A Multi-Perspective Approach for Risky User Identification in Social Networks

(Discussion Paper)

Antonio Pellicani[1,2,*], Gianvito Pio[1,2] and Michelangelo Ceci[1,2,3]

[1]Dept. of Computer Science, University of Bari "Aldo Moro", Via E. Orabona, 4, 70125 Bari, Italy
[2]Big Data Lab, National Interuniversity Consortium for Informatics (CINI), Via Volturno, 58, 00185 Roma, Italy
[3]Jožef Stefan Institute, Jamova Cesta 39, 1000 Ljubljana, Slovenia

Abstract

Social networks have become an integral part of modern communication, allowing people to connect and interact across the globe. However, they also bring along some negative phenomena, such as cyberbullying and social media addiction. As a result, monitoring user behavior and content has become essential to ensure a safe and responsible use of social networks. In this context, we recently proposed a novel system called SAIRUS, that we describe in this discussion paper. SAIRUS adopts three separate models to learn from multiple perspectives of social network data, namely the content posted by users, their relationships and their spatial closeness. We compare the system performance with 13 competitors on two real world datasets, demonstrating its superiority in identifying risky users and its usefulness as a tool for social network analysis.

Keywords

Social Network Analysis, User Risk Identification, Spatial Analysis

1. Introduction

Social networks enable people to connect and share news, opinions, and ideas through actions such as *posting*, *liking*, and *following* each other. This peculiarity fosters the creation of relationships and facilitates engagement in discussions on diverse topics and events. The widespread use of social networks has stimulated extensive research by the scientific community, mainly based on the use of Social Network Analysis (SNA) processes to explore the relationships and information exchange among users in the network [1]. In this context, our goal is to analyze social networks and identify *risky* users who engage in bad or illegal activities, such as drug selling or promotion, political or religious extremism, and discrimination against specific groups.

The identification of risky users is important for suspending suspicious accounts and preventing harmful behaviors in social network platforms. Many recent studies have focused on this area, including works on cyber-extremism and the identification of jihadist accounts [2, 3]. Methodologically, the identification of risky users can be approached as a node classification task

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy
*Corresponding author.
antonio.pellicani@uniba.it (A. Pellicani); gianvito.pio@uniba.it (G. Pio); michelangelo.ceci@uniba.it (M. Ceci)
0000-0002-4193-3486 (A. Pellicani); 0000-0003-2520-3616 (G. Pio); 0000-0002-6690-7583 (M. Ceci)

and thus can be generally categorized into three approaches: content-based, topology-based, and hybrid. Content-based approaches focus on analyzing user-generated content [4, 5], while topology-based approaches consider only user relationships (e.g., established through following, liking, or commenting actions) in the network [6, 7]. Finally, hybrid approaches combine the strengths of content-based and topology-based methods, making them particularly effective in classifying *borderline users* who may have a mix of both safe and unsafe content or relationships [8, 9]. A well-known example of users that may fall into this category are journalists.

It is noteworthy that social networks have become popular due to the possibility of interacting with them using mobile devices, which also integrate geolocation mechanisms. However, there have only been a few early attempts to use the spatial dimension in the analysis of (social) network data [10, 11], and many of the existing general approaches are unable to take into account the information conveyed by the geographic locations of the users, which can implicitly define new relationships among them. To fill this gap, we proposed SAIRUS [12], which takes into account the content generated by users, their relationships in the network, and their geographical positions to identify risky users. SAIRUS fuses three node classification models, each learned from a different perspective, using a stacked generalization approach to obtain a more robust final model, that also exploits the uncertainty of the predictions.

Unlike existing hybrid approaches that inject artificially-defined features related to a perspective into the other(s), SAIRUS allows a separate focus on each perspective and ultimately combines their contribution to learn a final classifier. Specifically, for the user-generated content, SAIRUS uses word embeddings to train two autoencoders specialized in identifying safe and risky users; for user relationships and spatial information, two separate embeddings based on the analysis of network data are extracted and two classifiers are trained on top. In the following section, we provide some details about such approaches adopted by SAIRUS.

2. The method SAIRUS

Before introducing SAIRUS, we provide a formal definition of a social network as a 4-tuple: $\langle N, C, E_C, E_T \rangle$, where:

- $N = N_L \cup N_U$ ($N_L \cap N_U = \emptyset$) is the set of users, either labeled (N_L) or unlabeled (N_U). Each labeled user is associated with the category *safe* or *risky*.
- C is the set of textual documents produced by users, that is, the posts. Each document $c \in C$ is associated with a *timestamp* and a *geographical location*.
- $E_C \subseteq N \times C$ refers to the relationship between users and the textual content they produce or share, specifically the action of creating or posting a particular textual content.
- $E_T \subseteq N \times N$ represents the topology of the social network, determined by the connections established between users through social relationships, e.g. *follows*.

In Figure 1, the four key stages performed by SAIRUS are depicted: *i)* the semantic content analysis of the textual documents generated by users, *ii)* the network topology analysis of user relationships, *iii)* the analysis of spatial closeness among users, and *iv)* the model fusion. In the following subsections, we briefly detail each of them.

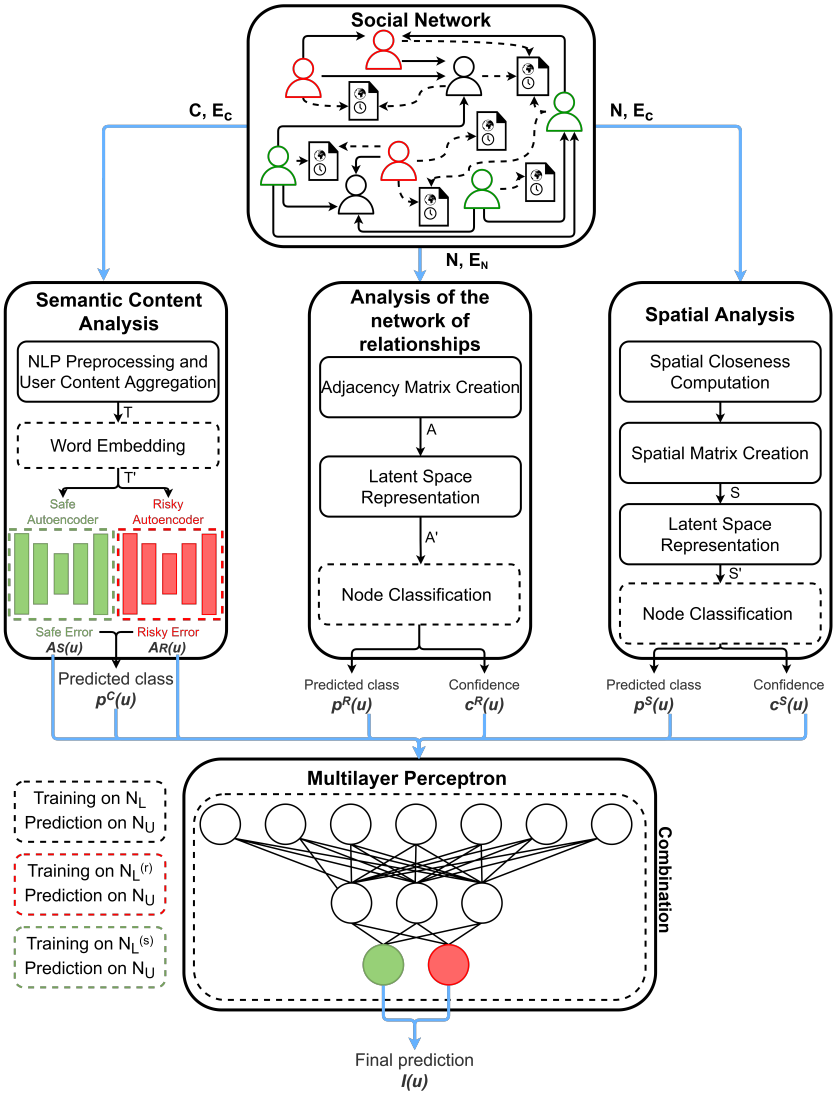

Figure 1: An overview of the SAIRUS architecture [12].

2.1. Semantic analysis of the user-generated content

The goal of this stage is to analyze the textual content produced by users and classify them as either safe or risky. It takes as input the set of textual documents C and the set of relationships E_C representing the link between users and the textual documents they posted. SAIRUS first applies standard Natural Language Processing (NLP) techniques such as tokenization, stopword removal, and stemming. Then, it concatenates all preprocessed documents posted by each user, taking into account the temporal order of the documents. This choice allows SAIRUS to implicitly capture the temporal evolution of the topics discussed by the user.

Subsequently, SAIRUS generates a k_c-dimensional feature vector for each user, by applying

the Word2Vec embedding method on each word of the concatenated documents. Specifically, an embedding for each user is obtained by summing up the embeddings of the words composing his/her concatenated document, according to the *additive compositionality* property [13].

In the final step, our attention is directed towards the labeled users N_L. We train two distinct one-class classifiers using stacked autoencoders: AR for the vector representation of labeled *risky* users and AS for the vector representation of labeled *safe* users. For the unlabeled users $u \in N_U$, we provide their corresponding vector representation to both autoencoders AS and AR and calculate their reconstruction errors $AS(u)$ and $AR(u)$. As a result, the semantic analysis of a user's textual content produces three outputs: *i)* the reconstruction error $AS(u)$ obtained by the autoencoder AS, *ii)* the reconstruction error $AR(u)$ obtained by the autoencoder AR, *iii)* the predicted label $p^c(u) \in \{S, R\}$ (safe or risky), computed according to the minimum error achieved by AS and AR. These outputs are used in the model fusion phase (see Figure 1).

2.2. Analysis of the network of relationships

SAIRUS considers the topology of the social network by directly analyzing the adjacency matrix $A \in \mathbb{R}^{|N|\times|N|}$, where $A_{ij} = 1$ if $(u_i, u_j) \in E_N$, $A_{ij} = 0$ otherwise, and u_i and u_j are the i-th and the j-th user of the network, respectively. However, the analysis of adjacency matrices may lead to issues due to high dimensionality and sparseness, since each user usually tends to establish relationships with a very small percentage of the whole set of users.

Many existent works rely on dimensionality reduction techniques to address the high dimensionality and sparseness problems. SAIRUS can work directly on the adjacency matrix $A \in \mathbb{R}^{|N|\times|N|}$, or on a transformed matrix $A' \in \mathbb{R}^{|N|\times k_r}$ resulting from the application of a dimensionality reduction technique to A, where k_r is a user-defined parameter. Specifically, SAIRUS can exploit PCA, autoencoders and Node2Vec, even if other techniques can be easily plugged in the workflow.

A node classification model is finally trained using the entire set of labeled users N_L. In this phase, SAIRUS exploits tree-based classifiers since they proved to provide optimal performances on classification problems in the semi-supervised scenario [14]. When provided with an unlabeled user $u \in N_U$, the learned decision tree returns the predicted label $p^R(u)$ and a confidence value $c^R(u)$, which is based on the purity of the training examples associated with the leaf node where u falls into. The predicted label and confidence value are then used in the model fusion phase, as illustrated in Figure 1.

2.3. Analysis of the spatial closeness among users

Similar to the analysis of the network of relationships, also the spatial analysis exploits an adjacency matrix built from the social network. In this case, SAIRUS uses a weighted matrix $S \in \mathbb{R}^{|N|\times|N|}$, where $S_{ij} = closeness(u_i, u_j)$ corresponds to the spatial closeness between the user u_i and the user u_j. Specifically, $closeness(u_i, u_j)$ is based on the geodetic distance $d(u_i, u_j)$ between the geographical locations of the users u_i and u_j that are estimated as the mode of the geographical locations associated to their posts on the social network. We standardize the distance $d(u_i, u_j)$ using the z-score normalization, obtaining $z(u_i, u_j)$, that allows us to distinguish two groups of user pairs: those who are spatially closer than the average (with

$z(u_i, u_j) < 0$) and those who are spatially more distant than the average (with $z(u_i, u_j) \geq 0$). Accordingly, we calculate $closeness(u_i, u_j)$ as follows:

$$closeness(u_i, u_j) = \begin{cases} \frac{z(u_i, u_j)}{min_z}, & \text{if } z(u_i, u_j) < 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where min_z is the minimum of the normalized distances between two users. Note that we further normalize $z(u_i, u_j)$ over min_z in order to obtain a value in the range $[0, 1]$, where 0 means that the users u_i and u_j are very far from each other (actually, more than the average) and 1 means that u_i and u_j are located precisely at the same location.

After computing the matrix S, we use a dimensionality reduction technique to obtain the reduced matrix $S' \in \mathbb{R}^{|N| \times k_s}$, where k_s is a user-defined parameter. Then, we train a node classification model on the labeled users N_L. Similar to the approach used for the network of relationships, we use a decision tree learner, which provides a predicted label $p^S(u)$ and a confidence value $c^S(u)$ for any unlabeled user $u \in N_U$. These outputs are then used in the model fusion phase (see Figure 1).

2.4. Model Fusion

The aim of the last step is to combine the results of the models based on the textual content, the network topology, and the spatial dimension to classify the unlabeled users in N_U. In SAIRUS, we use a Multi-Layer Perceptron (MLP) model to perform this task, following the Stacked Generalization approach [15].

The chosen MLP architecture is depicted in the bottom of Figure 1. It has an input layer comprising of 7 neurons, which considers the following inputs for a given user u: *i)* the reconstruction error values of the safe autoencoder $AS(u)$ and risky autoencoder $AR(u)$, along with the predicted label $p^c(u)$ derived from the semantic analysis component for the textual content; *ii*) the predicted label $p^R(u)$ and confidence value $c^R(u)$ obtained from the component responsible for analyzing the network of relationships; *iii*) the predicted label $p^S(u)$ and the confidence value $c^S(u)$ obtained from the component responsible for the spatial analysis. We use the sigmoid activation function in the hidden layer to capture any non-linear relationships between the input and output variables. In contrast, we use the softmax activation function in the output layer for the final classification.

It is noteworthy that our approach, which uses the stacked generalization framework, does not require any user-defined criteria/weight to merge the outputs of three distinct models. Moreover, in contrast to ensemble techniques that solely rely on combining predictions ($p^C(u)$, $p^R(u)$, and $p^S(u)$, in our case), SAIRUS can incorporate other features such as reconstruction errors $AS(u)$ and $AR(u)$, and prediction confidences $c^R(u)$ and $c^S(u)$, that make it more robust to the uncertainty of the predictions and to the possible presence of noise in the data.

3. Experiments

We collected a real-world dataset from Twitter to evaluate the performance of SAIRUS. The dataset was associated with sentiment scores for each tweet, which were computed using the

Stanford CoreNLP Toolkit and manually revised by three domain experts.

To label users as either *risky* or *safe*, two strategies were employed. The first strategy relied on identifying tweets containing specific **keywords** related to threats, terrorism, hate against immigrants, and women. The second strategy assigned a score to each user by summing the **sentiment** scores of their tweets. The assumption was that users with a higher number of negative sentiment tweets are more likely to be risky.

To ensure the accuracy of the labeling process, we initially labelled the top-ranked users as *safe* and the bottom-ranked users as *risky*, whose posts were also manually inspected by three expert reviewers. We also introduced a set of borderline users, who were initially classified as *risky* but had mostly safe connections, to introduce noisy data under controlled conditions. These users may correspond to journalists who share *negative* content for informational purposes, but have mostly connections with *safe* users. The resulting datasets consisted of 2241 safe users (including 263 borderline users) and 1467 risky users for the keyword strategy, and 2047 safe users (including 304 borderline users) and 1033 risky users for the sentiment strategy, with 11,659,043 and 13,970,379 tweets, respectively.

We assessed the performance of SAIRUS using *PCA*, *Node2Vec*, and *Autoencoders* for the reduction of the dimensionality. We also evaluated the results with different values of the embedding dimensionality, namely k_c for the semantic analysis of the textual content, k_r for the analysis of the network of relationships, and k_s for the spatial analysis. After conducting some preliminary evaluations, we chose the following parameter combinations for the experiments: $\langle k_c{=}128, k_r{=}256, k_s{=}256\rangle$, $\langle k_c{=}256, k_r{=}128, k_s{=}128\rangle$, and $\langle k_c{=}512, k_r{=}128, k_s{=}128\rangle$. For space constraints, here we report the best results (all the results can be found in [12]).

We compared the performance of SAIRUS with several other methods, including a Random Forest model (**RF**) with 100 trees, and two one-class classifiers based on autoencoders (**1C-AEs**) designed for content-based analysis, which is consistent with the methodology used in SAIRUS. We used different feature sets, each focusing on one or more perspectives, such as content (C), relationships (R), or spatial (S). For multiple perspectives, we concatenated the feature sets of each single perspective (C+R, C+S, R+S, and C+R+S). To embed the textual content, we used state-of-the-art systems such as Word2Vec (**w2v**) and Doc2Vec (**d2v**), with embedding dimensionality set to k_c, which is the same as that used by SAIRUS. The embedding of the network of relationships and of the spatial closeness, we used Node2Vec (**n2v**) with embedding dimensionality set to k_r and k_s, respectively, following the setting adopted for SAIRUS.

We adopted a stratified 5-fold cross-validation technique, which preserved the proportion of safe and risky users, as well as the ratio of borderline users within safe users. Our evaluation metrics included precision, recall, F1-Score, and accuracy, with the positive class being the *risky* label. In addition, we computed these measures specifically on the borderline users to determine the performance of the methods in handling noisy data.

3.1. Results

In Tables 1 and 2, we show the results obtained on the *sentiment* dataset and on the *keywords* dataset, respectively, where we emphasize the best result obtained for a given evaluation measure. By looking at the competitor solutions solely based on textual content, we notice that the use of w2v generally leads to better results than d2v (as also observed in [16]). On the other

Table 1
Results on the *sentiment* dataset, with $k_c = 512, k_r = 128, k_s = 128$.

	Configuration					All users				Bordeline			
	Classifier	C	R	S	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc
COMPETITORS	1C-AEs	✓(d2v)			0.567	0.546	0.540	0.654	0.500	0.363	0.419	0.727
	1C-AEs	✓(w2v)			0.612	0.604	0.605	0.635	0.500	0.155	0.233	0.310
	RF	✓(d2v)			0.562	0.524	0.490	0.672	0.500	0.443	0.469	0.887
	RF	✓(w2v)			0.687	0.686	0.686	0.686	0.500	0.165	0.248	0.331
	RF		✓		0.508	0.504	0.474	0.646	0.500	0.445	0.471	0.890
	RF			✓	0.498	0.499	0.464	0.645	0.500	0.458	0.478	0.917
	RF	✓(d2v)	✓		0.559	0.517	0.471	0.676	0.500	0.455	0.476	0.910
	RF	✓(w2v)	✓		0.681	0.680	0.680	0.680	0.500	0.146	0.225	0.292
	RF	✓(d2v)		✓	0.535	0.521	0.505	0.648	0.500	0.410	0.450	0.820
	RF	✓(w2v)		✓	0.602	0.602	0.602	0.602	0.500	0.198	0.283	0.396
	RF		✓	✓	0.503	0.502	0.479	0.636	0.500	0.430	0.462	0.860
	RF	✓(d2v)	✓	✓	0.539	0.519	0.498	0.653	0.500	0.428	0.461	0.857
	RF	✓(w2v)	✓	✓	0.607	0.607	0.607	0.607	0.500	0.179	0.263	0.358
SAIRUS	Dimensionality Reduction: AE	✓	✓		0.777	0.784	0.776	0.784	1.000	0.853	0.920	0.853
		✓		✓	**0.814**	**0.816**	**0.810**	**0.816**	1.000	0.847	0.890	0.847
			✓	✓	0.776	0.783	0.776	0.783	1.000	0.853	0.920	0.853
		✓	✓	✓	0.806	0.807	0.795	0.807	1.000	0.940	0.968	0.940
	Node2vec	✓	✓		0.757	0.758	0.757	0.758	1.000	0.770	0.868	0.770
		✓		✓	0.710	0.754	0.728	0.754	**1.000**	**0.970**	**0.985**	**0.970**
			✓	✓	0.776	0.772	0.774	0.772	1.000	0.787	0.879	0.787
		✓	✓	✓	0.788	0.786	0.773	0.786	1.000	0.953	0.976	0.953
	PCA	✓	✓		0.790	0.797	0.789	0.797	1.000	0.860	0.924	0.860
		✓		✓	0.566	0.612	0.585	0.612	1.000	0.943	0.970	0.943
			✓	✓	0.786	0.793	0.785	0.793	1.000	0.860	0.924	0.860
		✓	✓	✓	0.751	0.741	0.691	0.741	1.000	0.967	0.983	0.967
	None	✓	✓		0.800	0.779	0.744	0.779	1.000	0.877	0.925	0.877
		✓		✓	0.636	0.639	0.637	0.639	1.000	0.850	0.911	0.850
			✓	✓	0.793	0.768	0.727	0.768	1.000	0.950	0.974	0.950
		✓	✓	✓	0.755	0.719	0.655	0.719	1.000	0.967	0.983	0.967

hand, Random Forest (RF) and 1C-AEs showed comparable results, with no clear domination of one solution over the other. However, the adoption of features related to user relationships (R), to the spatial closeness (S), or a combination of these perspectives did not seem to provide a clear contribution to the competitors. This result confirms that simply injecting features coming from one perspective into the other could also compromise the classifier performances due to the possible introduction of issues related to the course of dimensionality.

In contrast, SAIRUS achieved the best results when leveraging the network of user relationships or the spatial dimension (or both). This was particularly evident in the sentiment dataset, where the F1-score reached ~ 0.8 when both user relationships and the spatial analysis were considered. These results demonstrate that the fusion strategy adopted by SAIRUS is more effective than the concatenation of features. In the keywords dataset, the configuration that leveraged both textual content and spatial analysis slightly emerged as the best. These results confirmed the relevance of the spatial perspective and the importance of properly modeling and exploiting it through a smart fusion strategy. Moreover, the obtained results prove that the spatial dimension is an important factor for predicting borderline users, regardless of network representation used. In other words, incorporating spatial information improves the accuracy of predictions for borderline users.

SAIRUS outperformed competitors in both datasets, demonstrating its ability to effectively

Table 2
Results on the *keywords* dataset, with $k_c = 128, k_r = 256, k_s = 256$

	Configuration					All users				Bordeline			
	Classifier	C	R	S	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc
COMPETITORS	1C-AEs	✓(d2v)			0.547	0.546	0.544	0.546	0.500	0.219	0.298	0.438
	1C-AEs	✓(w2v)			0.637	0.631	0.630	0.631	0.500	0.238	0.318	0.477
	RF	✓(d2v)			0.559	0.559	0.559	0.559	0.500	0.215	0.297	0.431
	RF	✓(w2v)			0.687	0.686	0.686	0.686	0.500	0.165	0.248	0.331
	RF		✓		0.496	0.496	0.494	0.496	0.500	0.254	0.337	0.508
	RF			✓	0.511	0.511	0.509	0.511	0.500	0.225	0.309	0.450
	RF	✓(d2v)	✓		0.567	0.567	0.566	0.567	0.500	0.221	0.303	0.442
	RF	✓(w2v)	✓		0.681	0.680	0.680	0.680	0.500	0.146	0.225	0.292
	RF	✓(d2v)		✓	0.544	0.544	0.543	0.544	0.500	0.231	0.313	0.462
	RF	✓(w2v)		✓	0.602	0.602	0.602	0.602	0.500	0.198	0.283	0.396
	RF		✓	✓	0.489	0.490	0.488	0.489	0.500	0.256	0.338	0.512
	RF	✓(d2v)	✓	✓	0.543	0.543	0.541	0.543	0.500	0.235	0.315	0.469
	RF	✓(w2v)	✓	✓	0.623	0.623	0.623	0.623	0.500	0.173	0.233	0.347
SAIRUS	Dimensionality Reduction: AE	✓	✓		0.599	0.862	0.696	0.616	0.600	0.419	0.493	0.419
		✓		✓	0.620	**0.870**	0.711	0.636	0.600	0.569	0.584	0.569
			✓	✓	0.667	0.766	0.713	0.691	1.000	0.700	0.822	0.700
		✓	✓	✓	0.632	0.858	0.711	0.641	0.600	0.538	0.565	0.538
	Node2vec	✓	✓		0.608	0.794	0.668	0.605	0.600	0.404	0.482	0.404
		✓		✓	0.657	0.824	0.708	0.648	0.600	0.492	0.538	0.492
			✓	✓	0.689	0.676	0.682	0.685	1.000	0.596	0.746	0.596
		✓	✓	✓	0.717	0.684	0.677	0.672	0.800	0.669	0.721	0.669
	PCA	✓	✓		0.618	0.867	0.709	0.633	0.600	0.462	0.522	0.462
		✓		✓	0.528	0.693	0.577	0.525	0.600	0.546	0.572	0.546
			✓	✓	0.687	0.776	**0.729**	**0.711**	1.000	0.746	0.854	0.746
		✓	✓	✓	0.682	0.661	0.645	0.646	0.800	0.727	0.760	0.727
	None	✓	✓		0.749	0.568	0.526	0.578	0.600	0.565	0.582	0.565
		✓		✓	0.543	0.696	0.585	0.537	0.600	0.527	0.561	0.527
			✓	✓	**0.892**	0.317	0.468	0.665	**1.000**	**0.938**	**0.968**	**0.953**
		✓	✓	✓	0.666	0.424	0.443	0.562	0.800	0.781	0.790	0.781

handle noisy users while maintaining high predictive accuracy. This is mainly due to the hybrid approach based on multiple perspectives to make final predictions. Moreover, its ability to capture information from user relationships and spatial closeness makes it a cutting-edge tool for distinguishing between safe and risky users in social networks, paving the way towards its adoption for the analysis of large amounts of data from geo-located mobile devices.

4. Conclusion

This paper discussed SAIRUS, a novel approach for identifying *risky* users in social networks. By combining multiple perspectives of social network data, including textual content, user relationships, and spatial closeness, SAIRUS can accurately classify users, outperforming 13 competitor systems that exploit either one perspective at a time or a combination thereof. In our experiments, SAIRUS also proved to be robust to the presence of noisy users.

In addition to its current capabilities, SAIRUS has the potential to incorporate the temporal dimension related to textual content and detect sudden changes in user behavior. Therefore, future work will focus on extending SAIRUS to make it able to capture the dynamism of the network of relationships and spatial closeness among users, providing a more comprehensive risk assessment of social network users.

Acknowledgments

The authors acknowledge the support of the European Commission through the H2020 Project "CounteR - Privacy-First Situational Awareness Platform for Violent Terrorism and Crime Prediction, Counter Radicalisation and Citizen Protection" (Grant N. 101021607). This work was also partially supported by the project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI, under the NRRP MUR program funded by the NextGenerationEU.

References

[1] S. Tabassum, F. S. Pereira, S. Fernandes, J. Gama, Social network analysis: An overview, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (2018).
[2] I. Awan, Cyber-Extremism: Isis and the Power of Social Media, Society 54 (2017) 138–149.
[3] A. Al-Rawi, J. Groshek, Jihadist Propaganda on Social Media: An Examination of ISIS Related Content on Twitter, Int. Journal of Cyber Warfare and Terrorism 8 (2018) 1–15.
[4] V. N. Uzel, E. Saraç Eşsiz, S. Ayşe Özel, Using fuzzy sets for detecting cyber terrorism and extremism in the text, in: ASYU 2018, 2018, pp. 1–4.
[5] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International conference on machine learning, 2014, pp. 1188–1196.
[6] S. A. Macskassy, F. Provost, Classification in networked data: A toolkit and a univariate case study, Journal of machine learning research 8 (2007) 935–983.
[7] M. Bilgic, L. Getoor, Effective label acquisition for collective classification, in: Proc. ACM SIGKDD 2008, KDD '08, ACM, 2008, pp. 43–51.
[8] M. Mateen, M. A. Iqbal, M. Aleem, M. A. Islam, A hybrid approach for spam detection for twitter, in: 2017 14th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2017, pp. 466–471.
[9] T. Hamdi, H. Slimi, I. Bounhas, Y. Slimani, A hybrid approach for fake news detection in twitter based on user features and graph embedding, in: Distributed Computing and Internet Technology, Springer International Publishing, Cham, 2020, pp. 266–280.
[10] R. Medina, G. Hepner, Geospatial analysis of dynamic terrorist networks, in: Values and violence, Springer, 2008, pp. 151–167.
[11] M. A. Masood, R. A. Abbasi, Using graph embedding and machine learning to identify rebels on twitter, Journal of Informetrics 15 (2021) 101121.
[12] A. Pellicani, G. Pio, D. Redavid, M. Ceci, Sairus: Spatially-aware identification of risky users in social networks, Information Fusion 92 (2023) 435–449.
[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, CoRR abs/1310.4546 (2013). arXiv:1310.4546.
[14] J. Levatic, D. Kocev, M. Ceci, S. Dzeroski, Semi-supervised trees for multi-target regression, Inf. Sci. 450 (2018) 109–127.
[15] D. H. Wolpert, Stacked generalization, Neural Networks 5 (1992) 241–259.
[16] G. De Martino, G. Pio, M. Ceci, PRILJ: an efficient two-step method based on embedding and clustering for the identification of regularities in legal case judgments, Artificial Intelligence and Law 30 (2022) 359–390.