Risky X Users Classification via Heterogeneous Graph Convolutional Network

Caliandro Rocco, Iacovazzi Antonio Raffaele
University of Bari Aldo Moro
r.caliandro6@studenti.uniba.it, a.iacovazzi6@studenti.uniba.it

July, 2024

Abstract

Social networks play a pivotal role in the evolution of today's society. With approximately 5.07 billion users worldwide, and an average daily usage time of 2 hours and 20 minutes (1), they have become a fundamental medium for expressing individuality, beliefs, and political ideas. Given their accessibility and popularity, social networks can also serve as powerful tools for spreading misinformation, inciting political hatred, and even organizing potential attacks, as evidenced by the events at the Capitol Hill in 2021. Among all social networks, *X* (formerly known as *Twitter*), with around 335.7 million of users (2), is one of the most widely used platforms for sharing political content, including content with malicious intent, by both ordinary users and prominent politicians[1]. Analyzing and understanding the contents and roles of each user could enable the prediction of when a cluster of individuals might engage in dangerous behavior. This involves identifying when a user actively spreads harmful content and analyzing their relationships with other potentially dangerous users to correctly classify them as a risky individual.

Several models have been developed for this kind of purposes like (4) or (5), but researchers are still striving to identify which models can effectively handle the complexity of data produced on social networks. These models should ideally integrate the multimedia context with the intricate relationships within the network.

This work, following the principle of the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology, has the aim of determine if a novel model based on Heterogeneous Graph Convolutional Networks

[1]For this reason Donald Trump was banned from *Twitter* in 2021 for his role in the Capitol Hills attacks of 06/01/2021 (3).

can effectively analyze the content and behavior of social network's users, potentially helping to achieve a safer online experience for everyone.

The results evidence interesting results that highlight the capabilities of such model in exploiting relationships in the social network, giving some insight for further researches.

1 Introduction: Business Understanding

In today's globalized world, social networks play a crucial role in connecting individuals, allowing them to share personal updates and express opinions. Given their significant role in modern society, it is fundamental to detect potentially dangerous users based on their shared contents and relationships to prevent the spread of misinformation, hate, and the formation of extremist political groups. Due to the underlying mechanisms that drive social networks' recommender systems, users who show interest in risky contents are increasingly exposed to similar contents and related users, contributing to their radicalization (6). Therefore, analyzing relationships is essential for accurately detecting a user's role.

Correctly identifying the role of an user as risky or safe, in a social network, is not a trivial task. Indeed there are borderline users who may share some risky content without being inherently dangerous, such as journalists or political experts. Furthermore, there are also silent users who engage in "lurking," the act of reading others' content without posting any personal content. This behavior is particularly dangerous when risky groups are the main source of information for these individuals, potentially leading them to adopt extremist ideas.

In both cases, a textual analysis may not accurately represent a user's role in the network. A safe borderline user might be detected as risky based on their textual content, while a dangerous silent user could be detected as safe due to their lack of produced risky content. Therefore, a comprehensive analysis that includes both content and relationships is necessary to accurately identify and address potential threats.

The objective of this work is to build a model capable of correctly identifying the previously unknown role (*safe* or *risky*) of a given user, based on the structure of the network and a set of already known users used as a training set.

To address the complex relationships within a network, a widely common approach is to use Graph Convolutional Networks (GCNs), an artificial neural network architecture that builds representations of each node by incorporating the representations of its neighbors [2]. This is particularly useful for tasks like node classification or link prediction. Given the network structure of social networks, it is straightforward to represent them as an organization of data through

[2] An useful and comprehensive introduction on GCNs can be found at (7).

various node components and relationships, thus forming a graph structure. For this reason, these architectures have become the prominent approach for social network-related tasks.

In this work, the ie-HGCN(8) is the chosen model for addressing our classification task, which will be thoroughly discussed in the Modeling section.

The criterion for success of this work is to achieve an improvement in the accuracy of node classification compared to plain text analysis, demonstrating that using the network structure as a feature in the model is crucial for this type of task.

The rest of the document is structured as follows: first, we present an analysis of the dataset to explore its characteristics; next, we discuss the preprocessing pipeline needed to feed the model. Following that, we delve into the properties and features of the ie-HGCN used as the architecture for training our model. Finally, we evaluate the obtained results and discuss the insights gained from this work and its potential future developments.

2 Data Understanding

Through social networks, a vast network of users can post text, images, and videos. However, on X, which is the focus of this study, the primary content is expressed as text, for this reason the dataset used take into account only the text of the involved users.

In the dataset object of our studies, a small portion of the X network is represented as a graph where nodes represents the users, and edges represents a relationship between them. Two types of relationships are represented in the dataset as different sets of edges. The first is the non-symmetrical *social network*, where connections are directed, based on the "follows" relationship between two users. This means a user can follow another without the follow being reciprocated. The second is the symmetrical *spatial network*, where connections are undirected. Here, the links represent geographical distances between nodes, implying that if node A is connected to node B with a weight of 0.8, then node B is also connected to node A with the same weight. To represent this symmetry, the *spatial network* includes both connections as two distinct edges, making the symmetry explicit.

Every node is represented as a concatenation of words derived from all the textual contents of the user, preprocessed using the usual NLP pipeline (*tokenization*, *stopword removal*, and *lemmatization*), resulting in a collection of representative words for that user. It is important to note that this representation is built while preserving the temporal order of each content, potentially allowing the model to consider this aspect as well. The preprocessed and labeled dataset used in this work has been inherited from the work of (9), which can be consulted as a

reference for a deeper understanding of its construction[3].

The dataset includes a total of 4701 users, with 2654 labeled as *safe* and 2047 as *risky*. In terms of edges, there are 13,078 connections in the *social network* and more than 15 million connections in the *spatial network*. It is evident that the two classes are not balanced, with a predominance of *safe* users over *risky* users. This imbalance would be even more pronounced in a real-world scenario, as (hopefully) the number of risky users will always be significantly lower than the number of safe users. This element must be taken into account when training and the evaluating a model, especially if based on artificial neural network as in our case, since they are sensitive to class imbalance. The dataset includes only users with a substantial amount of text produced, so the analysis on the previously described silent users cannot be conducted using this data.

2.1 Textual content analisys

In order to understand the main textual features that characterized each class of our dataset, an analysis on the prominent used word on each class was conducted. This analysis leverage the TF-IDF score computed on each word for each class, removing the intersection between the two categories.

TF-IDF stands for Term Frequency-Inverse Document Frequency. It is a statistical measure used to evaluate the importance of a word t in a document d (in this case, the entire concatenation of words for each class) relative to a collection of documents (corpus) D. It is computed by the following formula:

$$TF\text{-}IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

More information about TF-IDF can be found at (10).

In Table 1 and 2, there are the top 30 words for each category i.e. *safe* or *risk* ranked by their TF-IDF. Furthermore, in bold are reported such words that has a clear sentiment related to them.

[3]The dataset used for this work is actually an expansion of the one used in (9), the expansion was conducted by the same authors.

Top 30 TF-IDF Safe Scores		
Rank	Term	TF-IDF
1	competencies	0.00095331
2	adopters	0.00086030
3	peony	0.00080217
4	fluency	0.00072079
5	workflows	0.00061616
6	expeditions	0.00059291
7	explorations	0.00058128
8	programmatic	0.00056966
9	segmentation	0.00056966
10	visualizations	0.00055803
11	**gorgeously**	0.00054641
12	**gladness**	0.00053478
13	technologists	0.00053478
14	specialization	0.00051153
15	caramels	0.00048828
16	cowl	0.00048828
17	multilingual	0.00047665
18	undervalue	0.00047665
19	jeweled	0.00045340
20	puree	0.00045340
21	tourmaline	0.00045340
22	annotate	0.00044178
23	chesterton	0.00044178
24	filigree	0.00044178
25	iterate	0.00044178
26	pimento	0.00044178
27	**gallant**	0.00043015
28	romaine	0.00043015
29	**devotions**	0.00041852
30	standouts	0.00041852

Table 1: Top 30 safe terms and their TF-IDF scores

Top 30 TF-IDF Risky Scores		
Rank	Term	TF-IDF
1	**drunker**	0.00028062
2	**gayness**	0.00026585
3	ayatollahs	0.00022154
4	jugged	0.00022154
5	chitterlings	0.00020677
6	inflaming	0.00020677
7	**phoniness**	0.00020677
8	sidesteps	0.00020677
9	autocrats	0.00019200
10	dapping	0.00019200
11	federalize	0.00019200
12	**hoaxed**	0.00019200
13	impalas	0.00019200
14	statists	0.00019200
15	uncritically	0.00019200
16	cesspools	0.00017723
17	**obsequious**	0.00017723
18	wetbacks	0.00017723
19	acta	0.00016246
20	antitank	0.00016246
21	**blabbering**	0.00016246
22	dermal	0.00016246
23	disavowal	0.00016246
24	fornicate	0.00016246
25	**hoodlum**	0.00016246
26	imperialists	0.00016246
27	lionized	0.00016246
28	**mafioso**	0.00016246
29	munched	0.00016246
30	**ungovernable**	0.00016246

Table 2: Top 30 risky terms and their TF-IDF scores

We can observe that the list of especially risky words includes many negative terms such as "drunker," "gayness," "mafioso," and others. This provides insight into the types of words used by *risky*, and potentially borderline, users. These lists can also help us understand the kinds of words that will influence the classifier in its process.

2.2 Networks Analysis

In order to explore and understand the insight in the networks provided by the dataset, we used a data visualization tool called *Gephi* [4] written in Java to show how each network is structured. In particular, we study the networks taking into account two mainly configurations:

1. The full *social network*.
2. A portion of the *spatial network* imposing a threshold on the edge value equal or greater than 0.90. This because the spatial network is very large, obtaining a sub-network reducing the number of actual connection to 433024.

In this section we first analyze the *social network* and then we analyze the reduced *spatial network*. In Figure 1:

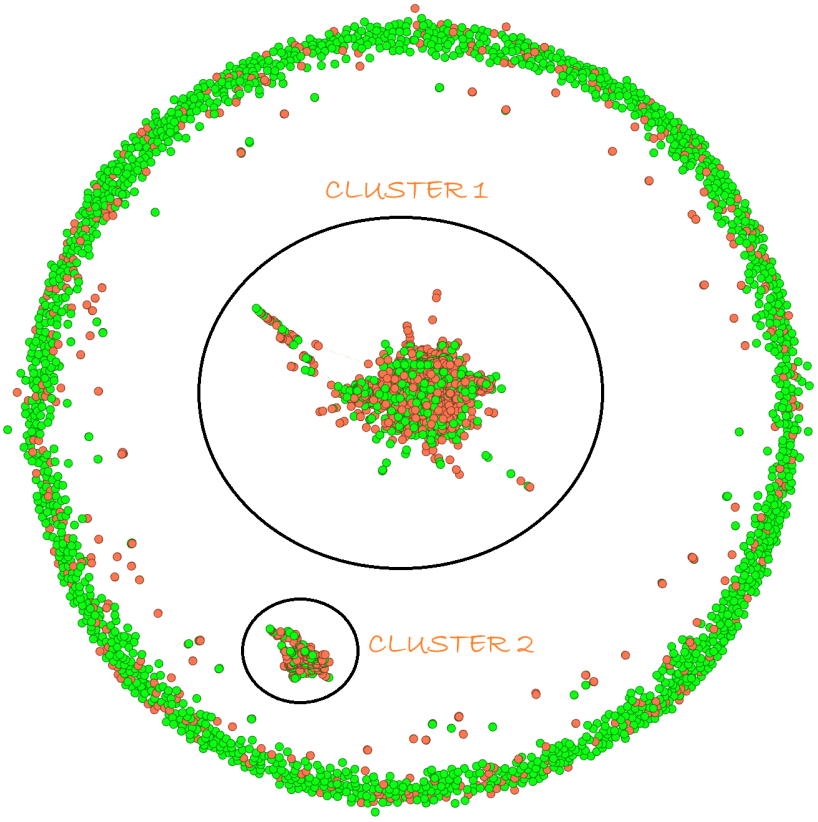

Figure 1: Social network

[4] https://gephi.org/

we can notice the full *social network* composed by two main clusters and a ring of mainly *safe* users that aren't connected with the rest of the network. Let us analyze in further detail the two clusters.

In Figure 2 and Figure 3,

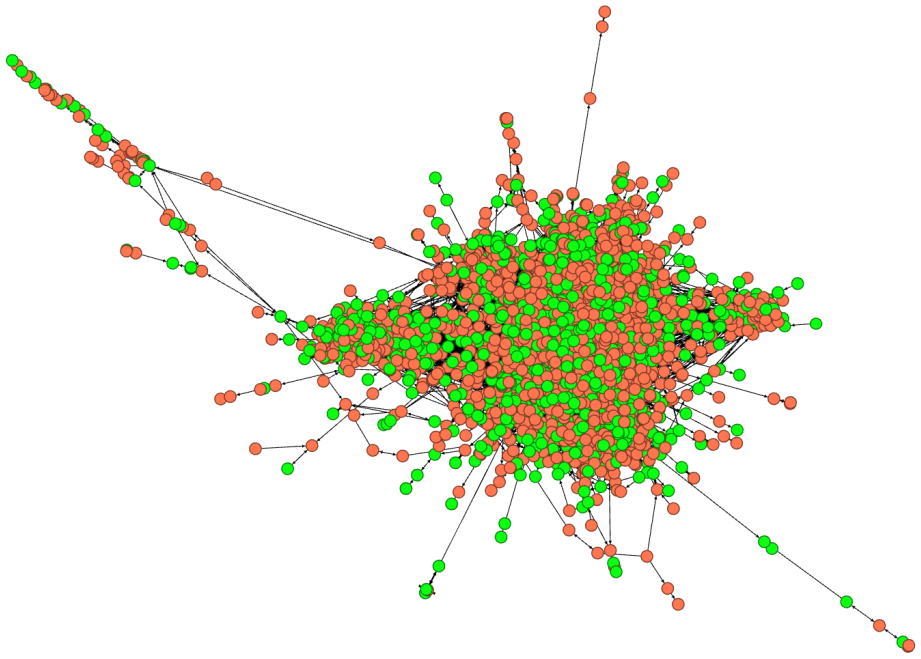

Figure 2: Social network cluster 1

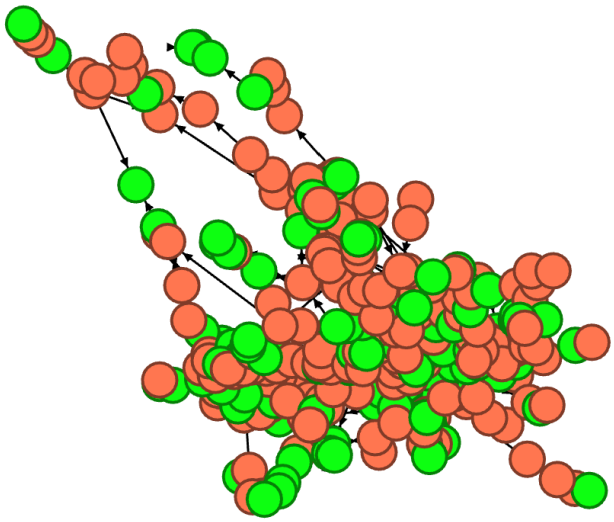

Figure 3: Social network cluster 2

we can see the two central clusters of the network composed by *risky* and *safe* users. We can analyze also an important aspect in social network: *risky* users are often connected each other

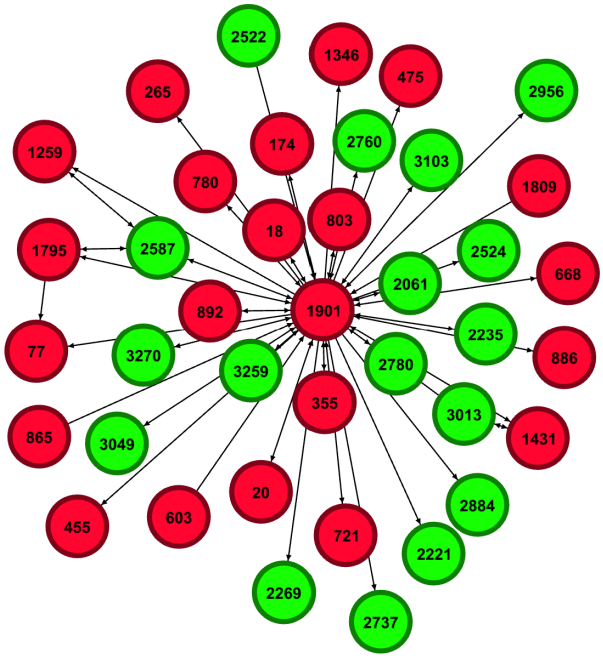

Figure 4: Social network risky users 1

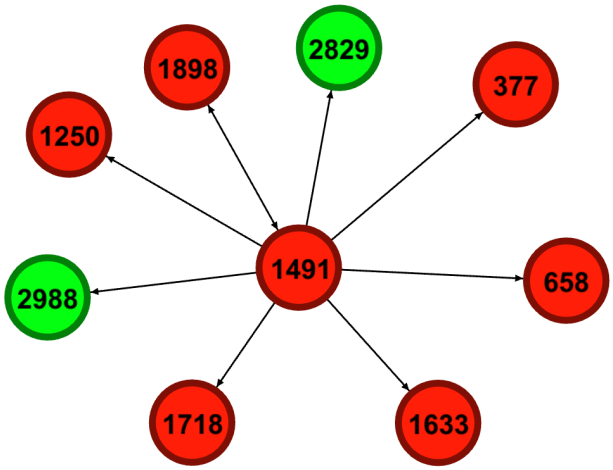

Figure 5: Social network risky users 2

as shown in Figure 4 and 5.

Finally, an important aspect is composed by borderline users. A typical example of such users is represented by journalists who share negative textual contents for informative purposes. Despite their contents, they are *safe* users and mainly connected with *safe* users as well. Indeed, they may usually publish posts containing unsafe words, increasing the chance of miss-classifications for content-based approaches.

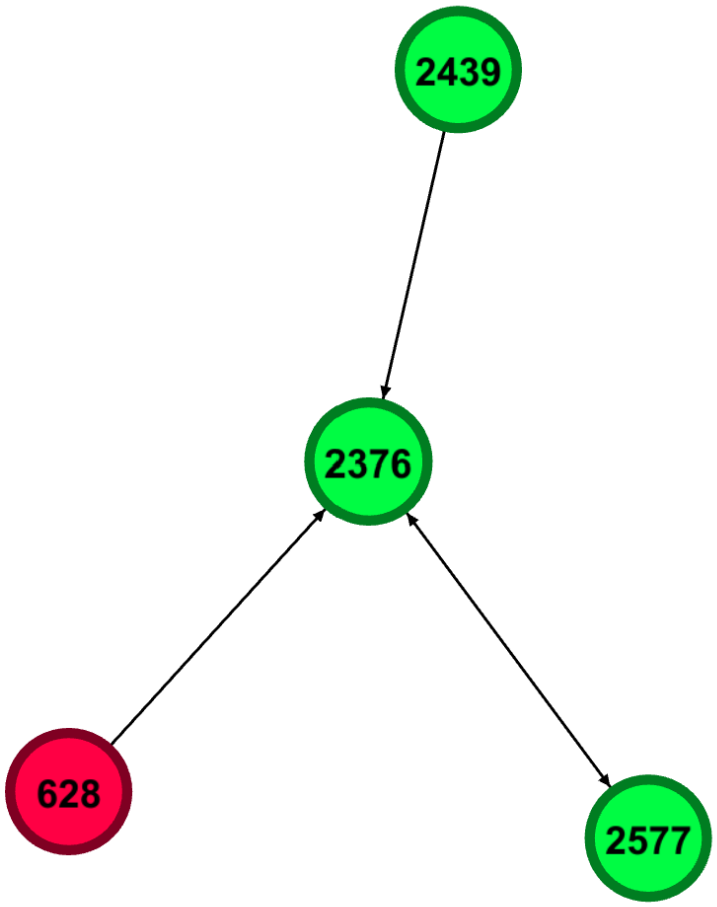

Figure 6: Social network borderline user 1

For example, in Figure 6 the user with ID=2376 is a *safe* borderline user mainly

connected with *safe* users (green labels). We can consider this as a borderline user because his posts contain words like "Trump"[5], strongly used by *risky* users, but he is labeled as *safe* in the dataset.

We can conduct a similar analysis on the reduced *spatial network* but in this case it doesn't make sense to analyze borderline users because the connections represent the physical distance rather than the user-following relationship. The algorithm used for the visualization of the reduced spatial network emphasizes the closeness of the nodes by visually aggregating them when they are more tightly connected. Furthermore, the existence of heterogeneous spatial clusters where *safe* and *risky* users are close, tends to contradicts the rule that *risky* and *safe* users form separate clusters. This because spatial clusters, except for specific situations, are inherently more heterogeneous. Consequently, our hypothesis is that spatial networks, when considered individually, are not as informative as social networks but can be used to support the latter. They can indicate the presence of clusters of potentially dangerous users who follow each other and they are also geographically close, thereby highlighting a potential danger situation.

In Figure 7 we can see the entire reduced *spatial network*.

[5]The word "Trump" appears 811 times related to *safe* users and 1287 related to *risky* users, so it it predominantly related to the *risky* class.

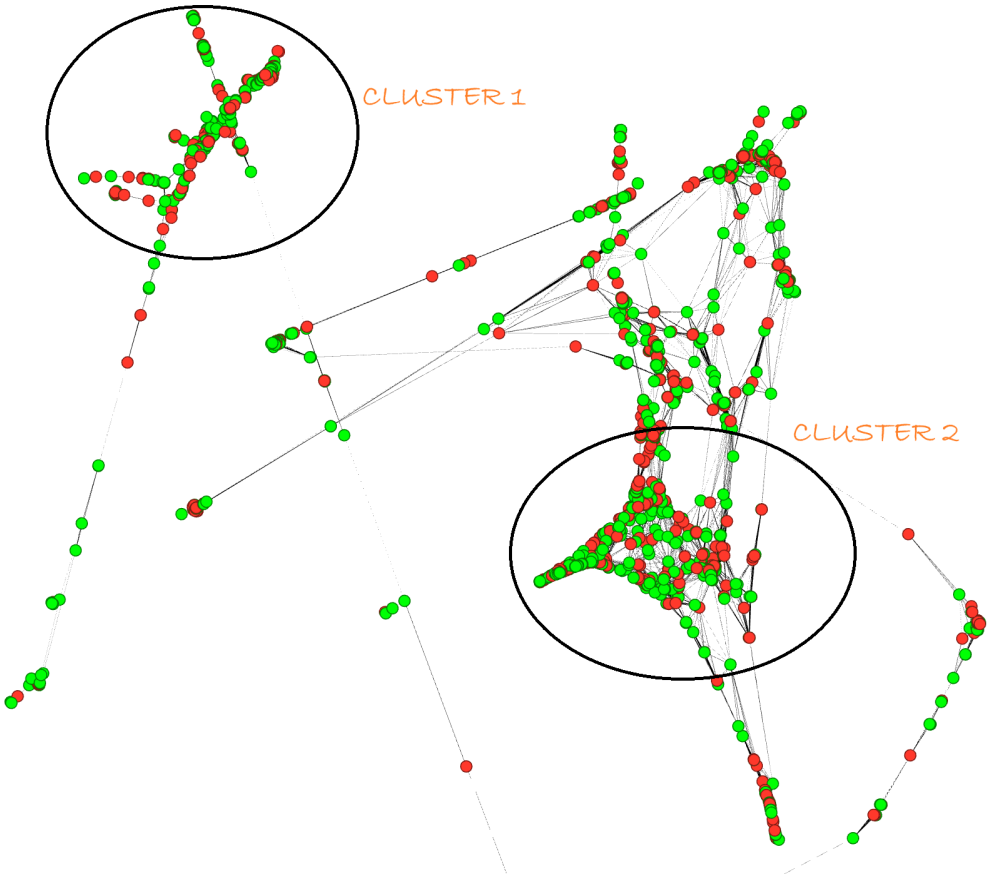

Figure 7: Reduced spatial network with threshold equal to 0.90

In this case as well the network is composed by two main clusters (details about the first cluster in Figure 8 and the second cluster in Figure 9).

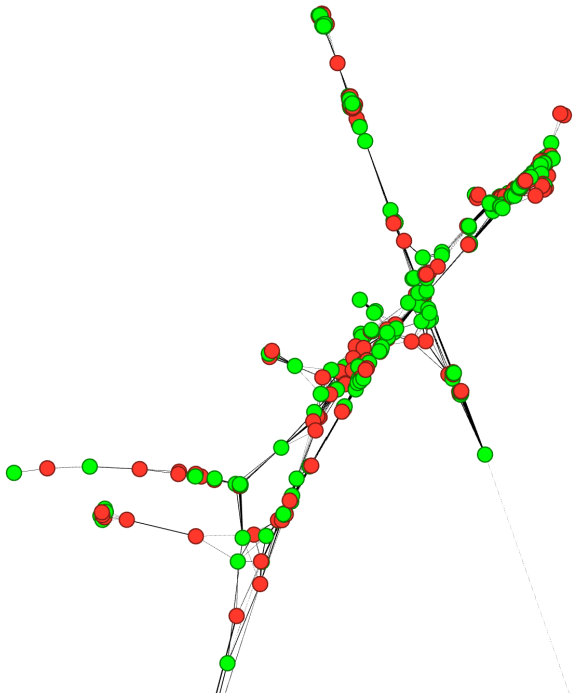

Figure 8: Reduced spatial, cluster 1

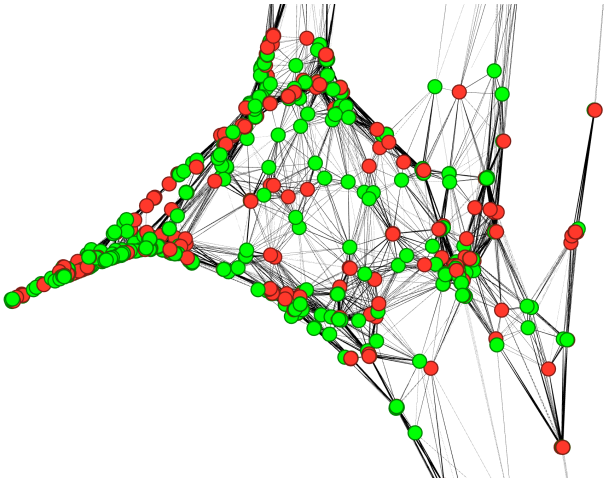

Figure 9: Reduced spatial network, cluster 2

In both the analyzed clusters, we can observe that the types of users are heterogeneous, as different geographical areas typically include both types of users.

However, there could also be small homogeneous clusters due to cultural influences in specific geographical areas. As we can see from Figure 10 where every

user deemed *risky* has established connections with other *risky* users, indicating a cluster of risk within the network.

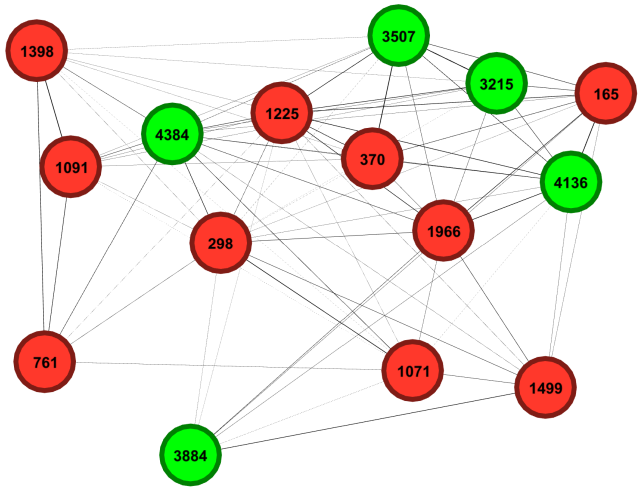

Figure 10: Reduced spatial network, cluster of risky users

The analysis of social and spatial networks reveals several key insights about the interconnections and classifications of users, particularly focusing on *risky* and *safe* borderline users. The networks are primarily composed of two central clusters, containing both *risky* and *safe* users. *Risky* users, in the case of *social networks* are interconnected, forming a sub-network within the larger network. This pattern highlights the tendency of *risky* users to associate closely with another. Borderline users, such as journalists who share negative content for informational purposes, are generally connected with *safe* users. The analysis of *spatial network* reveals that *risky* users could form small interconnected clusters within more heterogeneous ones. Taking into account both social and spatial aspects can reveal the presence of dangerous clusters within a specific geographical area. These findings underscore the importance of understanding the nuanced relationships and interactions within networks to effectively identify and manage different user groups, particularly in applications such as social media monitoring and social security.

3 Data Preparation

Data preparation is a crucial phase in the CRISP-DM methodology. It involves collecting, cleaning, transforming and organizing data to ensure that it is suitable for analysis and model training. This phase can be time-consuming, often taking

up to 80% of the total project time but it is essential for the success of any machine learning project (11).

Given the textual nature of the feature within the nodes, it's fundamental address a way to transform these data into numerical feature suitable for artificial neural network architectures. To address this result, we train three *Word2vec* models with embedding dimensions of 128, 256 and 512. *Word2vec* is a well established model that converts words into continuous vector space representation, allowing similar words to have similar vector representation. The algorithm, provided by the Gensim[6] library, accepts several parameters that affect both training speed and quality. The settings used in this work are:

- *vector_size*: the value is set to 128, 256 and 512. It is the dimension of the output embedding vector.
- *seed*: equal to 123, the seed is used for the random number generator and to ensure the experiment is randomized but reproducible.
- *window*: the value is configured to 10. The window is the maximum distance between the current and predicted word within a sentence.
- *min_count*: the value is 0 and it ignores all words with total frequency lower than 0.
- *sg*: is equal to 1 and indicates that the training algorithm is using the Skip-Gram methodology rather than Continuous Bag Of Word (CBOW).
- *workers*: the value is set to 5 and indicates how many worker threads are used to train the model.
- *epochs*: the number of epochs is 10 and indicates the number of iterations over the corpus (formerly: iter).

After the word2vec training phase, the embedding of each node are computed by using the following algorithm. The algorithm is used both for train nodes and for test nodes using all the trained word2vec models computing then three different embedded nodes datasets:

[6]https://radimrehurek.com/gensim/

Algorithm 1 computeW2Vembeddings, it takes as input the dimension of the embedder and an array of nodes. It return an array of embedding corresponding to the original list of nodes.

```
function computeW2Vembeddings(dim, nodes)
    node_list ← [ ]
    node_vect ← nodes['text']
    for s in node_vect do
        node_list.append(s.split(" "))
    end for
    w2v_model ← load_model(dim)
    node_embeddings ← [ ]
    for sentence in node_list do
        node_embedding ← zeros(dim)
        for word in sentence do
            if word in w2v_model.wv.index_to_key then
                ▷ We compute the value for the word and sum if the word is in the dictionary
                vector ← w2v_model.wv[word]
                node_embedding ← node_embedding + vector
            end if
        end for
        node_embeddings.append(node_embedding)
    end for
    return node_embedding
end function
```

As we can notice from Algorithm 1, the embedding of a node is computed by summing the obtained embedding vectors for each word in its list of words. Additionally, we also compute the embedding of each node using another algorithm, called Twitter4SSE, based on (12), that produce a representation at 768 dimensions. This different approach embeds directly the whole list of word of each node by using a pre-trained model based on Twitter data. Given the coherence with our dataset can be useful understand if a pre-trained embedder can influence positively the model and its results.

All the four embedded datasets are stored as .pkl file separated by train and test, potentially useful for future experiments.

Once the embeddings are created, the next step involves creating a matrix representation of the networks called adjacency matrix, built from the files containing the full dataset. When the chosen network type is the *social network*, all edge values are set to 1 where a connection between two nodes exists. For the *spatial network*, we first filter the data based on a specified threshold, considering only the edges whose related values greater than the threshold. Then, we build

the adjacency matrix by taking into account only the filtered values. We save then two pickle files: one containing train and test ids that will allow us to understand on which nodes we can perform forward propagation, and the other containing the built adjacency matrix. In the final pre-processing step, we perform the utility file operations. First, we load the train and the test embedding files created earlier, in the previous phase. We than combine the train and the test embeddings along with their associated user IDs in one single list. Afterward, we load the adjacency matrices created during the previous steps. Finally, we save all the processed data, ensuring it is ready for use in ie-HGCN model in an single .pkl file. These steps are crucial for the functioning of the model, which we will explain in the next chapter.

The whole pipeline is manageable by using a configuration file that easily allows to select which step perform and on which configuration, this in order to make easier to configure and repeat experiments.

4 Modelling: The ie-HGCN Model

As previously mentioned, the model used to conduct the experiments described in this work is the ie-HGCN (interpretable and efficient Heterogeneous Graph Convolutional Network)(8). This architecture is designed to effectively and efficiently learn representations of nodes in Heterogeneous Information Networks (HINs), which consist of networks composed of multiple types of nodes.

A key component of ie-HGCN is the hierarchical aggregation architecture, which includes object-level aggregation. This step involves aggregating information from a node's neighbors of different types by performing a projection from the neighbor representation to the current node representation, ensuring compatible representations before performing the actual aggregation by using convolution. Following object-level aggregation, a type-level attention mechanism is applied to learn and assign importance to different types of neighbors based on the objective task. Another key property of this model is its efficiency in learning and its interpretability. The weights learned by the model can be used as references to understand the meta-paths in the network that are related to the accomplishment of a certain task.

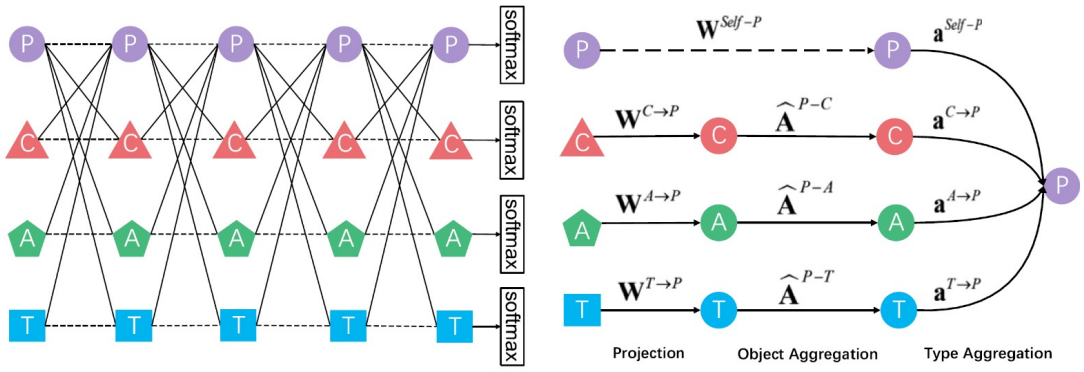

Figure 11: A schema of the aggregation mechanism performed bi ie-HGCN.

In figure 11 the authors explain the model using the DBLP dataset[7]. Based on the dataset, the authors construct an HIN with 4 object types: Paper (P), Author (A), Conference (C) and Term (T). Author (A) objects are labeled with the 4 research areas according to the conferences where they published papers. On the left-hand side, there is an instance of ie-HCGN with 5 layers. The solid lines stand for the relation-specific projection, and the dashed lines stand for the dummy self-relation projection. On the right hand side we can notice the P block in a layer computed performing the three phases aggregation. Additional detail on this model can be found at (8).

This model falls into the *transductive* learning category since it requires the entire network as input, training itself on the labeled nodes and performing predictions on the unlabeled ones. This model has been chosen for this study to explore its capabilities in the risk evaluation of users in social networks, given its flexibility and efficiency. It is also important to note that this model manages heterogeneity only at the node level, even though our dataset represents different kinds of links on the same node, however it can operate on homogeneous graph as well. This last aspect will be discussed in the final part of this work.

In our work we used this model adapted to the X dataset using the following configurations:

- *network_type*: can be spatial or social based to the type of network.
- *dimension*: a value equal to 128, 256, 512 or 768, it depends to the type of embedding we want to load.
- *threshold_dim*: it is -1 for social network or specified for spatial network (in this case can be set to 0, 0.49 or 0.69) allowing the model to load the proper preprocessed network.

[7]https://dblp.org/

- *cuda*: set to true, allows to use GPU for training if available.
- *lr*: learning rate set to 0.01.
- *weight_delay*: it is equal to 5e-4, a scaling factor used by the optimizer in order to apply the weight decay normalization.
- *type_att_size*: 64 it indicates the type attention dimension parameter.
- *type_fusion*: can be att (attention) or mean, in our case is valorized as att, represent the algorithm applied at last step of the node aggregation phase.
- *hid_layer_dim*: an array with values [64,32,16,8] representing the embedding dimensions of the nodes in the hidden layers of the model.
- *epochs*: equal to 250, number of training cycles.

The hyperparameters were selected by choosing the best set that maximizes the model's performance across a wide range of tasks based on the authors' research.

The model adapts the number of neurons of each layer to the number of node involved in the network, its input dimensions to the dimensionality of the node embeddings and its output dimensions to the number of classes we want to distinguish, which is two in our case. The outputs are computed using the log-softmax function on the final layer.

The source code is available on GitHub[8]. The experiments were run on Google Colab using the L4 GPU runtime, using the original proposed train-test split, which includes 3761 nodes in the training set and 940 nodes in the test set. We utilized all the different embedding dimensions and available network configurations to determine which configuration maximizes the model's capabilities in the risk node classification task.

5 Results and Evaluations

In this section we show the results obtained on the dataset where we emphasize in bold, the best results obtained for a given evaluation measure (column of a table). In addition to what we have described before, we decided to consider another network, called *no-network*, obtained by setting all the weights equal to 0 in order to not influence the classification. The results obtained through this network, as described in Table 3, are used as baseline for subsequent experiments. This serve just as basis for comparing the various model settings (social, spatial for each combination of dimensions and/or threshold) in order to understand how

[8]https://github.com/myDelevop/ie-HGCN

much the model improves its performances when the information of a network are involved in the classification task.

No-network baseline					
Dimension	**Macro F1**	**Micro F1**	**Precision**	**Recall**	**Accuracy**
128	0.7977	0.7979	**0.82**	**0.80**	**0.80**
256	**0.7998**	**0.800**	**0.82**	**0.80**	**0.80**
512	0.7946	0.7947	0.81	0.79	0.79
768	0.7919	0.7926	0.80	0.79	0.79

Table 3: Statistics for the no-network divided by the dimension of the embedding

Social network					
Dimension	**Macro F1**	**Micro F1**	**Precision**	**Recall**	**Accuracy**
128	0.7985	0.7989	0.81	0.80	0.80
256	0.8118	0.8128	**0.82**	0.81	0.81
512	**0.8188**	**0.8213**	**0.82**	**0.82**	**0.82**
768	0.8110	0.8138	0.81	0.81	0.81

Table 4: Statistics for the social network divided by the dimension of the embedding

Spatial network threshold = 0					
Dimension	**Macro F1**	**Micro F1**	**Precision**	**Recall**	**Accuracy**
128	0.7945	0.7947	0.81	0.79	0.79
256	**0.8019**	**0.8021**	**0.82**	**0.80**	**0.80**
512	0.3685	0.5670	0.65	0.57	0.57
768	0.7762	0.7766	0.79	0.78	0.78

Table 5: Statistics for the spatial network with threshold = 0 divided by the dimension of the embedding

As we can notice, there is a significant improvement in recall when the *social network* is involved in the classification task compared to the baseline. This behavior confirms our hypothesis about the informative nature of the *social network* in identifying *risky* users and potentially distinguish them from the borderline ones. We can notice this by analyzing in details the results obtained in the best *no-network* configuration with the best *social network* configuration.

Spatial network threshold = 0.49					
Dimension	**Macro F1**	**Micro F1**	**Precision**	**Recall**	**Accuracy**
128	0.8008	0.8011	**0.82**	**0.80**	**0.80**
256	**0.8041**	**0.8043**	**0.82**	**0.80**	**0.80**
512	0.3690	0.5681	0.76	0.57	0.57
768	0.7741	0.7745	0.79	0.77	0.77

Table 6: Statistics for the spatial network with threshold = 0.49 divided by the dimension of the embedding

Spatial network threshold = 0.69					
Dimension	**Macro F1**	**Micro F1**	**Precision**	**Recall**	**Accuracy**
128	0.7954	0.7957	0.81	**0.80**	**0.80**
256	0.7977	**0.7979**	**0.82**	**0.80**	**0.80**
512	0.3716	0.5691	0.76	0.57	0.57
768	**0.7868**	0.7872	0.80	0.79	0.79

Table 7: Statistics for the spatial network with threshold = 0.69 divided by the dimension of the embedding

No-network baseline detail on 256 dimension				
Class	**Precision**	**Recall**	**F1-Score**	**Support**
safe	0.89	0.73	0.81	531
risky	0.72	0.89	0.79	409

Table 8: Detailed results concerning the best configuration on the no-network

Social network detail on 512 dimension				
Class	**Precision**	**Recall**	**F1-Score**	**Support**
safe	0.85	0.83	0.84	531
risky	0.79	0.81	0.80	409

Table 9: Detailed results concerning the best configuration on the social network

It's evident that involving the *social network* helps the model balance precision and recall, improving performance for *risky* users. There is a substantial improvement in precision for the *risky* class, which also impacts the *safe* class recall. This implies that, without any information regarding social relationships, many *safe* users are classified as *risky*. This phenomenon is likely due to the presence of borderline users who are now more easily recognized.

On the other hand, using the *spatial network* does not result in significant improvements, supporting the previously described intuition about the lack of informativity given the heterogeneity of such networks when considered on their own.

We can also observe that using the *spatial network* with an embedding dimension of 512 results in a significant performance decline. The reason behind this behavior requires further analysis and verification.

Finally, we notice that the model achieves better results when using embeddings trained directly on the dataset (with dimensions of 128, 256, and 512). This behavior aligns with the idea that custom embeddings improve performance on a specific dataset. However, using a pretrained embedder could help the architecture be more flexible in different contexts and also reduce the time required for the embedding computation phase, bypassing the need for ad-hoc training. These factors should be considered when working in more complex and realistic contexts.

For the full set of detailed result, divided by class, please consult the results obtained at the GitHub repository available [9] in the folder *output_twitter*.

To conclude this section, we note that all the experiments were re-executed using the mean as the final aggregation mechanism. This was done to understand if the type-level attention could influence the model's accuracy, even though our dataset is homogeneous. The obtained results are similar to the ones already discussed, confirming that the type-level attention does not significantly affect the results in the context of homogeneous type datasets. For this kind of task, the model can be used with the simpler mean mechanism without any loss in accuracy. The actual results are not reported for brevity but can be found in the folder *output_twitter/mean* of the GitHub repository.

6 Conclusion and Future Works

The results obtained on the *social network* graph are promising, confirming our initial hypotheses. The model demonstrate its capability to manage this information effectively, enhancing performance compared to an approach based solely on textual analysis.

[9] https://github.com/myDelevop/ie-HGCN

By incorporating social network data, the model gains a more comprehensive understanding of context, sentiment, and user interactions, which enriches its predictive power. This integration allows for more nuanced and accurate interpretations, leading to improved outcomes and more informed decision-making processes. The success of this approach underscores the importance of leveraging diverse data sources in developing robust analytical models.

Experimenting starting from the new hypotheses is consistent with the iterative nature of the CRISP-DM methodology. Therefore, by revisiting and refining our hypotheses based on the latest findings, we align with the cyclical process of CRISP-DM, which emphasizes continuous improvement through iterative analysis. This approach ensures that each cycle of experimentation builds upon previous insights, leading to progressively more accurate and reliable models.

A possible new starting point for further development regards integrating the spatial network using a model capable of handling heterogeneity among links. This approach would allow for accurate management of cases where users are connected both socially and spatially, which are often more critical. By accounting for the complex relationships between social and spatial connections, the model should better identify and assess potential risks, leading to more effective monitoring and intervention strategies. This integration would enhance the model's ability to capture and analyze multifaceted interactions, ultimately improving its predictive accuracy and reliability in real-world scenarios. In order to achieve such result, the usage of an architecture capable of dealing with heterogeneous link like (13) must be investigated.

Furthermore, to effectively utilize the ie-HGCN model, it should be tested on a dataset where the objects are genuinely heterogeneous. Consider the potential relationships that exist between posts and comments; a simple list of words approach loses the dependencies between these contents. Therefore, this aspect should also be investigated. Testing ie-HGCN on such a dataset will reveal its ability to handle complex and varied relationships, offering a more accurate and nuanced understanding of the data. By exploring the connections between posts and comments, and potentially even multimedia contents, the model could capture the intricate dynamics and inter-dependencies that a simple list of words approach might overlook. This direction of investigation is crucial for enhancing the model's performance and ensuring that it can effectively manage and interpret heterogeneous data.

These new experiments are essential in order to refining the model and confirming its applicability to real-world data, where relationships and interactions can be complex and varied.

References

[1] D. Chaffey. (2024) Global social media statistics research summary may 2024. [Online]. Available: https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/ 1

[2] Statista. (2024) Number of x (formerly twitter) users worldwide from 2019 to 2024. [Online]. Available: https://www.statista.com/statistics/303681/twitter-users-worldwide/ 1

[3] B. Fung. (2021) Twitter bans president trump permanently. [Online]. Available: https://edition.cnn.com/2021/01/08/tech/trump-twitter-ban/index.html 1

[4] V. NidaUzel, E. Sarac, and S. Özel, "Using fuzzy sets for detecting cyber terrorism and extremism in the text," 10 2018, pp. 1–4. 1

[5] M. Bilgic and L. Getoor, "Effective label acquisition for collective classification," 08 2008, pp. 43–51. 1

[6] F. V. Alastair Reed, Joe Whittaker and S. Looney, "Radical filter bubbles: Social media personalisation algorithms and extremist content," 2019. 2

[7] A. P. A. B. W. Benjamin Sanchez-Lengeling, Emily Reif. (2021) A gentle introduction to graph neural networks. [Online]. Available: https://distill.pub/2021/gnn-intro/ 2

[8] J. L. W. Z. J. C. Q. W. Yaming Yang, Ziyu Guan, "Interpretable and efficient heterogeneous graph convolutional network," 2023. 3, 18, 19

[9] D. R. M. C. Antonio Pellicani, Gianvito Pio, "Sairus: Spatially-aware identification of risky users in social networks." 3, 4

[10] Wikipedia. Tf-idf. [Online]. Available: https://en.wikipedia.org/wiki/Tf-idf 4

[11] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," 2000. 16

[12] M. D. Giovanni and M. Brambilla, "Exploiting twitter as source of large corpora of weakly similar pairs for semantic sentence embeddings," 2021. 17

[13] Z. W. R. L. Y. C. D. Z. Shiping Wang, Sujia Huang, "Heterogeneous graph convolutional network for multi-view semi-supervised classification," 2024. [Online]. Available: https://doi.org/10.1016/j.neunet.2024.106438 24