# Countering the Spread: An Approach to Identify Misinformation Spreaders in Social Media

Antonela Tommasel, *Member, IEEE* and Juan Manuel Rodriguez

***Abstract*—Although social media generally provides a safe and enjoyable experience, it can also serve as a quick and easy means for spreading false news, misinformation, and other harmful content. These contents have been proven effective in influencing people's beliefs and behaviors, spanning from influencing political opinions to directly impacting public health, particularly during events such as the COVID-19 pandemic. Then, it becomes crucial to take proactive measures to identify the misinformation spreaders to mitigate their impact and influence over society. Existing approaches primarily focus on identifying spreaders by analyzing features such as writing style, content, user profiles, and engagement statistics. However, since fake or deceiving content is frequently crafted to closely resemble authentic information, traditional techniques alone prove insufficient to effectively identify either fake content or its spreaders. In this context, this work introduces a deep-learning model tailored for detecting misinformation spreaders in social media. Our model not only incorporates content-based features but also integrates patterns of social interactions and information propagation structures. By considering the multifaceted nature of misinformation spread, our approach provides a more holistic and accurate means of identifying its spreaders. An experimental evaluation focusing on COVID-related data yielded promising results, demonstrating a significant performance improvement compared to other techniques in the literature. Thus, this research contributes to the ongoing efforts to develop robust tools for reducing the adverse effects of misinformation in social media.**

***Index Terms*—Fake news spreaders, profiling, social media.**

## I. Introduction

In recent years, social media has significantly transformed how individuals access information, establishing itself as a primary news outlet [21]. Social media has become a key player in propagating and amplifying information, even contributing to the effectiveness of government communication strategies [8]. However, its unmoderated nature and susceptibility for fast propagation creates an ideal environment for propagating inaccurate or intentionally deceiving information, posing a challenge to ensuring access to reliable information.

Received 28 March 2024; revised 19 January 2025 and 26 February 2025; accepted 7 March 2025. Date of publication 28 March 2025; date of current version 2 October 2025. *(Corresponding author: Antonela Tommasel.)*

Antonela Tommasel is with Instituto Superior de Ingeniería del Software de Tandil (ISISTAN), National Council of Scientific and Technological Research (CONICET) - Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), 7000 Tandil, Argentina (e-mail: antonela.tommasel@isistan.unicen.edu.ar).

Juan Manuel Rodriguez is with Aalborg University, 9220 Aalborg, Denmark.

Digital Object Identifier 10.1109/TCSS.2025.3550029

As social media usage continues to rise, the spread of fake news, misinformation, and other forms of harmful content (e.g., rumors, conspiracies) consolidates as a major societal concern [30]. These phenomena have far-reaching effects from undermining democratic systems to posing public health issues, as evidenced during the COVID-19 pandemic [42], [52], exacerbated by the rapid generation and propagation of content within the evolving pandemic context [9]. This propagation is strengthened by newsfeed algorithms, which trap individuals in echo chambers and overwhelm them with unreliable content that appears authentic [3]. These dynamics, coupled with factors like political affiliation and mental health [51], can shape how people perceive and engage with content [21]. Continuous exposure to misinformation increases the likelihood of individuals accepting fake content as truth, particularly when it aligns with their beliefs and originates from seemingly trusted sources [20], [51]. Moreover, the spread of fake content blurs the line between what is genuine and what is not, fostering a sense of skepticism [21]. Consequently, in the long term, the overall trustworthiness, value, and quality of the news ecosystem may be jeopardized. Then, detecting and mitigating the propagation of fake content is essential.

Users play a crucial role in creating and propagating unreliable and fake content. Hence, it is vital to identify not only the unreliable content, but also the users spreading it, as they can offer valuable insights for devising strategies to promptly contain the propagation [43]. The influence of time and evolving context has added complexity to the task, due to the dynamic nature of shared content and interaction patterns [16], [35]. In this context, the need to identify spreaders efficiently and rapidly has never been more pressing.

### A. Contributions

Our main contributions are:

*We introduce a deep learning model for identifying fake news spreaders on social media.* Our model incorporates features extracted from the content shared and user profiles, as well as patterns of social interactions within user communities (by graph neural networks) and temporal conversation structures (by gated recurrent units).[1] Given the evolving nature of misinformation propagation and social media dynamics, our proposal addresses two distinctive settings. First, we aim to predict the behavior of unseen users in the

[1]We publicly release the processed data and code to foster transparency and future research.

future. Here, the goal is to determine whether the defined features and structures can generalize to the future activities of unknown users.

*We focus on the timely detection of spreaders over time.* In this scenario, we dynamically analyze features and patterns of communication and conversation propagation based on user behavior across consecutive time intervals. Unlike the first scenario, where features and structures are built based on complete propagation and interaction networks after posts have been spread, here we model the sequence and timing of interactions over time. This approach not only captures interaction and propagation structures, but also enables the understanding of temporal dynamics.

To validate our proposal, we conducted an experimental evaluation over a data collection focused on COVID-19 misinformation. Through ablation experiments, we assessed the contribution of each model's components for spreader profiling. Results showed that the proposed model outperforms traditional and state-of-the-art baselines in both detection scenarios, effectively identifying fake news spreaders.

## II. Related Work

Detecting fake news and other forms of harmful content has been widely studied, ranging from the analysis of linguistic and user profile features [46], [47] to network features [33] and their combinations [36], [37]; serving as a starting point for profiling fake news spreaders. While user profiling has been a common focus across various tasks, fake news spreader profiling remained comparatively understudied until recently.

### A. Textual and Psico-Linguistic Features

Giachanou et al. [12] considered content-based features, such as LIWC categories, personality traits, communication style, emotion, readability scores, and word embeddings (GloVe and BERT). Giachanou et al. [13] followed a similar approach for detecting conspiracy spreaders. Agarwal et al. [1] also evaluated the degree of clickbaitiness of posts using a pretrained model. In all cases, traditional binary classifiers or simple neural network models were trained. Kumar et al. [24] also focused on content for detecting hate speech and fake news spreaders. They trained different autoencoders using vectorized content from spreaders or nonspreaders. The best results were obtained using an autoencoder trained with the nonspreader TF-IDF content representation and a Naïve Bayes classifier. The authors suggested that traditional classifiers can outperform deep learning models. Similarly, Siino et al. [48] argued against pretrained Transformers as the optimal approach for spreader identification. They proposed a shallow CNN model based on content, which exhibited improvements over fined-tuned Transformers and traditional classifiers trained with $n$-grams.

### B. Hand-Crafted and User Profile Features

Sansonetti et al. [40] chose screenname and description length, number of followers/friends, average number of daily posts, overall sentiment score of posts, and age account, passed through a neural network. Shrestha and Spezzano [45] included demographic (age, gender, and political orientation) and behavioral features (time of day and week). As demographic features were unavailable, they were inferred from the content, potentially inducing biases and stereotypes [6]. Sharma and Sharma [44] combined word2vec with user and engagement features.

### C. Network Topology

Rath et al. [38] targeted the detection of spreaders within densely connected communities, relying on topology community assessments and interpersonal trust. They claimed that their approach outperformed others based on user activity, though no comparison with content-based approaches was provided. Similarly, Qureshi et al. [36] stated that topology-based features resulted more important than user-based ones.

### D. Combination of Features

Hamdi et al. [19] aimed to identify spreaders (defined as low-credibility users) by merging social graph node2vec embeddings with user profile features. The authors found Naïve Bayes to be among the best-performing classifiers. Cabral et al. [2] detected spreaders in WhatsApp groups based on group activity metrics (e.g., number of groups participation, messages sent, media shared, misinformation items sent), temporal factors (e.g., active days, messages per day) and network features (e.g., centrality, strength). Network features were computed based on cogroup graphs, where two users were connected if one sent a message to a group the other user belonged to. Then, spreaders were identified through feature thresholding and a logistic regression classifier. Tommasel et al. [50] also combined content and user features. User features were extracted from the user and direct interactions, disregarding the effect of neighbors (and their social relations) in user representation. Content was represented by BERT embeddings. A naïve representation of down-stream conversations was introduced by using the max-pooled representation of the involved posts, thereby ignoring both their conversational and temporal structures.

### E. Temporal Dimension

Ghanem et al. [10] segmented user timelines into fixed-size chunks, represented by semantic and stylistic features (e.g., emotions, morality, sentiment, style) and GloVe embeddings. While this approach provided a finer-grained characterization of users, chunks' fixed-size hindered the capture of sharing pattern dynamics. Plepi et al. [35] divided user timelines into monthly chunks, and represented users based on the posting activity in each chunk using embeddings from social and semantic similarity graphs. Chunk sequence information was encoded using a recurrent neural network. In both, all chunks were included in training, encoding information for all time slots simultaneously. Despite potential fluctuations in user behavior over time (and chunks), users' spreader condition was statically defined, considering their behavior across all chunks.

Despite the dynamic nature of social media and information propagation, most of the described techniques assume user behavior remains constant over time. Evaluations are often

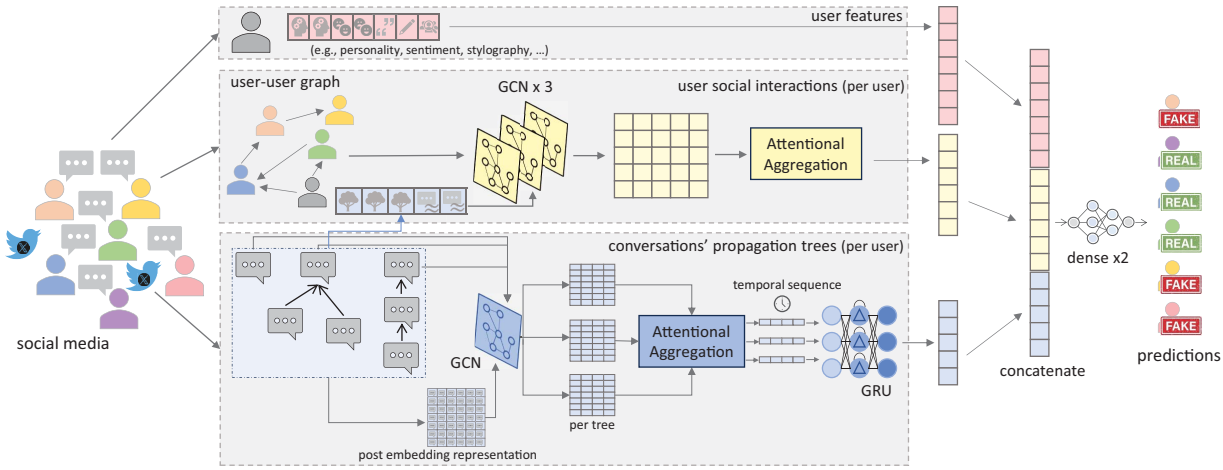


Fig. 1. Schematic diagram of the proposed model.

conducted in static settings, assuming that all user information is preknown. This static approach limits their applicability in scenarios like early spreader detection, where user behavior is limited, incomplete, and subject to change.[2] To address these limitations, our approach builds on existing work by integrating content-based, hand-crafted, and social network topology features with temporal information. By explicitly modeling temporal dynamics and propagation patterns, we aim to capture the evolving nature of user behavior and misinformation spread. Unlike prior methods that mostly treat these aspects independently, our model combines them into a unified framework, enabling a more comprehensive characterization and timely detection of spreaders. We evaluate our method in both static and early detection scenarios, highlighting its adaptability and performance in realistic settings where user behavior evolves over time. This holistic approach not only addresses the limitations of previous works but also advances the state-of-the-art in misinformation spreader detection.

## III. A Deep Learning Model For Fake News Spreaders Detection

We formulate the spreader detection problem as a binary classification task. For a user $u_i$, their historical social interactions network ($\tau(u_i)$), shared content ($T_{u_i}$), and conversation propagation trees ($PrT(T_{u_i})$), the goal is to learn a binary function $F(\tau(u_i), T_{u_i}, PrT(T_{u_i})) \rightarrow \{-1, 1\}$, where 1 defines $u_i$ as a spreader and $-1$ otherwise. Fig. 1 provides an overview of the proposed model architecture, which comprises two steps. The initial step processes users' neighboring and community social structures, conversation propagation structures, and temporal interaction sequences to generate a user representation. Then, the second step passes these user representations through a multilayer perceptron (MLP) to generate a prediction.

[2]As a side note, many of the described approaches lack comprehensive evaluations, often omitting state-of-the-art baselines. This hinders the understanding of their performance and robustness.

### A. User Representation

To account for the diverse and evolving nature of social media and fake news spreaders, we divide user representation into three components: user features, social interactions, and conversations' propagation structures.

*1) User Features:* These features are represented by a vector concatenating attributes such as personality traits, LIWC categories, sentiment, emotions, readability metrics, and user activity statistics.[3] If needed, features with extreme values (i.e., over 1 or below $-1$) are standardized and clipped to the $[-2, 2]$ range. This aims to prevent gradient exploding and speeds up training [26]. Then, this feature vector proceeds to the second step of the model.

*2) Social Interactions:* Social interactions are recognized as crucial for detecting fake news spreaders [47]. Therefore, this component models the user adjacency matrix through graph convolutional networks (GCNs) [23]. GCNs enable node representation by considering both intrinsic features and those from their interactions. To define users' neighborhoods, we extract the subgraph of all user interactions within a distance of 3. This defines users not only by their direct interactions but also by the indirect/implicit community structures. Then, the subgraph, combined with a feature vector, is processed through a three-layer GCN, with each layer defined as

$$GCN(X, \hat{D}, \hat{A}) = f\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}XW + b\right). \quad (1)$$

Here, $A$ represents the adjacency matrix of the subgraph, $I$ represents the identity matrix, $\hat{A} = A + I$, $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$, $f$ represents an activation function (here, ReLU), $X$ represents the users' feature matrix, $W$ represents a matrix of trainable weights, and $b$ represents the trainable bias vector. $X$ differs from the features in the previous component. Inspired in [15], [54], we consider features related to conversation propagation

[3]More details and the full set of features can be found at the companion repository: https://github.com/tommantonela/FN-spreaders.

structures generating the social interactions to capture similarities/differences in user interactions when discussing shared content (e.g., average tree height and size, reciprocity of tree interactions, average semantic similarity between the root of the tree and the other nodes, and the h-index of conversation trees).[4] If necessary, features undergo the same standardization and clipping process as described before.

The output of the GCNs is a matrix where each row vector represents a user in the graph. This matrix is summarized into a single vector using an attention-based aggregation mechanism [27].[5] Unlike pooling-based representations, using attention allows to effectively include neighbors' information into a user representation. The aggregation function is defined as:

$$\mathrm{Agg}(X) = \sum_{n=1}^{N} \mathrm{softmax}\left(h_{\mathrm{gate}}\left(X_n\right)\right) \odot X_n. \tag{2}$$

Here, $N$ represents the number of nodes, $X_n$ the features associated with node $n$, and $h_{\mathrm{gate}}\left(X_n\right)$ an arbitrary neural network that computes the contribution of each user to each feature. We modeled $h_{\mathrm{gate}}\left(X_n\right)$ as a two-layer MLP. The input $X$ is a $N \times F$ matrix, with both hidden and output layers of size $F$. The MLP outputs a matrix of the same size as $X$, with columns passed through the softmax function. Finally, the resulting matrix is element-wise multiplied by $X$ and summed across rows to obtain the vector representing the user graph.

*3) Conversation Propagation Structures:* Given the complex and evolving nature of information propagation, only relying on discriminant features from the conversation propagation structure may inadequately represent it, potentially resulting in a complex and biased representation [29]. For example, focusing solely on summary statistics such as centrality, cliques, or connected components might be too general (and better suited for large structures) [31]. To capture the semantics and dynamics of conversation propagation patterns, we represent each post (tweet) based on the tree derived from its chain of triggered replies/retweets in the conversation. Unlike the user graph that focuses on the complete set of user interactions, these trees focus on the cascades and chains originating from each conversation, regardless of their authors. This representation is complemented by the post embedding representation of all posts in the conversational tree defined as their pooled BERT [7] embeddings, and then passed through a single GCN. Based on Eq. (1), $A$ represents the tree, and $X$ the embeddings.

Similar to the user graphs, each conversation tree is summarised into a vector using the attention-based aggregation mechanism. To capture the temporal sequence of posts, vectors are passed through a gated recurrent unit (GRU), where they are processed from the oldest to the newest date. Finally, from the GRU's output, we select the vector representing the last post as the representation of the set of users' posts. This summarizes the entire sequence, with more weight given to the most recent posts compared to the older ones.

[4]More details can be found at the companion repository.

[5]This mechanism is known as attentional aggregation in the pyTorch geometric framework: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.aggr.AttentionalAggregation.html.

### B. Model Prediction

Once the partial user and tweet representations are obtained, they are concatenated with the user feature vector to form the final user representation. This vector then passes through two dense layers to generate the model's output. The first layer employs a ReLU activation function, while the second has a linear one, as the model's output is expected to be an arbitrary number. The proposed model acts as a soft-margin classifier, where values less than $-1$ indicate the negative class and values greater than 1 indicate the positive class (as explained in the following Section), but it allows values to be returned in the $[-1, 1]$ range, for which 0 acts as the class separator.

### C. Model Training

The model was trained using a Hinge loss function with class weights Eq. (3), where the error margin is set to 1, $y_t$ represents users' true class, $y_p$ the prediction, and $w_{y_t}$ the balanced class weight. Unlike binary cross entropy, which is another widely used loss function for binary classification, hinge loss provides a more aggressive penalty structure for misclassified samples, promoting better separation of classes. The margin allows the Hinge loss to select decision boundaries that are far from both classes, which enhances its robustness [39]. The values for $y_t$ can be 1 or $-1$, where 1 corresponds to the spreader class, and $-1$ corresponds to the nonspreader class. A correct prediction occurs when $y_t$ and $y_p$ have the same sign and $|y_p| \geq 1$, resulting in a $\mathrm{loss} = 0$. If $|y_p| < 1$, the loss increases as the margin is insufficient to consider the prediction correct. Conversely, a prediction is incorrect when $y_t$ and $y_p$ have opposite signs, also leading to an increase in loss

$$\mathcal{L}(y_t, y_p) = \frac{\sum w_{y_t} \cdot \max(0, 1 - y_t \cdot y_p)}{N}. \tag{3}$$

To address the unbalanced classes, which might result in bias towards the majority class, class weights[6] were defined as $w_{y_t} = |\mathrm{samples}|/(2 \cdot |y_t|)$. Thus, the model might prioritize misclassifying instances from the minority class over adjusting predictions from the majority class within the margin.

In an evolving timeline scenario, training is conducted incrementally in each time slot. The model of the first slot is initialized with random parameters. Then, in each slot, we progressively added the new shared content and their users, and conducted a two-epoch training of the model of the previous slot using Adam optimizer. Class weights were recalculated in each slot based on the updated class distribution.

### D. Model Complexity

The computational complexity analysis for processing a user using the proposed model assumes fixed-size operations have a complexity of $\mathcal{O}(1)$, following a similar assumption as Qureshi et al. [37] used for XGBoost. In this regard, the complexity of a linear layer depends on the dimensions of the input ($I$)

[6]Class weights followed the balanced sklearn strategy: https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html.

and output ($O$), resulting in a complexity of $\mathcal{O}(IO)$ due to $O$ scalar products in $\mathbb{R}^I$. Then, as $I$ and $O$ remain constant regardless of the analyzed user, the complexity is bounded. The computational cost of the GCN Eq. (1) is primarily determined by $\hat{D}^{-(1/2)}\hat{A}\hat{D}^{-(1/2)}X$, as the weight matrix multiplication and the bias addition are fixed. For a graph with $N$ nodes, where $A$ and $D$ are in $\mathbb{R}^{N\times N}$ and $X$ in $\mathbb{R}^{N\times F}$, the cost of the matrix multiplications is $\mathcal{O}\left(N^2+N^2+NF\right)=\mathcal{O}\left(N^2\right)$, as $F$ is constant because it represents the features of the node. As $\hat{D}^{-(1/2)}$ is a diagonal matrix, the multiplication $\hat{D}^{-(1/2)}\hat{A}$ can be computed in $\mathcal{O}\left(N^2\right)$, with the number of features $F$ constant. The cost of attentional aggregation is $\mathcal{O}(N)$, as it involves computing the weight for the $N$ nodes, multiplying them and summing the weighted nodes. Finally, the GRU has a computational complexity of $\mathcal{O}(L)$, where $L$ is the sequence length, assuming constant operations for fixed dimensions.

The user features component is forwarded directly to the concatenation component, hence its complexity is $\mathcal{O}(1)$. Considering $U$ the number of users in the user graph, the cost of processing the user–user graph is $\mathcal{O}\left(U^2\right)$ for each GCN and $\mathcal{O}(U)$ for the attentional aggregation. Hence, the total complexity of this component is $\mathcal{O}\left(3U^2+U\right)=\mathcal{O}\left(U^2\right)$. For the cost of processing the conversation propagation trees, let $T$ be the number of conversation trees for the user and $P$ the number of posts in the largest user tree. The complexity of processing all conversation trees is $\mathcal{O}\left(TP^2+TP+T\right)=\mathcal{O}\left(TP^2\right)$, including the cost of running the GCN and the attentional aggregation $T$ times and the GRU. Finally, concatenation and MLP operations have a complexity of $\mathcal{O}(1)$. As a result, combining all components, the model's total computational complexity for is $\mathcal{O}\left(U^2+TP^2\right)$.

## IV. Experimental Settings

To evaluate the performance of the proposed model, we conducted extensive experiments to address the following Research Questions:

1) *RQ1:* How does the proposed model compare with other state-of-the-art techniques for predicting the future spreading behavior of unseen users?
2) *RQ2:* How do the key components of the proposed model contribute to its performance?
3) *RQ3:* How does the proposed model perform when compared with other state-of-the-art techniques for the early detection of spreaders?

### A. Data Collection

Evaluation was based on the COVID-related misinformation dataset, Fibvid [22],[7] which consists of news claims from Politifact and Snopes. Each claim was searched on *Twitter* to retrieve its associated content.[8] The collection comprises 772 COVID-related news claims and 161 838 related tweets shared during 2020. Based on the labels assigned by Politifact and Snopes, 26% were labelled authentic and 74% were labelled fake. Tweets' labels were used to identify fake news spreaders. To achieve this, we computed a user score based on the proportion of shared tweets linked with fake news. Users were classified as spreaders if the proportion of shared fake content exceeded a threshold. We adopted a conservative definition of spreaders, setting the threshold to 0.5.

From the retrieved tweets, we defined both tweet and user graphs. In the tweet graph, nodes represented tweets, and directed edges the potential tweet actions (reply, quote, and retweet) following the flow of conversations [33], [34], [41]. Specifically, an edge from tweet $A$ to tweet $B$ indicated that $A$ was a reply/quote of $B$, while an edge from $A$ to $B$ indicated that $B$ was $A$'s retweet. The user graph, derived from the tweet graph, formed a conversational network where nodes represent users, and edges whether a user retweeted/replied/quoted/mentioned another. Similar to the tweet graph, edges follow the conversational flow and were weighted based on the number of interactions (e.g., mentions between two users). Unlike structural networks (i.e., based on the *X* follower/followee interactions or on *Facebook* Friends), which usually focus on reciprocal and long-term connections [18], conversational networks prioritize interactions related to the shared and consumed content, independently of whether users follow or are friends with each other. Furthermore, as users can engage with others outside their follower/followee network, the interaction network yields a broader and more diversified set of interactions.

To ensure that users actively engaged in social interactions as both sources and targets of tweet actions, we chose users belonging to the largest connected component (LCC) of the user graphs, provided they shared more than one tweet. The final collection comprised 24 430 users and over 200 k tweets. With a threshold of 0.5, group size was unbalanced, with 35% of users being spreaders and the remaining 65% nonspreaders. Notably, over 60% of users in each group exclusively shared either fake or authentic content. On average, users interacted with 4 ($\pm$26) unique users and shared 4 ($\pm$16) tweets. Nonspreaders tended to interact with more users than spreaders, while spreaders shared more tweets than nonspreaders. Significant differences were noted in the size and height of conversation trees, with nonspreaders' trees typically being taller and larger. Finally, differences were also observed in different psycho-linguistic markers (e.g., LIWC categories), the usage of punctuation and the presence of emotions.

### B. Baselines

The performance of the proposed model was evaluated against several baselines. First, simple baselines based on traditionally used features:[9]

1) *Tweet stats* (e.g., the percentage of tweets with URLs, mentions or hashtags, the percentage of tweets shared during the night and weekend, average tweet length, and average number of replies/retweets/favourites) as used in [10], [25].

[7]The collection is available at https://doi.org/10.5281/zenodo.4441377.
[8]Tweets were retrieved using: https://github.com/knife982000/FakingIt.
[9]The definition of each feature set, and additional feature sets can be found in the companion repository.

2) *User stats* (e.g., follower/friend ratio, follower/friend count, description/screen name length, account age, and account verification status) as used in [40].
3) *LIWC categories* (a subset of LIWC linguistic categories associated with personal pronouns, personal concerns, and cognitive/informal/perceptual processes) as used in [12], [13].
4) *Personality* (scores for the Big 5 traits and personality disorders like suspicious, emotional, and anxious, computed based on [32]) as used in [12].
5) *Punctuation* (e.g., proportion of commas, dots, semicolons, ellipsis, and exclamation/question marks) as used in [11], [12], [28].
6) *Readability* (metrics estimating the text complexity and the literacy level needed to understand it) as used in [12], [45].
7) *All features* (the combination of features that was used in the user feature component).
8) *Tree metrics* (features describing the conversation trees in terms of structure and semantic similarity between the shared posts, inspired in [15], [54]; these features served as input of the GCNs in the social interaction component).
9) *Content evolution* (the average semantic similarity between the first and last user posts, inspired in [4] and to account for variations in the shared content; these features served as input of the GCNs in the social interaction component).
10) *Node2vec* (embedding representation of users based on the user graph [17]).
11) *Content-based* (a content-based representation of the shared content based on pretrained word2vec, GloVe, BERT or SentenceTransformers).

Various classifiers were trained for each feature set, including decision trees, random forests, K-NN, SVM, gradient boosting, Naïve Bayes, and logistic regression models.

We also considered the closely related works of:[10]

1) Sharma and Sharma [44], based on hand-crafted user, tweet and psycho-linguistic features, and word2vec embeddings processed through a neural network.
2) Sansonetti et al. [40], based on user and sentiment features processed through a neural network.
3) *CheckerOrSpreader* [12], based on psycho-linguistic features and BERT embeddings processed by a neural network.
4) Shrestha and Spezzano [45], based on hand-crafted user and personality features (excluding demographic features) processed through a traditional classifier.
5) *FacTweet* [10], based on psycho-linguistic features and GloVe embeddings processed through a neural network.
6) Hamdi et al. [19], based on hand-crafted user and node2vec features processed through traditional classifiers.
7) Kumar et al. [24], based on an autoencoder representation of text embeddings and traditional classifiers.
8) Lampridis et al. [25], based on hand-crafted and psycho-linguistic features processed through a traditional classifier.
9) Agarwal et al. [1], based on hand-crafted and psycho-linguistic features and GloVe embeddings (excluding click-bait features) processed through a traditional classifier.
10) Siino et al. [48], based on processing the shared content through a shallow neural network.
11) Tommasel et al. [50], based on combining user and content (through a naïve conversation summarization) features.

When available, we used the original implementations of these baselines. Parameters were optimized following the procedures outlined in the original studies. We report the best configuration for each baseline based on a pairwise ranking using the selected evaluation metrics.[11] Minimal preprocessing was applied to tweets, including URL replacement and the removal of symbols and numbers.

### C. Evaluation Partitions and Metrics

Evaluation was carried out in an offline setting using temporal splits, simulating a scenario where the model learns from historical data to predict current or future behaviors. Two evaluation scenarios were defined to account for the evolving nature of social media dynamics, illustrated in Fig. 2.

*1) Predicting the Future Behavior of Unseen Users:* Here, we aim to assess how the proposed model generalizes to the activities of unseen users. To divide users into training and test sets, they were sorted considering the date of their first interaction (reply, mention, quote, retweet). The first 70% users were selected as training and the last 30% formed the test set. User representations are computed for the entire evaluation timeframe, ensuring no overlap between training and test sets. This scenario corresponds to performing a single update/inference pass of the model depicted in Fig. 1.

*2) Early Detection of Unseen Users:* In this scenario, we dynamically compute model's inputs in a time-evolving manner across consecutive time slots, performing an update/inference pass of the model in Fig. 1 for each slot. Shared tweets were sorted by date and divided into ten sequential slots based on their quantile distribution. Unlike the previous scenario where user representations considered the entire conversation propagation and interaction networks, here we create the user representation per slot, based on the cumulative content shared up to that point, allowing the modelling of the temporal dynamics of interactions. User presence in each slot is not guaranteed, enabling the analysis of interaction sequences and interval dynamics. User class is defined per slot, allowing the user class to change across slots, which adds more dynamism to the analysis. The training set remains consistent with the previous scenario, while the test set comprises users who shared content in the slot, but are not included in the training set. Here, users outside the LCC sharing content in the corresponding slot are also

[10]See Section II and the companion repository for more details.

[11]Implemented using the Ranky package: https://pypi.org/project/ranky/.

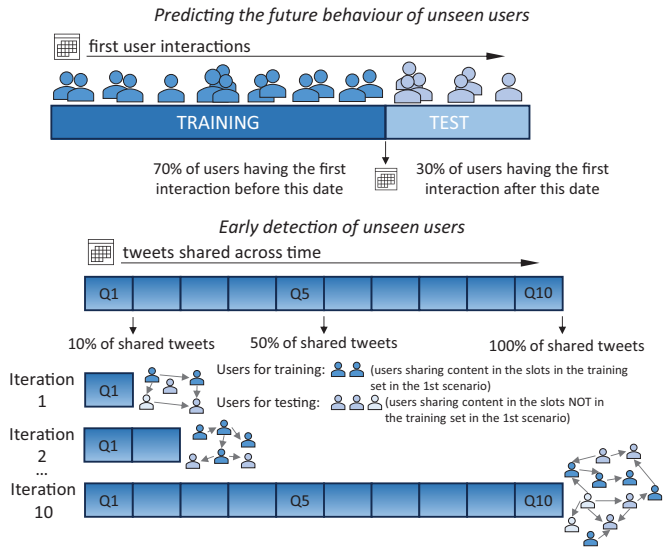


Fig. 2. Schematic representation of the evaluation scenarios.

included in the test set. This approach allows for the inclusion of diverse user types with varying amounts of shared content and interactions, simulating a real-world scenario where new users continuously emerge. As in the previous scenario, there is no overlapping between training and test users.

All evaluations used the same corresponding data partitions. Evaluation was based on binary (focusing on the spreader class) and weighted classification metrics.[12] Due to the task's nature, recall may be more relevant than precision. Finally, the Matthews coefficient [5], deemed more reliable than AUC-ROC in unbalanced data scenarios, was also considered.

### D. Implementation Details

All models, including baselines and the proposed model, were implemented in Python, with the support of sklearn, TensorFlow and PyTorch. The Adam optimizer was used, with a learning rate of $1e-3$, and parameters $beta_1 = 0.9$ and $beta_2 = 0.999$. Hyperparameter optimization focused on adjusting the size of the GCNs, the attentional aggregation mechanism, and the dense layers.[13] The size of the user GCNs was set to 32 (with evaluations for 32, 70, 100). The GRU size matched that of the GCNs to avoid additional parameters, while the attentional aggregation mechanism used a two-layer MLP with 32 dimensions in each layer (with evaluations for one-layer MLP, 70 and 100 dimensions). The size of the last two dense layers was set to 70 (with evaluations for 32, 70, and 100). Loss function evaluations included unweighted hinge, weighted and unweighted cross entropy, and treating spreader detection as a regression problem, considering MAE and MSE as loss functions. Different activation functions were tested, including ReLU, linear and leaky-ReLU, along with combinations of dropout and batch normalization layers. Training stopped when no loss changes were observed, typically converging after 20 epochs. The model was trained on an Asus Scar 15 with a Ryzen 9 5900HX and a NVidia GeForce RTX 3080. Training and evaluation over the entire timeframe took approximately 2.5 minutes and 20 seconds per epoch.

## V. Evaluation

We present the best results for each selected baseline, with the top scores in bold and the second-best underlined. Alongside classification metrics, we performed paired statistical analysis (with $\alpha = 0.01$)[14] on the obtained results for each method to determine the significance of observed differences.

### A. RQ1. Comparison With State-of-the-Art Techniques for Predicting the Future Behavior of Unseen Users

Table I summarizes the results for both the proposed model and the chosen baselines.[15] Siino et al. [48] is omitted due to consistently negligible results across all runs. As observed, *tweet stats* significantly outperformed *user stats*. *Tweet stats* also surpassed all *content-based* alternatives, suggesting their efficacy in identifying spreaders over the analysis of the entire shared content. This situation might be related to the semantic homogeneity of the data, where distinguishing subtle differences between a large number of fake and authentic content becomes challenging. Nonetheless, *content-based* embeddings performed significantly better than individual psycho-linguistic features such as *LIWC categories* and *personality*, hinting the similarities in fake and authentic content. *Content evolution* achieved higher results than *content-based*. Following the topically focused nature of the data collection, content analysis across time might allow detecting subtle style and expression changes that could contribute to spreader detection [4], [10], [35]. *Tree metrics* achieved high recall, reinforcing the importance of conversation structures in detection [15], [54], improving both psycho-linguistic and *content-based* features, and social interaction topology (as represented by *node2vec*). Although *node2vec* improved some of the other alternatives (e.g., *personality*, *LIWC*), the sparse social structure may limit its effectiveness as a spreader indicator. Finally, *all features* (as used in the user feature component of our proposed model) showcased the complementary nature of features and their combined ability to achieve generally higher results than when used individually.

State-of-the-art baselines generally exhibited similar precision to traditional ones, but mostly lower recall. Hamdi et al. [19] and Sharma and Sharma [44] yielded the worst recall results. Conversely, Sansonetti et al. [40] and Lampridis et al. [25] achieved the best precision/recall balance. In all cases, techniques were based on combinations of hand-crafted and psycho-linguistic features, and content or network embeddings. These findings emphasize the importance of selecting and combining features, with deep learning models not always being

[12]All metrics were implemented by scikit-learn: https://scikit-learn.org/stable/modules/model_evaluation.html.

[13]The results for the hyper-parameter optimization can be found in the companion repository.

[14]To counteract the problem of multiple comparisons, we used the Bonferroni correction, and Cohen's d to quantify the magnitude of the effects.

[15]The full set of results and more evaluation metrics can be found in the companion repository.

TABLE I
BEST PERFORMANCE COMPARISON FOR FAKE NEWS SPREADERS DETECTION (RQ1)

| | Binary (Spreader Class) | | Weighted | | Matthews |
|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Coeff |
| Our proposal | 0.84 | **0.905** | **0.861** | **0.859** | **0.719** |
| tweet stats | **0.923** | 0.77 | 0.849 | 0.833 | 0.68 |
| user stats | 0.67 | 0.304 | 0.583 | 0.521 | 0.123 |
| LIWC categories | 0.678 | 0.229 | 0.583 | 0.501 | 0.109 |
| personality | 0.598 | 0.32 | 0.533 | 0.493 | 0.042 |
| punctuation | 0.787 | 0.484 | 0.685 | 0.633 | 0.325 |
| readability | 0.734 | 0.349 | 0.631 | 0.559 | 0.205 |
| sentiment | 0.505 | 0.137 | 0.469 | 0.435 | −0.053 |
| all features | 0.852 | 0.636 | 0.761 | 0.731 | 0.492 |
| tree metrics | 0.745 | 0.797 | 0.729 | 0.731 | 0.447 |
| content evolution | 0.838 | 0.655 | 0.756 | 0.733 | 0.488 |
| node2vec | 0.657 | 0.381 | 0.579 | 0.537 | 0.128 |
| content-based | 0.8 | 0.584 | 0.715 | 0.687 | 0.404 |
| Sharma and Sharma [44] | 0.648 | 0.081 | 0.558 | 0.454 | 0.045 |
| Sansonetti et al. [40] | 0.54 | 0.832 | 0.412 | 0.502 | −0.144 |
| *FacTweet* [10] | 0.82 | 0.527 | 0.757 | 0.691 | 0.456 |
| Shrestha and Spezzano [45] | 0.779 | 0.333 | 0.659 | 0.569 | 0.243 |
| *CheckerOrSpreader* [12] | 0.505 | 0.164 | 0.472 | 0.444 | −0.05 |
| Hamdi et al. [19] | 0.831 | 0.064 | 0.664 | 0.462 | 0.114 |
| Kumar et al. [24] | 0.719 | 0.237 | 0.613 | 0.523 | 0.154 |
| Lampridis et al. [25] | 0.891 | 0.647 | 0.793 | 0.759 | 0.554 |
| Agarwal et al. [1] | 0.794 | 0.585 | 0.708 | 0.679 | 0.388 |
| Tommasel et al. [50] | 0.80 | 0.848 | 0.838 | 0.821 | 0.675 |

the optimal choice [24], [48]. Despite Sansonetti et al. [40] achieving high binary recall, its weighted metrics ranked among the lowest, with no statistically significant improvements observed. This was also reflected by the negative Matthews score, indicating an unbalanced confidence in correctly classifying elements of both classes. *FacTweet* [10] (including temporal information) and Tommasel et al. [50] achieved a good balance of precision and recall with significant improvements over traditional and state-of-the-art baselines.

Our model achieved the highest results, with average relative differences of 218% (±315%) and 48% (±28%) for binary and weighted recall and 323% (±427%) for Matthews score.[16] On average, the largest recall improvements were observed compared to state-of-the-art baselines. In most cases, observed differences were statistically significant, favouring our model. The differences in binary and weighted precision/recall were generally higher for baselines than for our proposed model, with the largest differences observed for state-of-the-art baselines (e.g., [19], [40], [44]). These findings suggest that the nonspreader class might impact baselines more than our proposed model, indicating our model's ability to predict both classes with comparable accuracy.

We also evaluated the proposed model under varying levels of available information for inference.[17] Based on the time slot division outlined in Section IV-C for users in the test set, our model consistently outperformed baselines in terms of recall and Matthews score across most time slots. While most baselines showed no significant changes in recall as more information became available, there were notable shifts in Matthews score. This indicated that despite binary recall not improving, the overall prediction quality for both classes enhanced, suggesting a stronger generalization capability.

[16]These reported values represent relative improvements over very low baseline scores, which result in large percentage increases. Also, the variations in ± values arise from the high performance variability of baseline models.

[17]The full set of results can be found in the companion repository.

*1) Evaluating Other Thresholds:* We evaluated two additional thresholds, 0.3 and 0.7. At 0.3, the spreader class included more users, even those sharing a small proportion of fake news, leading to a more distinct nonspreader class primarily composed of users solely sharing factual content. In contrast, at 0.7, the nonspreader class became more heterogeneous, including users who occasionally shared fake news, while the spreader class became more exclusive, primarily comprising users with a higher prevalence of fake news. Despite these changes, observed trends aligned with Table I,[18] where the proposed model consistently outperformed others, followed by *tweet stats*, Tommasel et al. [50], and Lampridis et al. [25].

For the 0.3 threshold, precision and recall generally increased compared to 0.5, but the Matthews score decreased, reflecting the added complexity of the task. Simpler models improved precision by reducing false positives, while state-of-the-art models improved recall by reducing false negatives. Conversely, the 0.7 threshold led to a decline in recall (when compared to 0.5) for approaches using content or psycho-linguistic features with traditional classifiers. Interestingly, deep learning models using the same features saw improved recall at this higher threshold. Overall, simpler models performed better in terms of precision at lower thresholds, but as the threshold increased, state-of-the-art techniques achieved superior precision. This highlights the limitations of simpler models in adapting to ambiguous class definitions (from a heterogeneous spreader class to a heterogeneous nonspreader class), while advanced models like neural networks effectively handled these nuances. Finally, these results confirm that the 0.5 threshold provides balanced classification performance, allowing models to effectively leverage a mix of features.

*2) Analysis of Mistaken Classifications:* To analyze classification mistakes, the differences between users in the training and test partitions were examined based on user features, social interaction features, and shared content. The analysis focused on the significance of feature distribution differences across various user subsets (e.g., correctly vs. incorrectly classified user, train set spreaders vs. test set spreaders). Additionally, the atypicality of test users was assessed using the Jensen-Shannon divergence [53] between their characteristics and the prototype (i.e., average) characteristics of train instances within the same class. Since Jensen-Shannon divergence requires probabilistic distributions, feature scores for each user were normalized.

Statistically significant differences were observed when comparing spreader and nonspreader users across train and test sets for various feature sets. Differences between the two classes were less frequent in the train than in the test set, and differences between spreaders in the train and test sets were less frequent than those between nonspreaders. This suggests that nonspreaders represent a more heterogeneous group, making them harder to model. These findings indicate the presence of a data distribution shift, a common occurrence in real-world scenarios, particularly when datasets are collected across different periods or contexts, such as during the evolving phases of a crisis like COVID-19.

[18]The results table is available in the companion repository.

The proposed model achieved an accuracy of 86%, with 9% false positives and 5% false negatives. This indicates that the model performed better in classifying spreaders, which exhibited less variability in their characteristics. Analysis of the Jensen–Shannon divergence across feature groups revealed differences for both types of errors (correctly vs. incorrectly classified spreaders and nonspreaders) for tree metrics, and user and tweet stats features. Additionally, spreaders showed differences in readability and punctuation, while nonspreaders exhibited differences in content evolution features. These findings highlight variations for the user and social interaction feature components, though all effect sizes were small. Model feature importance analysis using SmoothGrad [49] revealed that such feature sets were among the most important, which might help explain the misclassifications.[19]

Regarding shared content, significant differences were observed between correctly classified spreaders and nonspreaders. However, no significant differences were found across train/test sets for each class or between correctly/incorrectly classified users. As previously mentioned, this may be due to the dataset's topical focus, where the main subject is consistent across tweets, potentially reducing the distinguishability of differences. Additionally, the impact of each individual BERT feature on the model's output was two orders of magnitude lower than that of the other features, suggesting that content drift is less likely to substantially affect model performance compared to shifts in other feature types.

Lastly, when analyzing interaction sources (mentions, replies, mentions, and retweets), both spreaders and nonspreaders showed significant differences in the frequency of replies, while only nonspreaders showed differences in mentions. Divergence in interactions also revealed differences between correctly and incorrectly classified instances for both classes. Nonetheless, these features were shown to be less important than those in the user feature component.

Despite the observed differences, the proposed model exhibited robust performance, effectively learning generalizable patterns for both spreaders and nonspreaders. This highlights the model's ability to handle data and feature shifts, even when the differences involve features that significantly influence predictions, demonstrating the effectiveness of the defined components in capturing essential classification characteristics.

The evaluation showed the potential of our proposed model for improving fake news spreader detection compared to traditional and state-of-the-art baselines. Our model not only demonstrated strong recall for identifying spreaders but also maintained a balanced prediction across both classes. Additionally, its performance remained robust across varying thresholds, effectively handling the tradeoffs between class inclusivity and specificity.

### B. RQ2. Ablation Study of the Proposed Model

To assess the effectiveness and contribution of each component in our proposed model, we conducted an ablation study with the following variants:[20]

1) *SameFeatures-SocialInteraction*: The user feature component receives the same features as the social interaction component (i.e., *tree metrics*).
2) *SameFeatures-UserFeature*: GCNs receive the features of the user feature component (i.e., *all features*).
3) *SameFeatures-CombinedFeatures*: Both the user feature and the social interaction component receive the same set of combined features (i.e., *all features* and *tree metrics*).
4) *UserFeatures-no-user-features*: The user features component is removed.
5) *SocialInteraction-no-user-interactions*: The social interaction processing component is removed.
6) *FeatureSelection - UserFeatures*: Feature selection was performed on the user features component[21] based on correlation analysis. For feature groups with a correlation above $0.8$, only the feature with the largest variance was retained.
7) *FeatureSelection - SocialInteraction*: Similar to the previous case, but performing feature selection over the social interaction component features.
8) *FeatureSelection - UserFeatures + SocialInteraction*: Feature selection is applied to features of both the user features and social interaction components.
9) *Conversation-no-conversation*: The conversation processing component is removed.
10) *Conversation-BERT-last*: Instead of using pooled embeddings we used the embedding of the last token.
11) *Conversation-SentenceTransformer*: Instead of using the pooled BERT embeddings for post representation, we used the SentenceTransformer model.[22]
12) *Conversation-no-GRU*: The GRU was removed, so the temporal sequence of conversations is disregarded. Max pooling was used for merging the trees' representation.
13) *Conversation-no-Attention*: The attentional aggregation was removed, and conversations were represented by that of their root. By removing the aggregation, the root post gained more importance than the others.
14) *Training-Hinge-No-weight*: Class weights were not added to the Hinge function used in training.
15) *Training-Hinge-Squared*: The loss function was modified to: $\mathcal{L}(y_t, y_p) = \max(1 - y_t \times y_p, 0)^2$. This amplifies the penalty of misclassifications and outliers, forcing the model to learn a stronger decision boundary.
16) *Training-BCE (Binary Cross Entropy)*: The loss function was defined as $\mathcal{L}(y_t, y_p) = -(y_t \times \log(\sigma(y_p)) + y_t \times \log(1 - \sigma(y_p)))$, where $y_t \in \{0, 1\}$ and $\sigma(y_p)$ is the sigmoid function of $y_p$.

[19] SmoothGrad can be used to enhance model interpretability by estimating feature importance and highlighting which inputs most influence the model's predictions. Charts depicting feature importance distributions across different user groups (e.g., spreaders vs. nonspreaders, correctly vs. incorrectly classified instances) are available in the companion repository.

[20] More variants including hyperparameter modifications can be found in the companion repository.

[21] Feature selection was implemented using https://feature-engine.trainindata.com/.

[22] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

TABLE II
ABLATION STUDY OF THE PROPOSED MODEL (RQ2)

| | Binary (Spreader Class) | | Weighted | | Matthews |
|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Coeff |
| Our Full proposal | 0.84 | **0.905** | **0.861** | **0.859** | **0.719** |
| *SameFeatures SocialInteraction* | 0.81 | 0.859 | 0.82 | 0.819 | 0.638 |
| *SameFeatures UserFeature* | 0.865 | 0.645 | 0.783 | 0.76 | 0.545 |
| *SameFeatures CombinedFeatures* | 0.857 | 0.874 | 0.857 | 0.857 | 0.713 |
| *UserFeatures No-user-features* | 0.814 | 0.807 | 0.801 | 0.801 | 0.601 |
| *SocialInteraction No-user-interactions* | 0.853 | 0.826 | 0.834 | 0.834 | 0.667 |
| *Feature Selection UserFeatures* | 0.859 | 0.828 | 0.839 | 0.838 | 0.677 |
| *Feature Selection SocialInteration* | 0.849 | 0.886 | 0.858 | 0.857 | 0.715 |
| *Feature Selection UserFeatures + SocialInteraction* | 0.845 | 0.856 | 0.854 | 0.854 | 0.707 |
| *Conversation No-conversation* | 0.851 | 0.867 | 0.85 | 0.85 | 0.699 |
| *Conversation BERT-last* | **0.867** | 0.755 | 0.817 | 0.81 | 0.628 |
| *Conversation SentenceTransformer* | 0.851 | 0.793 | 0.82 | 0.818 | 0.638 |
| *Conversation No-GRU* | 0.853 | 0.863 | 0.85 | 0.85 | 0.699 |
| *Conversation No-Attention* | 0.851 | 0.85 | 0.843 | 0.843 | 0.686 |
| *Training Hinge-No-weight* | 0.856 | 0.853 | 0.847 | 0.847 | 0.693 |
| *Training Hinge-Squared* | 0.857 | 0.835 | 0.84 | 0.84 | 0.679 |
| *Training BCE* | 0.852 | 0.861 | 0.848 | 0.848 | 0.696 |

New models were trained from scratch for each evaluation to predict the future behavior of unseen users. Table II outlines the results for the described variations. The full model achieved the best results for almost all reported metrics, except for binary precision. Statistical analysis revealed that differences favouring the full model were significant in all cases, except for *SameFeatures-CombinedFeatures*. Removing any of three model components significantly reduced its performance. The largest significant effects were observed when either the user feature (*UserFeatures-No-user-features*) or the social interactions component (*SocialInteraction-No-user-interactions*) were removed. This highlights the complementary nature and importance of the components to the task.

Regarding the variations of the conversation propagation component, changing the embedding model (*Conversation-SentenceTransformer*) or how embeddings are computed (*Conversation-BERT-last*) had a larger effect than removing any component. This suggests that the quality of content representation significantly influences the model. Also, removing the attentional aggregation (*Conversation-no-Attention*) had a larger effect than removing the GRU (*Conversation-no-GRU*), suggesting that the contextual content of conversations might be more important than their temporal sequence.

Combining feature sets also had a significant effect on performance. *SameFeatures-SocialInteraction* outperformed *SameFeatures-UserFeature*, which yielded the lowest binary recall. However, *SameFeatures-SocialInteraction* was worse than *SameFeatures-CombinedFeatures*, emphasizing the importance of diverse and complementary conversation aspects. This reinforces the importance of selecting the features to integrate in the model. Further evaluations are needed to continue exploring the role of simple hand-crafted features and whether they could replace more computationally intensive model components without compromising its performance.

Feature selection had no significant impact on results. However, reducing the correlation threshold to 0.7 (resulting in the removal of more features) led to medium to large significant differences, favoring the use of all features. These findings suggest that neural network-based models effectively adapt to input features, learning their relative importance and handling their potential redundancy.

Varying the loss function did not result in significant variation.[23] However, using *Training-Hinge-No-weight* tended to favor the majority class (nonspreaders), resulting in a lower recall. Similarly, *Training-BCE* and *Training-Hinge-squared* exhibited a higher number of false negatives, which lead to a lower recall. These observations, combined with its prior use in a similar task (e.g., [14]), confirm the suitability of the hinge function for this application.

Results highlighted the significant contribution of each component to the proposed model's performance. Nonetheless, further studies into the relevance and interaction of userfeatures are needed.

### C. RQ3. Comparison With State-of-the-Art Techniques for the Early Detection of Spreaders

Table III presents the results for the early detection of unseen spreaders for each of the 10 time slots created following the second scenario in Section IV-C. For each baseline, a new model was trained for every slot based on the cumulative content shared up to that moment. Sansonetti et al. [40] is omitted as it achieved perfect recall at the expense of classifying every user as a spreader (extremely low precision).

In the initial time slots (Q1 to Q3), binary recall remained low, accompanied by even lower binary precision, indicating the challenge of identifying spreaders with limited available information. Regarding the traditional baselines, *content-based* achieved high binary recall in Q1 and Q2, followed by several psycho-linguistic features and their combinations. While *node2vec* performed well in Q2, it showed lower performance in Q1 and Q3, likely due to the sparse and limited interactions between users early on, for which adding only a few interactions (in the following slot) might significantly alter the social structures. Similarly, *tree metrics* and *content evolution* are expected to initially perform poorly due to the need for more interactions and content to effectively assess changes in

[23]Results using regression loss functions are omitted due to their lower performance compared to the reported ones. Full details can be found in the companion repository.

TABLE III
PERFORMANCE COMPARISON FOR BINARY RECALL ACROSS TIME SLOTS (RQ3)

| | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Our proposal | 0.143 | 0.185 | 0.135 | 0.415 | 0.149 | 0.356 | 0.636 | **0.858** | 0.795 | **0.816** |
| Tweet stats | 0.143 | 0.296 | 0.351 | 0.415 | 0.493 | 0.493 | 0.75 | 0.773 | 0.751 | 0.743 |
| User stats | 0.0 | 0.037 | 0.081 | 0.049 | 0.179 | 0.26 | 0.326 | 0.303 | 0.294 | 0.298 |
| LIWC categories | 0.143 | 0.0 | 0.054 | 0.073 | 0.194 | 0.274 | 0.335 | 0.303 | 0.273 | 0.26 |
| Personality | 0.286 | 0.111 | 0.081 | 0.049 | 0.299 | 0.356 | 0.297 | 0.308 | 0.327 | 0.327 |
| Punctuation | 0.143 | 0.0 | 0.0 | 0.0 | 0.418 | 0.452 | 0.469 | 0.417 | 0.414 | 0.435 |
| Readability | 0.143 | 0.111 | 0.108 | 0.122 | 0.313 | 0.329 | 0.394 | 0.368 | 0.363 | 0.347 |
| Sentiment | 0.143 | 0.074 | 0.054 | 0.024 | 0.119 | 0.123 | 0.177 | 0.171 | 0.164 | 0.164 |
| Tree metrics | 0.0 | 0.037 | 0.0 | 0.049 | 0.03 | 0.397 | **0.756** | 0.794 | **0.804** | 0.792 |
| Content evolution | 0.0 | 0.037 | 0.108 | 0.122 | 0.224 | 0.233 | 0.68 | 0.522 | 0.605 | 0.656 |
| node2vec | 0.0 | 0.444 | 0.054 | 0.293 | 0.045 | 0.0 | 0.43 | 0.347 | 0.207 | 0.168 |
| All features | 0.143 | 0.222 | 0.162 | 0.22 | 0.373 | 0.438 | 0.638 | 0.611 | 0.625 | 0.626 |
| Content-based | 0.225 | 0.276 | 0.106 | 0.12 | 0.543 | 0.361 | 0.402 | 0.204 | 0.255 | 0.297 |
| Sharma and Sharma [44] | 0.0 | 0.095 | 0.231 | 0.0 | 0.069 | 0.2 | 0.358 | 0.403 | 0.49 | 0.322 |
| *FacTweet* [10] | 0.0 | 0.0 | 0.024 | 0.307 | 0.367 | 0.456 | 0.107 | 0.46 | 0.474 | 0.454 |
| Shrestha and Spezzano [45] | 0.143 | 0.037 | 0.135 | 0.073 | 0.284 | 0.274 | 0.37 | 0.338 | 0.346 | 0.333 |
| *CheckerOrSpreader* [12] | 0.134 | 0.156 | 0.151 | 0.126 | 0.178 | 0.343 | 0.564 | 0.384 | 0.546 | 0.269 |
| Hamdi et al. [19] | 0.0 | 0.185 | 0.081 | 0.024 | 0.0 | 0.041 | 0.044 | 0.519 | 0.127 | 0.187 |
| Kumar et al. [24] | 0.011 | 0.123 | 0.087 | 0.031 | 0.578 | 0.236 | 0.213 | 0.127 | 0.13 | 0.236 |
| Lampridis et al. [25] | 0.048 | 0.206 | 0.092 | 0.12 | **0.744** | 0.511 | 0.596 | 0.509 | 0.576 | 0.645 |
| Agarwal et al. [1] | **0.429** | **0.476** | **0.538** | **0.5** | 0.414 | **0.7** | 0.653 | 0.605 | 0.665 | 0.688 |
| Tommasel et al. [50] | 0.328 | 0.111 | 0.135 | 0.073 | 0.179 | 0.191 | 0.582 | 0.698 | 0.703 | 0.747 |

users' communication patterns. For these initial slots, traditional baselines generally outperformed state-of-the-art ones. For example, baselines such as *all features* and *content-based* achieved significantly better results compared to some state-of-the-art baselines. However, as more information becomes available (e.g., starting in Q7), state-of-the-art baselines began to show improved performance.

Our proposed model demonstrated high performance in weighted metrics, achieving high recall in both the initial and final slots. Although its binary recall in the first slots was lower compared to content-based baselines, our model achieved the highest binary recall in the final slot. While our approach worked better as more information was available, results showcase the adaptability of our proposed model to varying levels of information availability, and its ability to achieve balanced performance across both classes over time. Overall, our approach exhibited statistically significant differences favouring it over traditional and state-of-the-art baselines, with increasingly higher effect sizes as time slots passed. Significant differences were observed across several slots, even when compared to *tweet stats*, which also demonstrated high binary recall. On average, for the last slot, binary and weighted recall were improved by 149.28% ($\pm$ 119.21%) and 25% ($\pm$ 14.99%), respectively, registering higher improvements over the state-of-the-art baselines.

These results stressed the challenge of detecting scenarios with limited information and unbalanced class distribution. Interestingly, in such scenario, complex techniques (like Sharma et al. [43], *FacTweet* [10], or Giachanou et al. [12]) did not exhibit a significant advantage over simpler methods. This observation may be related to the fact that, for such first slots, content is the predominant information source, as the information that social interactions and conversation structures can provide is limited, thus not effectively contributing to spreader detection. Additionally, as conversations start to unfold (and social interactions start to appear), they might result in noisy and highly changing structures, possibly explaining the variable performance of *node2vec* and *tree metrics*.

Furthermore, as the spreader condition is computed based on slot-specific information, it may lead to fluctuations in users' spreader condition, contributing to model noise. Then, as slots progressed, more content is shared, conversations grow and interactions stabilize, spreaders might be able to blend in more effectively. Consequently, information sources other than content become increasingly relevant for spreader detection, leading to sustained improvements in performance of complex techniques compared to simpler ones.

Finally, following Ghanem et al. [10] and Plepi et al. [35], we assessed the model's predictive ability by defining users' spreader condition based on their entire behavior, rather than the slot-specific one. These adjustments reduce class imbalance and noise since users are assigned to the spreader or nonspreader class only once, regardless of subsequent behavior changes in specific slots.

Results[24] indicated higher binary recall from Q1, accompanied by a lower weighted recall, and more consistent predictions across all slots. This suggest that current activities offer insights into future spreader behavior, even with limited available information. *Tweet stats* remained among the top-performing baselines, exhibiting high and moderate binary and weighed recall, respectively. Our proposed model consistently achieved high binary recall and Matthews score across nearly every slot. Overall, statistically significant differences favouring our proposed model were observed in every slot compared to most baselines, including *tweet stats* and *tree metrics*. On average, binary and weighted recall showed improvements of

[24]Results can be found in the companion repository.

110.59% (± 141.89%) and 22.9% (± 15%) for Q1 and 96.39% (± 76.79%) and 25.13% (± 14.68%) for Q9, respectively. These results hint the capability of our proposed model to predict future user behavior based on limited information.

Our proposed model demonstrated competitive and balanced performance in early spreader detection scenarios, while consistently achieving good performance as more slots passed.

## VI. Conclusion

This study introduced a deep learning model for identifying fake news spreaders on social media. By combining user and content-based features, conversation sequences and propagation structures, and user interaction features, the proposed model outperformed traditional and state-of-the-art baselines. It effectively predicted future behavior and demonstrated competitive and balanced performance in early spreader detection. Findings reinforced the significance of integrating diverse aspects of user characterization to create a more robust spreader identification model. The proposed model is not intrinsically dependent on specific topics but rather leverages the evolution of conversations, user interactions, and linguistic expression patterns. As a result, it can be applied to diverse data collections regardless of their topical focus. This suggests that the approach could be extended to identify spreaders in various domains, such as other public crisis or political elections, where similar conversation patterns may emerge.

We acknowledge several limitations that need to be tackled in future works. First, spreader definition relies on individually assigned post labels, which may be incorrect, and lead to mislabeling users. Future studies should explore diverse data collections across various scales, domains, topics and spreader ratios to validate the model's performance robustness. Additionally, model's performance could be evaluated for detecting rumours, hoaxes or conspiracy spreaders. Second, the definition of the spreader condition relies on a threshold, creating a thin and arbitrary line to separate users. To address this, a new class could be introduced to characterize borderline users (e.g., "potential spreaders"), or the problem could be reframed into a regression task to estimate the likelihood of a user being a spreader. Third, considering the competitive performance of simple hand-crafted features, further evaluations are needed to determine their contribution and potential to replace more computationally complex models without compromising overall performance. Finally, extending the proposed model to offer explanations could enhance user comprehension of spreaders' impacts, increase decision awareness, and shade light on the motivational aspects behind spreading harmful content [21].

## Ethical Statement

Research relies on publicly available *Twitter* data initially collected and labelled by third parties. Personal user information is not included in the analysis, and user identities are not disclosed. Adhering to *Twitter*/*X*'s TOS, the shared data only includes user and tweet IDs, and aggregated content and activity features. This study aims to provide tools to identify users engaging in harmful behaviors within their social circles, thereby fostering decision awareness. Nonetheless, it is crucial to emphasize that the findings of this study should not be used to publicly criticize or stereotype identified spreaders. Finally, it is important to consider additional biases, including those originating from the data collection and tagging process, before applying any results from this study in real-world scenarios.

## References


[1] R. Agarwal, S. Gupta, and N. Chatterjee, "Profiling fake news spreaders on Twitter: A clickbait and linguistic feature based scheme," in *Proc. NLDB*. New York: Springer, 2022, pp. 345–357.

[2] L. Cabral, A. D. F. Martins, J. M. Monteiro, J. C. Machado, and W. Franco, "FakeSpreadersWhatsApp.BR: Misinformation spreaders detection in Brazilian Portuguese whatsapp messages," in *Proc. ICEIS,* vol. 1, 2023, pp. 219–228.

[3] E. D. V. Calderon, T. L. James, and P. B. Lowry, "How facebook's newsfeed algorithm shapes childhood vaccine hesitancy: An algorithmic fairness, accountability, and transparency (fat) perspective," *Data Inf. Manage.*, vol. 7, 2023, Art. no. 100042.

[4] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Antisocial behavior in online discussion communities," in *Proc. AAAI ICWSM*, vol. 9, 2015, pp. 61–70.

[5] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over f1 score and accuracy in binary classification evaluation," *BMC Genom.*, vol. 21, no. 1, pp. 1–13, 2020.

[6] T. Davidson, D. Bhattacharya, and I. Weber, "Racial bias in hate speech and abusive language detection datasets," 2019, *arXiv:1905.12516*.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[8] S. Fissi, E. Gori, and A. Romolini, "Social media government communication and stakeholder engagement in the era of covid-19: Evidence from Italy," *Int. J. Public Sect. Manag.*, vol. 35, no. 3, pp. 276–293, 2022.

[9] I. Freiling, N. M. Krause, D. A. Scheufele, and D. Brossard, "Believing and sharing misinformation, fact-checks, and accurate information on social media: The role of anxiety during covid-19," *New Media Soc.*, vol. 25, no. 1, pp. 141–162, 2023.

[10] B. Ghanem, S. P. Ponzetto, and P. Rosso, "Factweet: Profiling fake news twitter accounts," in *Proc. SLSP*. New York: Springer, 2020.

[11] A. Giachanou, E. A. Ríssola, B. Ghanem, F. Crestani, and P. Rosso, "The role of personality and linguistic patterns in discriminating between fake news spreaders and fact checkers," in *Proc. NLDB*. New York: Springer, 2020, pp. 181–192.

[12] A. Giachanou, B. Ghanem, E. A. Ríssola, P. Rosso, F. Crestani, and D. Oberski, "The impact of psycholinguistic patterns in discriminating between fake news spreaders and fact checkers," *Data Knowl. Eng.*, vol. 138, 2022, Art. no. 101960.

[13] A. Giachanou, B. Ghanem, and P. Rosso, "Detection of conspiracy propagators using psycho-linguistic characteristics" *J. Inf. Sci.*, vol. 49, no. 1, pp. 3–17, 2023.

[14] M. H. Goldani, R. Safabakhsh, and S. Momtazi, "Convolutional neural network with margin loss for fake news detection," *Inf. Process. Manage.*, vol. 58, no. 1, 2021, Art. no. 102418.

[15] V. Gómez, A. Kaltenbrunner, and V. López, "Statistical analysis of the social network and discussion threads in slashdot," in *Proc. 17th WWW*, 2008, pp. 645–654.

[16] S. Gong, R. O. Sinnott, J. Qi, and C. Paris, "Fake news detection through temporally evolving user interactions," in *Proc. PAKDD*. New York: Springer, 2023, pp. 137–148.

[17] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD*, 2016, pp. 855–864.

[18] I. Guy, *People Recommendation on Social Media*, Springer International Publishing, Cham, 2018, pp. 570–623.

[19] T. Hamdi, H. Slimi, I. Bounhas, and Y. Slimani, "A hybrid approach for fake news detection in twitter based on user features and graph embedding," in *Distrib. Comput. Internet Technol.* New York: Springer, 2020, pp. 266–280.

[20] B. Jiang, M. Karami, L. Cheng, T. Black, and H. Liu, "Mechanisms and attributes of echo chambers in social media," 2021, *arXiv:2106.05401*.

[21] M. Karami, T. H. Nazer, and H. Liu, "Profiling fake news spreaders on social media through psychological and motivational factors," in *Proc. 32nd ACM Conf. Hypertext Social Media*, 2021, pp. 225–230.
[22] J. Kim, J. Aum, S. Lee, Y. Jang, E. Park, and D. Choi, "FIBVID: Comprehensive fake news diffusion dataset during the covid-19 period," *Telemat. Inform.*, vol. 64, 2021, Art. no. 101688.
[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th ICLR*, 2017.
[24] G. Kumar, J. P. Singh, and A. K. Singh, "Autoencoder-based feature extraction for identifying hate speech spreaders in social media," *IEEE Trans. Comput. Soc. Syst.*, vol. 11, no. 4, pp. 4819–4827, Aug. 2024.
[25] O. Lampridis, D. Karanatsiou, and A. Vakali, "Manifesto: A human-centric explainable approach for fake news spreaders detection," *Computing*, 2022, pp. 1–23.
[26] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. New York: Springer, 2002.
[27] Y. Li, C. Gu, T. Dullien, O. Vinyals, and P. Kohli, "Graph matching networks for learning the similarity of graph structured objects," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2019, pp. 3835–3845.
[28] J. Ma, W. Gao, Z. Wei, Y. Lu, and K.-F. Wong, "Detect rumors using time series of social context information on microblogging websites," in *Proc. 24th ACM Int. CIKM*, 2015, pp. 1751–1754.
[29] J. Ma, W. Gao, and K.-F. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguist.* ACL, pp. 708–717.
[30] S. McKay and C. Tenove, "Disinformation as a threat to deliberative democracy," *Political Res. Quart.*, vol. 74, no. 3, pp. 703–717, 2021.
[31] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," 2019, *arXiv:1902.06673*.
[32] Y. Neuman and Y. Cohen, "A vectorial semantics approach to personality assessment," *Sci. Rep.*, vol. 4, no. 1, pp. 1–6, 2014.
[33] F. Pierri, C. Piccardi, and S. Ceri, "A multi-layer approach to disinformation detection in us and italian news spreading on twitter," *EPJ Data Sci.*, vol. 9, no. 1, 2020, Art. no. 35.
[34] F. Pierri, C. Piccardi, and S. Ceri, "Topology comparison of twitter diffusion networks effectively reveals misleading information," *Sci. Rep.*, vol. 10, no. 1, 2020, Art. no. 1372.
[35] J. Plepi, F. Sakketou, H.-J. Geiss, and L. Flek, "Temporal graph analysis of misinformation spreaders in social media," in *Proc. TextGraphs-16: Graph-Based Methods Natural Lang. Process.*, 2022, pp. 89–104.
[36] K. A. Qureshi, R. A. S. Malick, M. Sabih, and H. Cherifi, "Complex network and source inspired covid-19 fake news classification on twitter," *IEEE Access*, vol. 9, pp. 139636–139656, 2021.
[37] K. A. Qureshi, R. A. S. Malick, M. Sabih, and H. Cherifi, "Deception detection on social media: A source-based perspective," *Knowl. Based Syst.*, vol. 256, 2022, Art. no. 109649.
[38] B. Rath, A. Salecha, and J. Srivastava, "Detecting fake news spreaders in social networks using inductive representation learning," in *Proc. 2020 IEEE/ACM ASONAM*, 2020, pp. 182–189.
[39] L. Rosasco, E. De Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, 2004.
[40] G. Sansonetti, F. Gasparetti, G. D'aniello, and A. Micarelli, "Unreliable users detection in social media: Deep learning techniques for automatic detection," *IEEE Access*, vol. 8, pp. 154–167, 2020.
[41] J. Sanz-Cruzado and P. Castells, "Enhancing structural diversity in social networks by recommending weak ties," in *Proc. 12th ACM Conf. Recomm. Syst.*, NY, USA. ACM, 2018, pp. 233–241.
[42] C. Shao, G. L. Ciampaglia, O. Varol, A. Flammini, and F. Menczer, "The spread of fake news by social bots," 2017, *arXiv:1707.07592*.
[43] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, and Y. Liu, "Combating fake news: A survey on identification and mitigation techniques," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 3, pp. 1–42, 2019.
[44] S. Sharma and R. Sharma, "Identifying possible rumor spreaders on twitter: A weak supervised learning approach," in *Proc. 2021 IJCNN*. Piscataway, NJ, USA: IEEE Press, 2021, pp. 1–8.
[45] A. Shrestha and F. Spezzano, "Characterizing and predicting fake news spreaders in social networks," *Int. J. Data Sci. Anal.*, 2021, pp. 1–14.
[46] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newslett.*, vol. 19, no. 1, pp. 22–36, 2017.
[47] K. Shu, S. Wang, and H. Liu, "Beyond news contents: The role of social context for fake news detection," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 312–320.
[48] M. Siino, E. Di Nuovo, I. Tinnirello, and M. La Cascia, "Fake news spreaders detection: Sometimes attention is not all you need," *Information*, vol. 13, no. 9, 2022, Art. no. 426.
[49] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," 2017, *arXiv.1706.03825*.
[50] A. Tommasel, J. M. Rodriguez, and F. Menczer, "Following the trail of fake news spreaders in social media: A deep learning model," in *Adj Proc 30th ACM UMAP*, 2022, pp. 29–34.
[51] Y. Wu and B. Garrison, "Falsehood and satire on social media: Does partisan-motivated reasoning influence fake news sharing?" *Communication Res. Pract.*, 2023, pp. 1–19.
[52] K.-C. Yang et al., "The covid-19 infodemic: Twitter versus facebook," *Big Data Soc.*, vol. 8, no. 1, 2021.
[53] M. Yuksekgonul, L. Zhang, J. Zou, and C. Guestrin, "Beyond confidence: Reliable models should also quantify atypicality," in *Proc. ICLR 2023 Workshop Pitfalls Limited Data Comput. Trustworthy ML*, 2023.
[54] J. Zhang, C. Danescu-Niculescu-Mizil, C. Sauper, and S. J. Taylor, "Characterizing online public discussions through patterns of participant interactions," in *Proc. ACM Human-Comput. Interact.*, vol. 2, CSCW, 2018, pp. 1–27.

**Antonela Tommasel** (Member, IEEE) received the Ph.D. degree in computer sciences from Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina, in 2017.

She is a Researcher with the National Council of Scientific and Technological Research (CONICET), Argentina and a member of Instituto Superior de Ingeniería del Software de Tandil (ISISTAN) Research Institute, Tandil, Argentina. She is also an Assistant Professor with UNCPBA. Her research interests include recommender systems, applied NLP, user modeling, social computing, social media, user profiling, misinformation spread, bias in hate-speech, fairness in recommender systems, and applied LLMs.

**Juan Manuel Rodriguez** received the Ph.D. degree in computer science from the Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina, in 2012.

He was an Assistant Professor with UNCPBA, until 2024 and is a Researcher with National Council of Scientific and Technological Research (CONICET), Tandil, Argentina. Currently, he is an Assistant Professor with Aalborg University, Denmark, since 2024. His research interests include a variety of topics in artificial intelligence, machine learning, and information retrieval, with a strong focus on recommender systems, multimodal data processing, and social media analysis. His work includes leveraging deep learning for news recommendation, multimodal hate speech detection, and text-to-image retrieval.