataset after filtering are shown in Table 1. The data we use contains 9 types of node features, denoted as 'n1' to 'n9' in Table 2, which consist of the node feature type set R_X and 12 different types of edges, denoted 'e1' to 'e12' shown in Figure 1, which are the edge type set R_E. In our experiment, we predict the misinformation spread which is to predict the spread value $\overline{Y}$ of the *misinformation* type of nodes, and reveal the key factors that make misinformation spread which is to measure the importance values Im_i for each $Input_i \in \{R_X \cup R_E\}$.

Table 1. Three versions of the dataset. The 6 node types in R_V and the numbers of nodes in these node types are shown in the table. Misinformation is a type of nodes representing tweets that have been labelled as discussing a non-factual claim, a claim is a short description of the misinformation provided by a fact-checker and a reply is a tweet that replies to a tweet.

Dataset	Misinformation	Claim	User	Hashtag	Image	Reply
MuMiN-Small	3,589	2,049	140,113	25,472	986	163,113
MuMiN-Medium	9,326	5,318	290,199	49,575	2,397	356947
MuMiN-Large	22,835	12,509	564,789	85,501	6,309	754,097

[4] The claim_reviewer is the URL for the fact-checking website that reviewed the claim and the 'lang' is an abbreviation of 'language'.

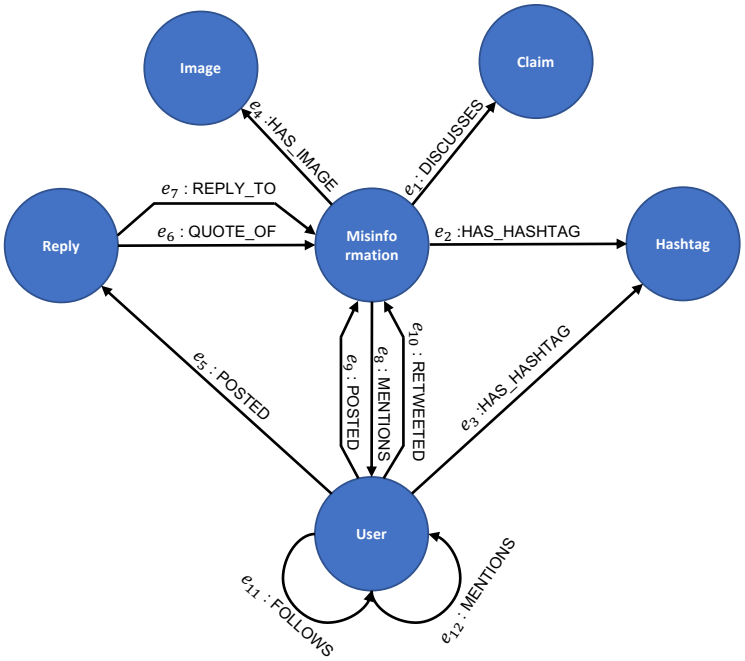

Fig. 1. The edge types R_E present in the data we use, denoted as e_1 to e_{12}. The figure shows a metagraph that consists of the nodes representing all 6 different node types in the dataset and all the edge types between them.

5.2 Prediction Module Evaluation

To comprehensively evaluate the performance of our prediction module, HeteroGraphSAGE, we conducted a series of comparative experiments on the MuMiN dataset. These experiments allow us to assess the effectiveness and efficiency of HeteroGraphSAGE in comparison to other state-of-the-art methods, providing valuable insights into its capabilities for handling heterogeneous graph data.

Experiment Setup This experimental evaluation aims to measure the effectiveness of HeteroGraphSAGE. We selected two GNNs that are commonly used in the field of social network analysis, Graph Convolutional Networks (GCN) and Graph Attention Networks (GAT), as the baseline methods. Since both GCN and GAT are designed for homogeneous graphs, we extended them to HeteroGCN and HeteroGAT, respectively, by applying HeteroGraphConv. The performance is evaluated in terms of Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and R-squared (R2).

The baseline methods and our proposed method are all based on 2-layer HeteroGraphConv and the dimension of each layer is set to 512. All parameters are trained using the AdmaW [19] optimizer with a learning rate $3e^{-4}$ and a

Table 2. The node feature types[4] R_X and node types.

Input	Features Types	Associated Node Types
n1	misinformation_text	Misinformation
n2	misinformation_lang	Misinformation
n3	claim_embedding	Claim
n4	claim_reviewer	Claim
n5	image_embedding	Image
n6	hashtag_embedding	Hashtag
n7	user_profile	User
n8	reply_text	Reply
n9	reply_lang	Reply

dropout rate 0.2. For HeteroGAT, each layer contains 3 attention heads. We used the pre-set train/valid/test splits provided by the MuMiN dataset, which claims that these pre-set splits can better cover distinct events [23] and thus better measure the ability of the model to generalise to unseen misinformation topics. The number of training epochs is set to 100.

Comparison Results The results of the experiments are shown in Table 3. HeteroGraphSAGE has significant advantages for misinformation spread prediction tasks on all three versions MuMiN dataset. The quality of our proposed regression model was assessed using three metrics, with the best performance on MAPE and MSE indicating the accurate prediction of the misinformation spread values y_i, and the best performance on R2 showing the good fit of the data.

Table 3. Performance of the prediction module based on different GNN models. For the MSE and MAPE evaluation metrics, a smaller value indicates better performance, whereas, for R2, a larger value indicates better performance.

Data	Model	MAPE	MSE	R2
MuMiN-Small	HeteroGCN	0.1752	0.5412	0.7684
	HeteroGAT	0.1660	0.5558	0.7622
	HeteroGraphSAGE	**0.1511**	**0.4214**	**0.8197**
MuMiN-Medium	HeteroGCN	0.1351	0.3213	0.8241
	HeteroGAT	0.1436	0.4000	0.7810
	HeteroGraphSAGE	**0.1239**	**0.3091**	**0.8308**
MuMiN-Large	HeteroGCN	0.1321	0.2692	0.8372
	HeteroGAT	0.1308	0.2792	0.8312
	HeteroGraphSAGE	**0.1134**	**0.2091**	**0.8735**

5.3 What Factors Make Misinformation Spread?

We then trained the HeteroGraphSAGE on the MuMiN-small dataset to obtain the trained model ϕ and then explained ϕ by using our gradient-based and perturbation-based XAI methods respectively. We considered the 9 types of node features and 12 types of edges as shown in Table 2 and Figure 1 as the $Inputs$ which are the factors we aim to measure the Im.

Experiment Setup The HeteroGraphSAGE was trained with the same settings as in the previous experiments in Subsection 5.2. For the gradient-based method, the number of modified inputs k is set to 50. For the perturbation-based method, the number of training epochs is set to 100, the learning rate is set to 0.1. The purpose of hyper-parameters in Eq. 9 is to make the terms of the loss function balance during optimizing, and we set α_1, α_2, β_1 and β_2 to 0.05, 1.0, 1.0 and 0.1.

Experiment Results and Qualitative Analysis The explanation results using perturbation-based and gradient-based methods are shown in Figure 2. Both methods consider the text of the misinformation ($n1$: *misinformation_text*) to be the most important factor in the spread of misinformation, which is also corroborated by marketing research, for example [3], which claims the message content itself can contribute to the virality.

The perturbation-based explanation considers that the four important factors after the text of the tweet are the text of reply ($n8$: *reply_text*), the embedding of the claim ($n3$: *claim_embedding*), the embedding of image ($n5$: *image_embedding*) and the users description ($n7$: *user_profile*). The reply text can include other users' opinions, stimulating engagement, which can amplify the original tweet and then contribute to spreading further, engaging more users. The claim is a short description of the misinformation, which can be seen as a summary of the misinformation. The explanation considers that images can help the spread of related misinformation, where a similar conclusion is also found in marketing research [18] that high-quality images can lead to engagement with related Tweets.

The gradient-based explanation considers four different types of edges as important factors for spreading misinformation: a user follows another user ($e11$: User *follows* User), a user retweeted misinformation ($e10$: User *retweeted* Misinformation), a user has a hashtag ($e3$: User *has_hashtag* Hashtag), and a misinformation tweet has a hashtag ($e2$: Misinformation *has_hashtag* Hashtag). In contrast to the perturbation-based approach, the gradient-based approach gives a less plausible explanation. The following relationship and retweeting interactions are utilized in many studies [27, 1] about information diffusion, but it is difficult to explain intuitively how the hashtag relationship contributes to the spread of misinformation.

In summary, the perturbation-based method considers node features to be more important, while the gradient-based method considers edges to be more

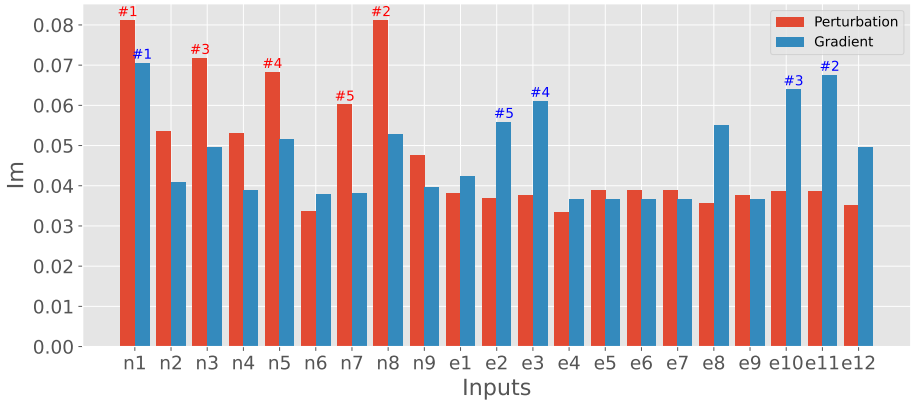

Fig. 2. *Im* values calculated by two different explanation methods. The top five *Inputs* which are considered as important in each method are marked.

important. In the following subsection, we compare the two explanation methods quantitatively to see which one is more plausible.

5.4 Which Explanation Shall We Believe?

While visualizations can provide insights regarding whether the explanations are reasonable to humans, such evaluations are not entirely trustworthy due to the lack of ground truth. In this subsection, we calculate the fidelity which can quantitatively measure the explanation methods.

The $Fidelity^+$ metric was originally proposed in [26, 38] based on the intuition that if the important factors identified by explanation methods are discriminative to the model, the predictions should change significantly when these features are removed. In this study, we extend $Fidelity^+$ to be defined as the difference between the original predictions $\phi(M, G)$ and the new predictions $\phi(M, G^{1-\sum_{i=1}^{N} Input_i})$ after masking out N important *Inputs*, as follows:

$$Fidelity^+ = \frac{1}{N}(\phi(M, G) - \phi(M, G^{1-\sum_{i=1}^{N} Input_i})), \tag{10}$$

where i is the i^{th} most important *Input* indicated by the explainer, N is the number of *Inputs* to be removed and $G^{1-\sum_{i=1}^{N} Input_i}$ indicates the graph removed N most important *Inputs*. For $Fidelity^+$, higher values indicate better explanations, and more discriminative *Inputs* are identified.

In contrast, the $Fidelity^-$ [26, 38] was proposed to study prediction change by keeping important input features and removing unimportant features. The $Fidelity^-$ is defined as the difference between the original predictions $\phi(M, G)$ and the new predictions $\phi(M, G^{\sum_{i=1}^{N} Input_i})$ where G only contains the impor-

tant *Inputs*:

$$Fidelity^{-} = \frac{1}{N}(\phi(M, G) - \phi(M, G^{\sum_{i=1}^{N} Input_i})) \tag{11}$$

For $Fidelity^{-}$, lower values indicate less important *Inputs* are removed so that the explanations results are better.

For the measurement of $Fidelity^{+}$, we conducted experiments by removing the top 1 to top 7 most important *Inputs*, with N ranging from 1 to 7. However, for the $Fidelity^{-}$ measurement, it was challenging to keep only a few *Inputs* and still construct a graph. Therefore, we set N from 12 to 18 for this measurement. The results are shown in Figure 3. We can observe that the perturbation-based approach works better, which supports the previous intuitive observation in Subsection 5.3.

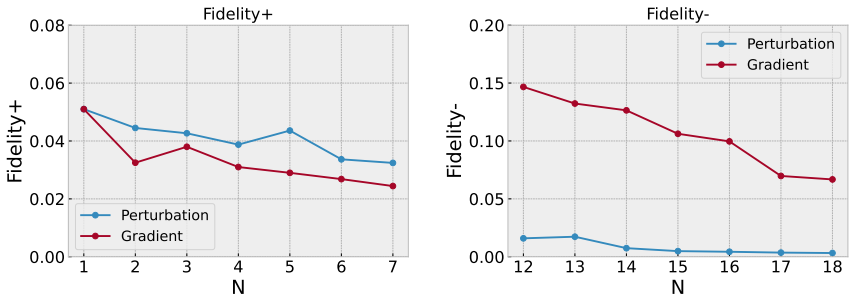

Fig. 3. Fidelity. For $Fidelity^{+}$, the higher values indicate better explanations, and for $Fidelity^{-}$, the lower values mean the explanations are better.

6 Conclusion and Future Works

In this paper, we proposed a novel XAI pipeline, called MisinfoExplainer, to explore the factors contributing to misinformation spread on social networks. The proposed MisinfoExplainer made use of the heterogenous convolutional GraphSAGE (HeteroGraphSAGE) to predict the misinformation spread with the trained model explained by XAI methods. We provided two XAI methods for explaining the trained model: a gradient-based method that exploits the gradients of the input in the model, and a perturbation-based method that perturbs the input of the model to obtain explanations. The experimental results showed that our proposed pipeline can obtain an accurate model for misinformation spread prediction, and that HeteroGraphSAGE is superior to other methods on a large-scale misinformation-labelled social network dataset. We obtained the factors that contribute to misinformation spread by explaining the prediction model using the two proposed XAI methods. Through qualitative analysis and

quantitative measurement, we concluded that the perturbation-based method provides better explanations than the gradient-based method.

Limitations and Future Work All experiments in this study are conducted under the assumption that the dataset has classified certain tweets as misinformation. Our XAI method is constrained by the model of misinformation spread, which incorporates the use of spread indicators, such as the number of retweets. In future work, we aim to develop more precise models of misinformation spread and explore advanced XAI techniques to provide comprehensive explanations for the spread process. Nonetheless, we firmly believe that the current research approach in this study, which involves modeling the spread and utilizing XAI to investigate the factors contributing to its occurrence, is a valid and valuable research direction.

References

1. Amati, G., Angelini, S., Capri, F., Gambosi, G., Rossi, G., Vocca, P.: Twitter temporal evolution analysis: Comparing event and topic driven retweet graphs. IADIS International Journal on Computer Science & Information Systems **11**(2) (2016)
2. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al.: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion **58**, 82–115 (2020)
3. Berger, J., Milkman, K.L.: What makes online content viral? Journal of marketing research **49**(2), 192–205 (2012)
4. Bian, T., Xiao, X., Xu, T., Zhao, P., Huang, W., Rong, Y., Huang, J.: Rumor detection on social media with bi-directional graph convolutional networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 549–556 (2020)
5. Bo, H., McConville, R., Hong, J., Liu, W.: Social network influence ranking via embedding network interactions for user recommendation. In: Companion Proceedings of the Web Conference 2020. pp. 379–384 (2020)
6. Bo, H., McConville, R., Hong, J., Liu, W.: Social influence prediction with train and test time augmentation for graph neural networks. In: 2021 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2021)
7. Bo, H., McConville, R., Hong, J., Liu, W.: Ego-graph replay based continual learning for misinformation engagement prediction. In: 2022 International Joint Conference on Neural Networks (IJCNN). pp. 01–08. IEEE (2022)
8. Cao, Q., Shen, H., Gao, J., Wei, B., Cheng, X.: Popularity prediction on social platforms with coupled graph neural networks. In: Proceedings of the 13th international conference on web search and data mining. pp. 70–78 (2020)
9. Chen, T., Li, X., Yin, H., Zhang, J.: Call attention to rumors: Deep attention based recurrent neural networks for early rumor detection. In: Trends and Applications in Knowledge Discovery and Data Mining: PAKDD 2018 Workshops, BDASC, BDM, ML4Cyber, PAISI, DaMEMO, Melbourne, VIC, Australia, June 3, 2018, Revised Selected Papers 22. pp. 40–52. Springer (2018)
10. Chen, X., Zhang, F., Zhou, F., Bonsangue, M.: Multi-scale graph capsule with influence attention for information cascades prediction. International Journal of Intelligent Systems **37**(3), 2584–2611 (2022)
11. Chen, X., Zhou, F., Zhang, K., Trajcevski, G., Zhong, T., Zhang, F.: Information diffusion prediction via recurrent cascades convolution. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE). pp. 770–781 (2019). https://doi.org/10.1109/ICDE.2019.00074
12. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: A survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO). pp. 0210–0215. IEEE (2018)
13. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
14. Hamilton, W.L.: Graph representation learning. Synthesis Lectures on Artifical Intelligence and Machine Learning **14**(3), 1–159 (2020)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)

16. Huang, Q., Yamada, M., Tian, Y., Singh, D., Chang, Y.: Graphlime: Local interpretable model explanations for graph neural networks. IEEE Transactions on Knowledge and Data Engineering (2022)
17. Kumar, S., Asthana, R., Upadhyay, S., Upreti, N., Akbar, M.: Fake news detection using deep learning models: A novel approach. Transactions on Emerging Telecommunications Technologies **31**(2), e3767 (2020)
18. Li, Y., Xie, Y.: Is a picture worth a thousand words? an empirical study of image content and social media engagement. Journal of Marketing Research **57**(1), 1–19 (2020)
19. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
20. Ma, H., McAreavey, K., McConville, R., Liu, W.: Explainable ai for non-experts: Energy tariff forecasting. In: 2022 27th International Conference on Automation and Computing (ICAC). pp. 1–6. IEEE (2022)
21. Monti, F., Frasca, F., Eynard, D., Mannion, D., Bronstein, M.M.: Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:1902.06673 (2019)
22. Nekovee, M., Moreno, Y., Bianconi, G., Marsili, M.: Theory of rumour spreading in complex social networks. Physica A: Statistical Mechanics and its Applications **374**(1), 457–470 (2007)
23. Nielsen, D.S., McConville, R.: Mumin: A large-scale multilingual multimodal fact-checked misinformation social network dataset. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 3141–3153 (2022)
24. Pennycook, G., Rand, D.G.: The psychology of fake news. Trends in cognitive sciences **25**(5), 388–402 (2021)
25. Perotti, A., Bajardi, P., Bonchi, F., Panisson, A.: Graphshap: Motif-based explanations for black-box graph classifiers. arXiv preprint arXiv:2202.08815 (2022)
26. Pope, P.E., Kolouri, S., Rostami, M., Martin, C.E., Hoffmann, H.: Explainability methods for graph convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10772–10781 (2019)
27. Qiu, J., Tang, J., Ma, H., Dong, Y., Wang, K., Tang, J.: Deepinf: Social influence prediction with deep learning. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2110–2119 (2018)
28. Schlichtkrull, M.S., De Cao, N., Titov, I.: Interpreting graph neural networks for nlp with differentiable edge masking. arXiv preprint arXiv:2010.00577 (2020)
29. Shi, Y., McAreavey, K., Liu, W.: Evaluating contrastive explanations for ai planning with non-experts: a smart home battery scenario. In: 2022 27th International Conference on Automation and Computing (ICAC). pp. 1–6. IEEE (2022)
30. Shi, Z., Cartlidge, J.: State dependent parallel neural hawkes process for limit order book event stream prediction and simulation. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1607–1615 (2022)
31. Shu, K., Cui, L., Wang, S., Lee, D., Liu, H.: defend: Explainable fake news detection. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 395–405 (2019)
32. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International conference on machine learning. pp. 3319–3328. PMLR (2017)

33. Vosoughi, S., Roy, D., Aral, S.: The spread of true and false news online. science **359**(6380), 1146–1151 (2018)
34. Vu, M., Thai, M.T.: Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. Advances in neural information processing systems **33**, 12225–12235 (2020)
35. Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., et al.: Deep graph library: A graph-centric, highly-performant package for graph neural networks. arXiv preprint arXiv:1909.01315 (2019)
36. Yang, X., Burghardt, T., Mirmehdi, M.: Dynamic curriculum learning for great ape detection in the wild. International Journal of Computer Vision pp. 1–19 (2023)
37. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32** (2019)
38. Yuan, H., Yu, H., Gui, S., Ji, S.: Explainability in graph neural networks: A taxonomic survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)
39. Zuo, W., Raman, A., Mondragón, R.J., Tyson, G.: Set in stone: Analysis of an immutable web3 social media platform. In: Proceedings of the ACM Web Conference 2023. pp. 1865–1874 (2023)