HyperGraphDis: Leveraging Hypergraphs for Contextual and Social-Based Disinformation Detection

Nikos Salamanos\textsuperscript{1}, Pantelitsa Leonidou\textsuperscript{1}, Nikolaos Laoutaris\textsuperscript{2}, Michael Sirivianos\textsuperscript{1}, Maria Aspri\textsuperscript{3}, Marius Paraschiv\textsuperscript{2}

\textsuperscript{1} Cyprus University of Technology, Limassol, Cyprus
\textsuperscript{2} IMDEA Networks, Madrid, Spain
\textsuperscript{3} LSTECH ESPANA SL, Barcelona, Spain

nik.salaman@cut.ac.cy, pl.leonidou@edu.cut.ac.cy, nikolaos.laoutaris@imdea.org, michael.sirivianos@cut.ac.cy, aspri@lstech.io, marius.paraschiv@imdea.org

Abstract

In light of the growing impact of disinformation on social, economic, and political landscapes, accurate and efficient identification methods are increasingly critical. This paper introduces HyperGraphDis, a novel approach for detecting disinformation on Twitter that employs a hypergraph-based representation to capture (i) the intricate social structures arising from retweet cascades, (ii) relational features among users, and (iii) semantic and topical nuances. Evaluated on four Twitter datasets – focusing on the 2016 U.S. Presidential election and the COVID-19 pandemic – HyperGraphDis outperforms existing methods in both accuracy and computational efficiency, underscoring its effectiveness and scalability for tackling the challenges posed by disinformation dissemination. HyperGraphDis displays exceptional performance on a COVID-19-related dataset, achieving an impressive F1 score (weighted) of approximately 89.5%. This result represents a notable improvement of around 6% compared to existing methods. Additionally, significant enhancements in computation time are observed for both model training and inference. In terms of model training, completion times are accelerated by a factor ranging from 2.3 to 9.3 compared to previous methods. Similarly, during inference, computation times are 1.3 to 7.2 times faster than the state-of-the-art.

Introduction

Social media platforms have evolved from mere fora for informal conversation and idea exchange into essential tools for a diverse array of entities, including non-governmental organizations, businesses, governments, and media outlets. Their ubiquity and ability to personalize content according to user behavior make them pivotal in the modern social landscape. Platforms like Twitter (currently known as X) and Facebook now serve as the principal sources of information on a worldwide scale (Westerman, Spence, and Van Der Heide 2014). Nevertheless, the dissemination of false information on these platforms can have detrimental effects on society (Chenzi 2021), politics (Safieddine 2020), and the economy (Visentin, Pizzi, and Pichierri 2019; Cheng and Chen 2020), making disinformation a rapidly escalating cyber threat (Belova and Georgieva 2018). Contemporary techniques for identifying disinformation on social media chiefly focus on partial data, either concerning user behavior or the topology of the local network (see Section Related Work). Such methods may lead to unsatisfactory results, given that the effectiveness of disinformation spread relies on a multifaceted interplay of intrinsic elements like user traits and extrinsic factors such as network features and post content.

In this study, we aim to improve disinformation detection accuracy and computational efficiency on social media by introducing an integrated approach that considers both intrinsic and extrinsic variables. For this purpose, we propose HyperGraphDis, an innovative approach utilizing a hypergraph data structure to encapsulate both the content of Twitter retweet cascades and the relational information among users. On Twitter, users create messages (tweets), and others share these messages (retweets). A posted message initiates a root-tweet (the original tweet that gets retweeted), followed by a sequence of retweets forming a tree-like graph known as a retweet cascade. The choice of a hypergraph stems from its ability to represent complex relationships more naturally than traditional graphs. Unlike standard graphs with vertices connected by edges, a hypergraph features hyperedges that can connect more than two nodes. This allows for a nuanced representation of Twitter interactions, considering both the content of tweets and retweets (Twitter cascades) and the network of users involved in these cascades.

In the initial phase of hypergraph construction, we apply a graph partitioning algorithm to the Twitter social network, where nodes represent users and edges represent social connections. This process effectively identifies user clusters with similar interaction patterns or content sharing. The essence of graph partitioning is to optimize the divisions in such a way that intra-cluster connections are maximized while inter-cluster connections are minimized, thereby identifying natural communities within the Twitter user base.

Upon identifying these user clusters, we undertake a transformation wherein each user in a cluster is substituted with a list of Twitter cascades in which they have partaken. In the hypergraph framework, each previously discovered cluster becomes a hyperedge. The Twitter cascades, which are sequences of tweets and retweets initiated by a root-tweet, are represented as the nodes within these hyperedges. This transformation inherently reshapes the problem space: it turns the complex task of disinformation classification on Twitter from an intricate, multi-variable problem into a more...
straightforward node classification problem within the hypergraph.

The use of hypergraph data structures offers several advantages over traditional graphs. Unlike conventional graph edges that connect only pairs of nodes, a hyperedge links an entire cluster of users, thereby enriching the relational information in the dataset by capturing multi-way associations. This multi-faceted relational insight becomes even more potent when considering hyperedge intersections, which naturally coincide with the overlap of user clusters. Such overlaps allow for a nuanced understanding of classes of users sharing specific common features, further augmenting the dataset’s expressiveness. Moreover, hypergraphs inherently result in less dense network representations, yielding benefits in terms of reduced computational processing times and more efficient storage requirements. This makes hypergraphs an excellent choice for the effective modeling of complex relational systems.

Moreover, we enrich each Twitter cascade node with a set of features, such as the frequency of specific keywords, sentiment analysis scores, or the presence of verified accounts (with the Twitter verification badge). By doing so, we transform the issue of classifying Twitter cascades in terms of disinformation into a node classification task within the hypergraph. For the node classification task, we employ hypergraph neural networks, a modern machine learning approach optimized for hypergraph structures.

We evaluate our approach on four datasets: (i) an extensive dataset on the 2016 U.S. Presidential Election; (ii) a substantial collection of tweets related to the COVID-19 pandemic, along with social engagement data (Li et al. 2020); and (iii) the Health Release and Health Story datasets, integral parts of the FakeHealth repository (Dai, Sun, and Wang 2020). Overall, the HyperGraphDis shows exceptional performance (Table 3). More specifically, evaluating it with MM-COVID achieves an impressive F1 score of around 89.5%. It outperforms the Meta-graph method (Paraschiv et al. 2022) by approximately 6%, and Cluster-GCN (Chiang et al. 2019) by 33%. Additionally, HyperGraphDis outperforms HGFND (Jeong et al. 2022) by 6.4% and other state-of-the-art ML text-based techniques by 10% (Table 4). Furthermore, noteworthy enhancements are observed in the computation time for both model training and inference processes. Regarding model training, the completion time is notably expedited, ranging from 2.3 times to 9.3 times faster than the second-best method (in terms of F1 score). Similarly, computational time exhibits increased efficiency during inference, ranging from 1.3 times to 7.2 times faster than the second-best method across the four datasets.

The results demonstrate that our method not only yields accuracy figures that are comparable to or exceed existing methods relying on both tweet content and user relationships, but it also does so with the advantages of reduced data size and faster classification run times. Moreover, the inherent design of our method makes it highly scalable, allowing it to adapt to varying sizes and complexities of data.

**Data availability** Part of the U.S. Elections dataset is publicly available (Salamanos et al. 2023) under proper restrictions for compliance with Twitter’s Terms of Service (Tos), General Data Protection Regulation (GDPR). The Fake-Health repository and the MM-COVID dataset are publicly available (Dai, Sun, and Wang 2020), (Li et al. 2020).
learning models like RoBERTa-large to assess the veracity of claims in medical datasets. This work complements efforts in understanding web-based relationships, where Hanley, Kumar, and Durumeric (2023) utilize Common Crawl data to analyze hyperlinks between websites. In Hanley and Durumeric (2023), the authors present a comprehensive framework to study Reddit communities, targeting misinformation, toxicity, and political polarization. Similarly, Pratelli et al. (2023) delve into the behavior of social bots on Twitter, categorizing them based on their political leanings during the 2018 U.S. midterms. Balcı, Sirivianos, and Blackburn (2023) extend this by introducing metrics, MRD and MRS, to quantify user migrations across subreddits on the Reddit social platform. Furthermore, Elmas, Overdorff, and Aberer (2023), the focus is on detecting repurposed Twitter accounts using supervised learning, shedding light on their characteristics and behaviors.

In Kozik et al. (2023), the authors conducted a meta-analysis of state-of-the-art Machine Learning techniques for text-based fake news detection, including Random Forest, CNN-based, LSTM-based, and BERT-based models. They evaluated these models on benchmark datasets, including MM-COVID. We evaluated the HyperGraphDis method using MM-COVID, consistently achieving superior performance compared to all mentioned models in terms of F1 score. The SOMPS-Net framework (Dhanasekaran et al. 2021), with the Social Interaction Graph and Publisher and News Statistics components, achieved a 72.7% accuracy and 79.6% F1 score on the Health Story dataset for fake news prediction. Evaluating HyperGraphDis on the same dataset, we observed an accuracy of 75.2% and F1 score of 72.2%. We attribute this decline in F1 score to the inherent character of the cascades within the Health Story dataset, which tend to be either extremely small or predominantly structured like star-like graphs. This characteristic leads to embedding vectors that exhibit high levels of similarity, consequently affecting the overall effectiveness of our method.

The methods most closely related to ours are Paraschiv et al. (2022) and Jeong et al. (2022). In Paraschiv et al. (2022), the Meta-Graph method captures content and relational information. Jeong et al. (2022) introduces a hypergraph-based neural network (HGFND) for detecting relational fake news with limited labeled data. They formulate fake news detection as a semi-supervised node classification task in a hypergraph, capturing group-level interactions among news articles. Our analysis reveals that HyperGraphDis outperforms these methods, providing superior results with enhanced computational efficiency.

In evaluating the performance metrics of our proposed method, several key factors underscore its superiority, particularly when compared to classical graph-based approaches. First, the inherently reduced connectivity of the hypergraph structure yields faster computational times and minimizes hardware resource requirements. This efficiency is not achieved at the expense of informational richness; to the contrary, hyperedges encapsulate a wealth of multi-way relationships among nodes, providing a more nuanced understanding of underlying data patterns. Additionally, the intersections of hyperedges serve as focal points for identifying user subsets with highly detailed common features, adding an extra layer of depth to the dataset’s relational information content.

**Similar Approaches**

**Meta-graph**

In Paraschiv et al. (2022), the authors introduce a meta-graph data structure to represent events in a social network. Formally, a meta-graph $G = (V, E)$ consists of vertices $V$ and edges $E$, where each vertex corresponds to a social network event and is characterized by a feature vector $X_v$.

The node feature vector $X_v$ encapsulates user attributes, tweet content, and cascade structural information. Specifically, $X_v = (C_{emb} ||| X_{ua} ||| T_{emb} ||| S . . . )$, where $|||$ signifies vector concatenation. The term $C_{emb}$ denotes the cascade embedding vector, $X_{ua}$ encompasses concatenated user features, $T_{emb}$ is the text embedding vector, and $S$ represents sentiment analysis scores if available. Perozzi, Al-Rfou, and Skiena (2014) and Devlin et al. (2019).

Edges in the meta-graph are constructed based on common users or topics and are described by edge feature vectors $R_{u_{ij}}$. An example vector is $R_{u_{ij}} = (N_{ui} ||| V_{ui} ||| H_{ui} . . . )$. Here, $N_{ui}$ indicates the number of common users between cascades $i$ and $j$, $V_{ij}$ is a graph similarity metric, and $H_{ij}$ is a content similarity metric between the root-tweets of the two cascades (Zager and Verghese 2008). Blondel et al. (2008) Zhao et al. (2020) Bisseling (2020) Gomaa and Fahmy (2013). Both node and edge feature vectors are designed for extensibility ease, allowing the incorporation of additional relational information when needed.

Despite its utility in certain scenarios, the meta-graph method presents several notable limitations that hamper its broader applicability. One primary drawback is the high density of its graph representation, which demands considerable computational resources and results in an increased hardware footprint. Additionally, the meta-graph edges are confined to capturing bipartite connections between entities, failing to represent more complex relational information. This restrictiveness also impedes the exploitation of partial feature overlaps among different user clusters. To overcome these challenges, we introduce HyperGraphDis, a novel approach that employs a hypergraph data structure to address these inherent shortcomings effectively.

**HyperGraph for Fake News Detection (HGFND)**

In Jeong et al. (2022), the authors tackle the challenge of detecting fake news by framing it as a semi-supervised node classification task within the context of a hypergraph. They propose a hypergraph construction methodology that establishes connections between news pieces (nodes) based on both the news content and social context—the propagation of news on Twitter. They create a hypergraph with three hyperedge types: i) **User**: a hyperedge for each user I.D., linking news pieces shared by a specific user; ii) **Time**: a hyperedge for a proximal time range, connecting news pieces shared during the same period; iii) **Entity**: a hyperedge for each entity, connecting news pieces that share the same content entity (e.g., "Covid-19").
Finally, they introduce HGFND, a hypergraph neural network model designed to handle noisy relations within the hypergraph. HGFND employs a dual-level attention mechanism, assessing the significance of news relations at both node and hyperedge levels. Experimental results on two benchmark datasets (Politifact and Gossipcop) demonstrate HGFND’s high performance in the news classification task. We evaluate the performance of HGFND in the cascade classification task with the hypergraph constructed following our approach (see Section The HyperGraphDis Method) and the results are shown in Table 3.

Cluster-GCN

In (Chiang et al. 2019), the authors present Cluster-Graph Convolutional Network (Cluster-GCN) that leverages graph clustering to achieve optimized Stochastic Gradient Descent (SGD)-based training in a mini-batch fashion. Cluster-GCN first partitions the graph utilizing METIS (Karypis and Kumar 1998) – a graph partitioning algorithm, to derive non-overlapping clusters of nodes. As a mini-batch of size $q$, it randomly selects $q$ clusters to construct a subgraph with nodes from the chosen clusters and the links connecting them in the original graph.

For each SGD update on a mini-batch, this approach constrains GNNs to convolve exclusively within their designated subgraphs, thereby addressing the issue of neighborhood expansion. This restriction results in optimized memory consumption and reduced computational costs during training, allowing large-scale training on graph structures. In this work, we evaluate the performance of Cluster-GCN on the Metagraph data structure as a baseline. Note that METIS in Cluster-GCN partitions the Metagraph in clusters of cascades, while in HyperGraphDis, METIS generates clusters of users applied on the social network.
The HyperGraphDis Method

The HyperGraphDis method addresses the detection of disinformation in social networks by exploiting network structure as well as post content and user sentiment. As we will show in the next section, the network structure has a clear impact on classification results. As part of the HyperGraphDis method, a hypergraph is constructed from the user-user social network as well as the individual tweet-retweet cascades, which will serve as the nodes of the hypergraph. The hypergraph structure is used to perform binary classification of retweet cascades into fake and not-fake classes. Two primary challenges need to be tackled to apply our methodology to the raw Twitter data. First, the actual structure of the social network, specifically the follower graph, remains largely unknown, as access to users’ follower lists is not always available. Second, the data obtained through the Twitter API offer limited insights into the origins of influence within retweet cascades. This is implemented by the social network as a privacy-preserving measure. The only accessible information consists of the root-tweet and the root-user associated with a particular retweet. In simpler terms, all retweeters are linked to the root user, creating a star-like cascade structure that does not always accurately represent user-to-user interactions or the course of information diffusion. The methodology steps to address the above issues are the following:

Phase-1: The hypergraph construction

Given a large collection of tweets, we construct a hypergraph with the retweet cascades as nodes. (i) From the overall collection of tweets (tweets, replies, retweets), we construct the user-user social network – the graph of interactions between the users. This graph represents an approximation of the Twitter social network. (ii) We partition this graph (in disjoint sets of nodes/users) by applying METIS – a state-of-the-art graph partitioning algorithm. Then, in each partition (set of users), we replace the users with the retweet cascades that have participated. In simple words, clusters of cascades replace clusters of users, and each of these clusters represents a hyperedge.

Figure 1 presents a toy example of the main HyperGraphDis phases for the hypergraph construction where the classification of disinformation is taking place. The hypernodes are the cascades that are represented by feature vectors (as defined in the next Section, “Phase-2: The Hypernodes features vectors”). In our methodology, a hyperedge connects two or more cascades if the users of the same partition of the Twitter social graph participate in these cascades. To illustrate this, consider the social network and the corresponding graph partition in Figure 1(A) & (B). Initially, we create a user-to-cascades mapping, as shown in Figure 1(C). Subsequently, we replace each user with a given partition from Figure 1(B) with the cascades they have engaged with from Figure 1(C). This process leads to the formation of the hypergraph (Figure 1(D)) where the hyperedges connect these sets of cascades. The construction of the hypergraph is scalable since several types of cascade connections (hyperedges) can be defined. For example, we can construct hyperedges based on tweet hashtags; i.e., two or more cascades belong to the same hyperedge if the root-tweets have the same hashtag. In Figure 2, we illustrate the process of building the cascades hypergraph. This begins with the formation of the social network and follows through to capturing users’ participation in cascades, all based on the selected Twitter dataset under consideration. A detailed description of the proposed pipeline is as follows:

(A) Social Network Construction Initially, we create an approximation of the Twitter social graph, also known as the follow-graph, grounded in the activities of users within our dataset. Within this social network, each user corresponds to a node. We add a directed edge for any interaction between two nodes. The “interactions” are defined by the user-user actions in the Twitter platform, such as “retweet,” “reply”, or “mention”. Hence, an edge(i,j) between two nodes implies that the node i follows the node j. The output of this process is a multi-graph because of the possibility of multiple interactions (edges) between two nodes. We remove the duplicate edges and keep the earliest one.

(B) Graph Partitioning Having the resulting Twitter social graph, we conduct graph partitioning to identify communities or groups of users. After applying the selected graph partitioning technique, we get a list of users per partition. In this paper, we leverage the METIS graph partitioning algorithm [Karypis and Kumar, 1998].

(C) User to Cascades mapping We define a list of users that are involved in each cascade based on the raw Twitter data. Finally, we derive a user-to-cascade mapping that denotes that a given user i participated in the set of cascades {x, y, z}.

(D) The final Hypergraph Using the graph partition from Step-(B) and the mapping User-to-Cascades from Step-(C), we replace the users in the partition clusters with the list of cascades they participated in. This process results in sets of cascades (removing the duplicates), which will de-
fine the hyperedges of the final hypergraph.

**Phase-2: The Hypernodes features vectors**

A retweet cascade is a series of chain events upon the same root-tweet where users have retweeted the root-tweet, whereas some others have retweeted a retweet of a friend (retweet of a retweet). This tree-like structure reflects who was influenced by whom during the retweeting process. Unfortunately, the raw data by Twitter API does not provide that since it contains only the root-user and retweeter I.D.s. In short, the raw data represents a star-like graph where all retweeters are only connected to the root user. This form lacks sufficient structural information.

To tackle this issue, we construct a subgraph comprising the star-like graph along with past interactions involving the users who have participated in the cascade in question. Specifically, for a given retweeter i who engaged in the retweet action at date t, we determine their pre-date t connections among their friends and ascertain which of them are among the retweeters in this cascade. Subsequently, we augment this collection of edges, if present, with the star-like configuration where each retweeter maintains a connection with the root user. Finally, we eliminate any duplicate edges and transform the resulting graph into an undirected representation. Through this methodology, the subgraph consistently retains its connectivity.

The rationale of this approach is that each cascade (i.e., a node of the hypergraph) should have the corresponding subgraph structure as a feature that will reflect the social interactions that the retweeters had. For this purpose, we apply the DeepWalk algorithm (Perozzi, Al-Rfou, and Skiena 2014) to produce the embedding of a cascade subgraph to a low dimensional space. For the implementation, we used the `Karate Club` extension for the NetworkX (Rozemberczki, Kiss, and Sarkar 2020). The DeepWalk embedding is a $N \times 128$ matrix, where $N$ is the number of retweeters. The default parameters are: (i) Number of random walks = 10; (ii) Length of random walks = 80 and for embeddings (iii) Dimensionality of embedding = 128.

Figure 2 provides a toy example: (a) Left-hand figure: the raw data that are provided by the Twitter API correspond to a star-like graph structure with limited structural information regarding the user-user relation between the retweeters. (b) Right-hand figure: The augmented Bottom figure: the final cascade object – a set of features that represent the retweet cascade. The feature vector consists of the individual users’ features, the DeepWalk embeddings as well as topic and sentiment analysis (as the tweet-text features). This will be part of the training data of HyperGraphDis.

**Phase-3: Cascade Classification**

We formulate the cascade classification problem as a node classification task. The HyperGraphDis uses a Hypergraph Convolution (HypergraphConv) layer (Bai, Zhang, and Torr 2021) followed by three linear (fully connected) layers. Given a hypergraph structure from Phase 1 and 2, the nodes, their feature vectors, and the hyperedge index serve as input to the HypergraphConv layer, where information aggregation occurs within hyperedges, considering the collective influence of interconnected nodes. The output is a set of transformed node features representing the updated state of each node. The architecture employs Rectified Linear Unit (ReLU) activation functions for non-linearity after each layer. We add dropout regularization between linear layers to enhance generalization. The output of the pipeline represents the model’s predictions (i.e., cascade is fake or non-fake). While training Graph Neural Networks (GNNs), it is crucial to highlight the absence of label leakage issues. This is attributed to the fact that, during the training phase, the labels of nodes beyond the training dataset remain unknown. This ensures the integrity of the training process, preventing unintended information leakage from labels outside the designated training set.

**Dataset Descriptions**

**US Elections Dataset** The dataset in (Salamanos et al. 2023), encompassing 152.5 million tweets from 9.9 million users during the 2016 U.S. presidential election, was harvested between September 21st and November 7th, 2016, using the Tweepy Python library. Each tweet is feature-rich, encapsulating 27 attributes related to the tweet and the corresponding Twitter account, including tweet-ID, tweet-text, user-ID, screen name, and more. Additional metadata, such as mentions and embedded URLs, are also part of the dataset. The dataset also emphasizes retweet cascades, specifically those rich in transmitted information and broad user engagement. This includes cascades where the original tweet contains at least one web or media URL and has been retweeted by a minimum of 100 users. In total, 46.4K such retweet cascades, involving 19.6 million tweets, are included for analysis. With labeled media and web URLs – embedded in the tweets – categorized as “fake,” “non-fake,” or “unknown,” derived through a multi-step methodology and manual annotation using Fact-Checking tools, the dataset provides a well-curated set of 43.9K URLs. There are 6525 URLs labeled as “fake” and “non-fake” that are present in 8323 cascades (root tweets).

**MM-COVID Dataset** The authors in (Li et al. 2020) created MM-COVID, a multilingual and multimodal dataset to combat misinformation on COVID-19. Despite the news content, the dataset also contains Twitter’s social network and spatial-temporal information. The dataset consists of approximately 4K fake news articles and about 7K trustworthy articles related to COVID-19 across six languages. There are over 1K fake news documents in the English language subset and approximately 5K legitimate articles. We dehydrated the user engagement data only for the English language subset using Twarc (a Python library for collecting data via Twitter API). Only a subset of the data was available to collect at the time of the dehydration of the existing dataset. In summary, we used 53,869 tweets, 10,387 replies, and 85,121 retweets by 93,152 users for the experimental evaluation. From this dataset, we kept the cascades that have at least 10 participants. The tweets are in English only; the quoted tweets have been discarded.

**Dataset Descriptions**

**US Elections Dataset** The dataset in (Salamanos et al. 2023), encompassing 152.5 million tweets from 9.9 million users during the 2016 U.S. presidential election, was harvested between September 21st and November 7th, 2016, using the Tweepy Python library. Each tweet is feature-rich, encapsulating 27 attributes related to the tweet and the corresponding Twitter account, including tweet-ID, tweet-text, user-ID, screen name, and more. Additional metadata, such as mentions and embedded URLs, are also part of the dataset. The dataset also emphasizes retweet cascades, specifically those rich in transmitted information and broad user engagement. This includes cascades where the original tweet contains at least one web or media URL and has been retweeted by a minimum of 100 users. In total, 46.4K such retweet cascades, involving 19.6 million tweets, are included for analysis. With labeled media and web URLs – embedded in the tweets – categorized as “fake,” “non-fake,” or “unknown,” derived through a multi-step methodology and manual annotation using Fact-Checking tools, the dataset provides a well-curated set of 43.9K URLs. There are 6525 URLs labeled as “fake” and “non-fake” that are present in 8323 cascades (root tweets).

**MM-COVID Dataset** The authors in (Li et al. 2020) created MM-COVID, a multilingual and multimodal dataset to combat misinformation on COVID-19. Despite the news content, the dataset also contains Twitter’s social network and spatial-temporal information. The dataset consists of approximately 4K fake news articles and about 7K trustworthy articles related to COVID-19 across six languages. There are over 1K fake news documents in the English language subset and approximately 5K legitimate articles. We dehydrated the user engagement data only for the English language subset using Twarc (a Python library for collecting data via Twitter API). Only a subset of the data was available to collect at the time of the dehydration of the existing dataset. In summary, we used 53,869 tweets, 10,387 replies, and 85,121 retweets by 93,152 users for the experimental evaluation. From this dataset, we kept the cascades that have at least 10 participants. The tweets are in English only; the quoted tweets have been discarded.

**Dataset Descriptions**

**US Elections Dataset** The dataset in (Salamanos et al. 2023), encompassing 152.5 million tweets from 9.9 million users during the 2016 U.S. presidential election, was harvested between September 21st and November 7th, 2016, using the Tweepy Python library. Each tweet is feature-rich, encapsulating 27 attributes related to the tweet and the corresponding Twitter account, including tweet-ID, tweet-text, user-ID, screen name, and more. Additional metadata, such as mentions and embedded URLs, are also part of the dataset. The dataset also emphasizes retweet cascades, specifically those rich in transmitted information and broad user engagement. This includes cascades where the original tweet contains at least one web or media URL and has been retweeted by a minimum of 100 users. In total, 46.4K such retweet cascades, involving 19.6 million tweets, are included for analysis. With labeled media and web URLs – embedded in the tweets – categorized as “fake,” “non-fake,” or “unknown,” derived through a multi-step methodology and manual annotation using Fact-Checking tools, the dataset provides a well-curated set of 43.9K URLs. There are 6525 URLs labeled as “fake” and “non-fake” that are present in 8323 cascades (root tweets).

**MM-COVID Dataset** The authors in (Li et al. 2020) created MM-COVID, a multilingual and multimodal dataset to combat misinformation on COVID-19. Despite the news content, the dataset also contains Twitter’s social network and spatial-temporal information. The dataset consists of approximately 4K fake news articles and about 7K trustworthy articles related to COVID-19 across six languages. There are over 1K fake news documents in the English language subset and approximately 5K legitimate articles. We dehydrated the user engagement data only for the English language subset using Twarc (a Python library for collecting data via Twitter API). Only a subset of the data was available to collect at the time of the dehydration of the existing dataset. In summary, we used 53,869 tweets, 10,387 replies, and 85,121 retweets by 93,152 users for the experimental evaluation. From this dataset, we kept the cascades that have at least 10 participants. The tweets are in English only; the quoted tweets have been discarded.
outright deletion of Twitter user accounts. (2) deletions of tweets by the original authors. Several factors contribute to this discrepancy: (1) modifications in user settings that restrict publishing from the FakeHealth repository (Dai, Sun, and Wang 2020). While the authors of the repository are constrained by Twitter’s user privacy policies from publicly disseminating the full extent of user engagement and network data, they have circumvented this limitation by offering an API [https://github.com/EnyanDai/FakeHealth](https://github.com/EnyanDai/FakeHealth). This API facilitates the acquisition of comprehensive data pertaining to users’ social engagements and networks. Using this API, we collected the following datasets: i) Health Release with a size of 60,006 – 43,245 tweets, 1,418 replies, and 15,343 retweets and ii) Health Story with a size of 487,195 – 357,851 tweets, 23,632 replies, and 105,712 retweets. It is noteworthy that the final data metrics we report exhibit a marginal reduction, approximately 1.1%, compared to the figures disseminated by the original authors. Several factors contribute to this discrepancy: (1) modifications in user settings that restrict public accessibility to their tweets, (2) deletions of tweets by Twitter in adherence to their internal policies, and (3) the outright deletion of Twitter user accounts.

FakeHealth Datasets To rigorously assess the efficacy of our proposed methodology, we employ two datasets originating from the FakeHealth repository [Dai, Sun, and Wang 2020](https://www.mapequation.org/infomap/). While the authors of the repository are constrained by Twitter’s user privacy policies from publicly disseminating the full extent of user engagement and network data, they have circumvented this limitation by offering an API [https://github.com/EnyanDai/FakeHealth](https://github.com/EnyanDai/FakeHealth). This API facilitates the acquisition of comprehensive data pertaining to users’ social engagements and networks. Using this API, we collected the following datasets: i) Health Release with a size of 60,006 – 43,245 tweets, 1,418 replies, and 15,343 retweets and ii) Health Story with a size of 487,195 – 357,851 tweets, 23,632 replies, and 105,712 retweets. It is noteworthy that the final data metrics we report exhibit a marginal reduction, approximately 1.1%, compared to the figures disseminated by the original authors. Several factors contribute to this discrepancy: (1) modifications in user settings that restrict public accessibility to their tweets, (2) deletions of tweets by Twitter in adherence to their internal policies, and (3) the outright deletion of Twitter user accounts.

Table 1: Cascades, social network and metagraph statistics per dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#fake (users)</th>
<th>#non-fake (users)</th>
<th>#nodes (cascades)</th>
<th>#edges</th>
<th>#connected components</th>
<th>Giant Component % of nodes</th>
<th>Infomap #comm</th>
<th>Louvain #comm</th>
<th>#nodes (full extent)</th>
<th>#edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Election</td>
<td>5,890</td>
<td>2,433</td>
<td>482,171</td>
<td>34,691,548</td>
<td>1</td>
<td>100%</td>
<td>2047</td>
<td>14</td>
<td>8,323</td>
<td>7,763,018</td>
</tr>
<tr>
<td>Health Story</td>
<td>3,574</td>
<td>9,689</td>
<td>147,607</td>
<td>187,096</td>
<td>12,719</td>
<td>73.75%</td>
<td>12894</td>
<td>12850</td>
<td>13,263</td>
<td>100,001</td>
</tr>
<tr>
<td>MM-Covid</td>
<td>578</td>
<td>1,230</td>
<td>38,813</td>
<td>50,688</td>
<td>4</td>
<td>99.78%</td>
<td>47</td>
<td>101</td>
<td>1,808</td>
<td>87,948</td>
</tr>
<tr>
<td>Health Release</td>
<td>929</td>
<td>1,040</td>
<td>18,470</td>
<td>21,502</td>
<td>1,595</td>
<td>73.32%</td>
<td>1645</td>
<td>1659</td>
<td>1,969</td>
<td>11,581</td>
</tr>
</tbody>
</table>

Table 2: Cascade features

<table>
<thead>
<tr>
<th>Features</th>
<th>US Election</th>
<th>MM-Covid</th>
<th>Fake Health</th>
</tr>
</thead>
<tbody>
<tr>
<td>User-related</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DeepWalk Embedding</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Account Creation Date</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Verified Account</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Account Language</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Max Statuses count</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Max Favorites count</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Max listed count</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Reaction Time</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Max Followers count</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Max Friends count</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Max impression count</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Max reply count</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Max quote count</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max like count</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text-related</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sentiment Analysis</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Topic Detection</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Evaluation of Performance Benefits

In this section, we evaluate HyperGraphDis across all four previously mentioned datasets and simultaneously compare it with the Meta-graph, HGFND and Cluster-GCN approaches. First, we present the basic structures across the four datasets that are the input of the HyperGraphDis and the other three methods, and then we discuss the actual classification performance.

The code has been implemented in Python and executed in the CPU of an ASUS laptop with Intel Core i7-6700HQ CPU 2.60GHz × 8; 32GB RAM; Ubuntu 22.04.3 LTS. The HypergraphConv [Bai, Zhang, and Torr 2019], GCNConv [Kipf and Welling 2017] and Cluster-GCN (Chiang et al. 2019) have been implemented in PyTorch Geometric [Fey and Lenssen 2019], maintaining the following hyper-parameters constant across experiments: #epochs = 100, dropout = 0.5, learning rate = 5*10^{-4}. For the HGFND we use the official implementation[4] We use the default parameters dropout = 0.3 and learning rate = 0.001. We train the model for 100 epochs. We use an 80/20 train-test split on the labeled cascade set across all datasets and methods. We repeat the experiments for five trials, where we regenerate the train-test split randomly in each trial. Finally, we use the official implementation of Infomap[4] executing multi-level clustering for 10 internal trials. Louvain has been implemented by NetworkX[4].

Cascades, Social Networks and Metagraphs The statistical characteristics of the cascades, social networks, and metagraphs constructed for the four datasets are presented in Table[5]. In the US Election dataset, since the metagraph constructed is extremely dense (8323 nodes/cascades are connected by 15,946,910 undirected edges), we applied the disparity filter (Serrano, Boguñá, and Vespignani 2009) for α = 0.5, which depicts the best classification results according to (Paraschiv et al. 2022). The 49% of the edges survived the significance test.

Cascade features In all datasets, we define a threshold for the number of initial retweeters (users) that we will include in the features (to prevent out-of-memory issues). The thresholds are US-Election: 250; MM-COVID: 50; FakeHealth: 60. After constructing the feature matrix, we apply dimensionality reduction using PCA. The dimensions are reduced to US-Election: 90; MM-COVID: 60; FakeHealth: 60.
Table 1 presents the selected parameter values and outlines the user and text-related features employed in the creation of cascade features for each dataset. Note that certain features are not available for certain datasets.

**User-related features** The user-related features of a given node/cascade with $N$ users (the retweeters plus the root–user) may include the following: The DeepWalk embedding of the correspondent subgraph – a $N \times 128$ matrix; The dates of users’ account creation; The maximum value of users’ followers count; The maximum value of users’ friends count; The maximum value of users’ favorites count; The maximum value of users’ statuses count; The maximum value per user and metric across all the tweets that she has posted: listed count; impression count; reply count; quote count; like count; A Boolean identifier whether the users’ account is verified; The language of the users’ accounts; users’ “reaction time”, namely, the time difference between a retweet and the root-tweet.

**Text-related features** In all datasets, we have included the sentiment (label, score) of the root–tweet – two values, either (2, score) for ‘Positive’ label or (1, score) ‘Negative’ label. We use the Transformer library for the sentiment analysis. In addition, we performed topic detection for the US-Election dataset only, following the approach of Paraschiv et al. (2022). For this purpose, we employ a Hierarchical Dirichlet Process (HDP) (Teh et al. 2005; Newman et al. 2009), using the Tomotopy library. Then, we include the most representative topic of the root–tweet in the cascades’ feature vector based on the three topic models (the three versions of the HDP model). For the two FakeHealth datasets, we set as topic (per cascade) the news’ “tags”. This information is provided by the “reviews” in the FakeHealth repository.

**Main Results**

In this section, we delve into the results obtained from implementing our HyperGraphDis method across various datasets and compare them against existing state-of-the-art models. The results are summarized in Table 2. Specifically, we experimented with varying cluster sizes, ranging from 20 to 5000 (US-Election and Health Story), all of which were derived using the METIS method for graph partitioning (see Table 1). For MM-COVID and Health Release, the largest METIS partition we conducted is 1000 due to the small size of their social networks (size in terms of the number of users, see Table 1). We have conducted the following experiments: 1) Evaluation of HypergraphDis on hypergraphs constructed by METIS partition versus on hypergraphs produced by Infomap (Rosvall and Bergstrom 2008) and Louvain (Blondel et al. 2008) community detection algorithms. Both approaches are applied to users’ social networks as we presented in Figure 1. Table 1 presents the general statistics of the social networks. Generally, the community detection methods generate clusters of users that are imbalanced in size, whereas METIS clusters are generally balanced.

2) HyperGraphDis versus the vanilla version of Meta-graph.
METIS methodology involves recursively partitioning a graph into more manageable subgraphs to identify densely connected regions. The process minimizes inter-cluster connections and balances load distribution, ensuring roughly equal cluster sizes. On the other hand, although Louvain and Infomap are highly efficient in identifying communities within complex networks, the communities they produce are not well interconnected and are not balanced in size. An indicator for that is the training time on the hypergraphs produced by the Infomap is very low for the US-Election Dataset, although the number of 2074 communities is quite high (see Table 1). HyperGraphDis (Infomap) achieved a 77.17% F1 score in 113.77 seconds, whereas HyperGraphDis (METIS;500) achieved 85.73%, but the training time took 1704.78 sec.

In summary, the results confirm that HyperGraphDis offers a compelling improvement over existing methods for identifying misinformation on Twitter. Not only does it deliver more accurate and nuanced results, but it also does so more efficiently, which is an essential quality for real-time or large-scale applications.

**Ablation Analysis**

We conducted an ablation analysis on MM-COVID and Health Release to quantify (i) the effect of the number of HyperGraphConv layers and (ii) the effect of the training set size. In short, we selected two datasets where HyperGraphDis exhibited the best and worst performance (see Table 3). The ablation analysis was performed on the hypergraphs constructed by METIS for different numbers of clusters (ranging from 100 to 1000).

**Table 3: HyperGraphDis versus other state-of-the-art approaches**
Figure 3 presents the average values (from 5 trials) of accuracy and weighted test F1 score. The experiments with different numbers of HypergraphConv layers (from 1 to 5) indicate that applying several layers does not increase the utility of the model (Figures 3(a) & (b)). Only in Health Release did we achieve the best accuracy and F1 score with a model having two HypergraphConv layers.

We also experimented with varying training set sizes, ranging from a train-test split of 10/90 to 80/20. Figure 3(c) shows that in MM-COVID, a training set size of around 40% is adequate to produce a model that can achieve an F1 score close to 89%. In Health Release (Figure 3(d)), a training set size of 60% is close to optimal.

Further Discussion on HyperGraphDis Performance

In this study, we evaluated the HyperGraphDis method using the MM-COVID and Health Story datasets. Other studies that used these two datasets are the following:

**MM-COVID** In [Kozik et al. 2023], a meta-analysis explored state-of-the-art Machine Learning techniques for text-based fake news detection. Models, including Random Forest, CNN-based, LSTM-based, and BERT-based, underwent formal evaluation on benchmark datasets, such as MM-COVID. Notably, the BERT-based model achieved the highest F1 score (79.4%), while Random Forest had the lowest score (63.3%). The HyperGraphDis, with an F1 score of 89.48%, showcases a significant 10-26% improvement over conventional text-based methodologies. This performance advantage underscores the value of incorporating social network graph data and user-related features alongside text-based features in enhancing fake news detection tasks.

**Health Story** SOMPS-Net framework [Dhanasekaran et al. 2021], comprising two essential components: the Social Interaction Graph and the Publisher and News Statistics. Tested with the Health Story dataset, it achieved a 72.7% accuracy and 79.6% F1 score in the realm of fake news prediction. We tested HyperGraphDis using the same dataset, although we observed slightly diminished results regarding the F1 score, specifically an accuracy of 75.21% and an F1 score of 72.23%. We attribute this decline in performance to the inherent characteristics of the cascades within the Health Story dataset, which tend to be either extremely small or predominantly structured like star-like graphs. This characteristic leads to embedding vectors with significant similarity, impacting the overall effectiveness of our method.

**Limitations** Although HyperGraphDis is computationally efficient (in terms of training and inference process time), making it suitable for disinformation detection in large datasets, there are still some limitations. First, a large number of tweets is needed to construct the social network – a crucial component of the HyperGraphDis pipeline (see Figure 1). The social network – the user-user interaction graph – is constructed by the interaction between the users (replies, retweets, mentions). For the produced social network to be valuable, it needs to have a rich graph structure, which hea-
not upload any of the datasets to the cloud. We executed the experiments locally on our machines, so we did not present any identifiable user information from the datasets that have already been published. We did not use the projects MedDMO (Grant Agreement no. 101083756) and INCOGNITO (Grant Agreement no. 824015).

Table 4: Other disinformation detection methods on MM-COVID and Health Story datasets.

<table>
<thead>
<tr>
<th>Data Source</th>
<th>ML Model</th>
<th>Acc</th>
<th>F1 Score</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>MM-COVID</td>
<td>Random Forest</td>
<td>87.6%</td>
<td>63.3%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CNN</td>
<td>88.1%</td>
<td>67.3%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LSTM</td>
<td>90.4%</td>
<td>75.0%</td>
<td>news content</td>
</tr>
<tr>
<td></td>
<td>BERT</td>
<td>92.5%</td>
<td>79.4%</td>
<td></td>
</tr>
<tr>
<td>Health Story</td>
<td>SOMPS-Net</td>
<td>GCN</td>
<td>72.7%</td>
<td>79.6%</td>
</tr>
<tr>
<td></td>
<td>Dhanasekaran et al. 2021</td>
<td>GCN</td>
<td>72.7%</td>
<td>79.6%</td>
</tr>
</tbody>
</table>

Conclusion

The problem of disinformation on social media is a pressing issue with substantial social, economic, and political implications. In this paper, we introduce a novel method, HyperGraphDis, designed to combat the spread of disinformation by analyzing Twitter data through the lens of a hypergraph data structure. Hypergraphs offer several advantages over traditional graph-based methods. They provide enhanced scalability, which is crucial for efficiently analyzing vast social media datasets. We applied HyperGraphDis to diverse datasets, including health and socio-political domains to validate our approach. The method’s versatility was evident, showcasing its potential across various applications. Comparative tests against state-of-the-art approaches revealed significant accuracy and F1 score improvements, crucial for imbalanced datasets common in disinformation detection. Furthermore, our method displayed improvements in computational cost and run times, reinforcing its advantages in terms of scalability and efficiency.

Ethics

This work followed the principles and guidelines on executing ethical information research and using shared data (Kelly et al. 2012). The suggested methodology complies with the GDPR and ePrivacy regulations. We used Twitter datasets that have already been published. We did not use or present any identifiable user information from the datasets (e.g., Twitter user IDs). We applied text preprocessing to clean the tweets from any information that could identify specific Twitter accounts. Finally, we implemented and executed the experiments locally on our machines, so we did not upload any of the datasets to the cloud.

Acknowledgments

This work has been funded by the European Union under the projects MedDMO (Grant Agreement no. 101083756) and INCOGNITO (Grant Agreement no. 824015).

References


