Hindawi
Complexity
Volume 2021, Article ID 7955637, 15 pages
https://doi.org/10.1155/2021/7955637

Research Article

Uncovering Cybercrimes in Social Media through Natural Language Processing

Julián Ramírez Sánchez,[1] **Alejandra Campo-Archbold**,[1] **Andrés Zapata Rozo**,[1] **Daniel Díaz-López**,[1] **Javier Pastor-Galindo**,[2] **Félix Gómez Mármol**,[2] **and Julián Aponte Díaz**[3]

[1]*School of Engineering, Science and Technology, Universidad del Rosario, Carrera 6 # 1 2 C - 16, Bogotá 111711, Colombia*
[2]*Faculty of Computer Science, University of Murcia, Edificio 32, Campus de Espinardo, Murcia 30100, Spain*
[3]*Armada Nacional de Colombia, Carrera 54 # 26 - 25, CAN, Bogotá 111321, Colombia*

Correspondence should be addressed to Daniel Díaz-López; danielo.diaz@urosario.edu.co

Received 3 October 2021; Accepted 25 November 2021; Published 10 December 2021

Academic Editor: Kai Hu

Among the myriad of applications of natural language processing (NLP), assisting law enforcement agencies (LEA) in detecting and preventing cybercrimes is one of the most recent and promising ones. The promotion of violence or hate by digital means is considered a cybercrime as it leverages the cyberspace to support illegal activities in the real world. The paper at hand proposes a solution that uses neural network (NN) based NLP to monitor suspicious activities in social networks allowing us to identify and prevent related cybercrimes. An LEA can find similar posts grouped in clusters, then determine their level of polarity, and identify a subset of user accounts that promote violent activities to be reviewed extensively as part of an effort to prevent crimes and specifically hostile social manipulation (HSM). Different experiments were also conducted to prove the feasibility of the proposal.

1. Introduction

Information and communications technology (ICT) has revolutionized our society, and artificial intelligence in particular is currently leading such a revolution, taking a central role able to remarkably impact the near future of humankind [1]. Thus, researchers devoted to artificial intelligence raised the following question: Could a machine replace some people's functionalities and become a central axis for the next generations in certain aspects of their lives? Starting from such a question, different advances have been made in that regard, and in this paper, we specifically review the ability of artificial intelligence to understand human language.

Natural language processing (NLP) is the area of artificial intelligence focused on interpreting human communication through computational machine learning models [2]. Uncovering the essence of human words is one of the goals of NLP, which allows algorithms to get the meaning of full sentences expressed by people. In this way, an NLP model could understand the expressiveness of a phrase, interpret the desires or emotions of a person from the use of certain words, or even establish similarities of intentions between sentences [3, 4]. Thus, NLP brings a promising future for the understanding of human language, which may be useful in different scenarios such as customer service, advertising, voice translation, and profiling of suspects, among others [5].

In turn, NLP similarity models are used to find the closeness between two texts according to their meaning [6]. To process every text and perform any machine learning task on it, it must be first converted into a numerical format. The understanding of the semantics of a phrase and the consequent determination of similarity may be used in a variety of fields and for different purposes such as (i) finding similar user questions in online forums to assign them the same answer, (ii) spotting similar online documents to detect plagiarism, (iii) recommending similar news in online newspapers to improve journalistic research, or (vi)

identifying similarities between posts on social media and profile groups of users.

Even if the use of NLP models in cybersecurity is a recent research field, there have been some proposals that aim to build classifiers of radical and nonradical online users [7] and to develop annotation and word embedding methods [8]. Other proposals of the use of NLP models in cybersecurity aim to design models to detect hate speech in cyberspace [9] and to interact with suspects to profile their interest regarding online child sexual abuse [3].

In this context, the paper at hand proposes a solution to uncover cybercrimes in social media through NLP. It uses an NLP similarity model to identify groups of user accounts in social media that generated messages promoting violence and hate, thus impacting public safety. This last situation is an undesirable use of ICT that goes beyond the legitimate social protest and is considered a cybercrime, as it may be part of a set of coordinated activities aimed to provoke instability. Instability provoked by a threat agent is known as a campaign of HSM and is one of the most difficult cyber operations to unveil as it may face typical challenges of any worldwide cyber incident [10] (no sovereignty, anonymity, and lack of regulation, among others), making it difficult to identify the actual threat agent behind such campaign [11]. Hence, our solution aims to support labors of LEA in the prevention of cybercrimes, helping profile suspects through the generation of clusters of users and the understanding of their polarization.

The remainder of the paper is structured as follows. Section 2 describes some remarkable related works found in the literature. Section 3 proposes the solution applying NLP models, as part of a data science lifecycle, for the detection and prevention of cybercrimes. Then, the application of the previous proposal in a cybersecurity context is presented in Section 4, which contains the evaluation and analysis of the obtained results. An analysis of the application of NLP models as part of a national cyber defense strategy is included in Section 5. Last but not least, Section 6 contains some highlights derived from the work done and sheds light on some future research directions.

2. State of the Art

Several scientists have worked on NLP to support cybersecurity and cyber defense activities [12], such as protecting systems, detecting suspect movements and groups, monitoring risky scenarios, or finding criminal profiles, as can be seen in Table 1.

In cybersecurity, Tamura and Matsuura [13] proposed the combination of a Markov chain and packet flow similarity to improve anomaly detection during scanning attacks against industrial control systems (ICSs). A packet was designated as suspect if both the Markov chain model and the similarity model detected irregularities in terms of time and content, respectively. The detection of cyberattacks on network services can be complemented with social media feeds. On the one hand, Chambers et al. [14] implemented two NLP models, a continuous bag-of-words (CBOW) model and a topic-based generative model (partially labeled Dirichlet allocation, PLDA) that processed tweets for binary classification (attack or nonattack) and characterization of users behavior with topics. On the other hand, Khandpur et al. [15] used a similarity model (convolution tree kernel), domain generation algorithm, dynamic query expansion, and clustering to detect "account hijacking," "data breaches," and "DDoS attacks" on Twitter. These types of attacks were also extracted, in the same social network, by Ritter et al. [16] through the employment of named-entity recognition (NR) and semisupervised expectation regularization.

Another approach for reviewing security on Android applications was addressed by Kong et al. [17], who designed the system AUTOREB to categorize Google Play app reviews within four security categories ("spamming," "financial issue," "overprivileged permission," and "data leakage") and aggregate the overall app risk level. The latter was achieved through a bag-of-words (BOW) and sparse support vector machine (SVM) classifier and the former with crowdsourcing techniques. Moreover, to extract indicators of compromise (IOC) from public sources, Liao et al. [18] worked on its automation to improve cyber threat intelligence. They particularly proposed iACE, a model that uses NLP (dependency parsing and topic term extraction through part-of-speech tagging (POST)), classifiers, and graph mining to analyze technical and distinguishing IOCs and their context.

Moreover, people express their thoughts on social media, which in extreme cases may suppose hate crime. In this sense, some works in the literature have a focus on the detection of hate speech. Kohatsu et al. [9] proposed HaterNet, an intelligent system that employed a long short-term memory neural network with a multilayer perceptron neural network, in conjunction with a series of classifiers (e.g., linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), random forest (RF), ridge logistic regression (RLR), and support vector machine (SVM)) to infer a tweet that contains a hate message. A similar approach was proposed by Khan et al. [19], annotating tweets as "hate speech," "offensive," or "nonoffensive" using a sequential convolutional neural network (SCNN). Gambäck and Sikdar [20] also used convolutional neural networks (CNNs) to classify tweets according to four predefined categories (racism, sexism, both, and non-hate-speech). In contrast with previously mentioned works, Malmasi and Zampieri [21] applied a linear support vector machine (SVM) to annotate tweets with one of three tags ("hate," "offensive," or "ok"). Additionally, to improve the performance of hate speech detection, Qian et al. [22] introduced intra- and interuser representation learning by considering user's historical posts and reinforcing them through similarity with all other users. The proposal employed bidirectional long short-term memory (Bi-LSTM) and a gradient-based deep reinforcement learning model.

In the context of terrorism and radicalization, Araque and Iglesias [7] explored the emotional characteristics and semantic similarity for the detection of radicalization in online newspapers and social media, thus classifying users with both logistic regression and linear support vector machines (SVM). Nouh et al. [23] also use propaganda magazines to build a radical corpus (with TF-IDF scores and

Table 1: Application of NLP in cybersecurity and cyber defense.

Work	Field	Scenario	Goal	Application of NLP	Complement techniques
Tamura et al. [13]	Cybersecurity	Industrial control networks	Detect anomalies in packet flow	Similarity model	Markov chain model
Chambers et al. [14]	Cybersecurity	Twitter	Detect cyberattacks and analyze user behavior	Continuous bag-of-words (CBOW) model and topic-based model (PLDA)	×
Khandpur et al. [15]	Cybersecurity	Twitter	Detect cyberattacks in social media	Similarity model, domain generation algorithm, and dynamic query expansion	×
Ritter et al. [16]	Cybersecurity	Twitter	Detect cyberattacks in social media	Name-entity recognition	Expectation regularization
Kong et al. [17]	Cybersecurity	Google Play	Evaluate the security of Android apps through user reviews	Bag-of-words (BOW) + classifier (sparse SVM)	Crowdsourcing techniques
Liao et al. [18]	Cybersecurity	Technical articles	Discovery indicators of compromise	Dependency parsing and topic extraction (POST)	Classifier (SVM), classifier (logistic regression), and graph mining
Pereira-kohatsu et al. [9]	Hate crime	Twitter	Identify and monitor hate speech	Long short-term memory NN + multilayer perceptron	Classifiers (LDA, QDA, RF, RLR, and SVM)
Muhammad et al. [19]	Hate crime	Twitter	Classify messages as hate speech, offensive, or nonoffensive	Sequential CNN (SCNN)	×
Gambäck and Sikdar [20]	Hate crime	Twitter	Classify tweets as "racist," "sexist," "both," or "non-hate-speech"	Convolutional NN (CNN)	×
Malmasi and Zampieri [21]	Hate crime	Twitter	Annotate tweets with labels "hate," "offensive," or "ok"	Linear SVM	×
Qian et al. [22]	Hate crime	Twitter	Analyze real-life extremists and hate groups	Bidirectional LSTM (bi-LSTM) + deep reinforcement learning	×
Araque and Inglesias [7]	Radicalization	Twitter and online newspapers	Categorize radical users	Sentiment analysis and similarity model	Logistic regression and linear SVM
Nouh et al. [23]	Radicalization	Twitter	Categorize radical tweets	Language model and sentiment analysis	Classifiers (RF, NN, SVM, and KNN)
Chen [24]	Radicalization	Dark web	Categorize forum postings	Ensemble SVR	Clustering
RED-Alert [25]	Radicalization	Social media	Monitor social networks in real time	Semantic analysis, lexical analysis, and domain-specific ontologies	Social network analysis and complex event processing
Iqbal et al. [26]	Cybercrime	Chat logs	Summarize conversations into crime-related topics	Named-entity recognition, semantic analysis, similarity model	Information visualizer
Pastrana et al. [27]	Cybercrime	Underground forums	Detect cybercrime topics and identify potential victims	Logistic regression and topic extraction	Social network analysis and clustering (K-means)
Bhalerao et al. [28]	Cybercrime	Underground forums	Analyze posts and replies for the identification of supply chains	Classifiers, (FT, LR, SVM, and XGBoost)	×
Our work	Cybercrime	Twitter	Identify suspect groups	Similarity model and sentiment analysis	Clustering (K-means) and graph mining

word embedding) and infer their psychological/behavior signals. The resulting features were tested in different classifiers to categorize a sample of tweets as radical or not, with a random forest and a neural network achieving the best performance. Chen [24] incorporated dark web forums to measure radicalization, designing an approach through an ensemble support vector regression (SVR) to infer whether a forum posting presents "violence," "anger," "hate," or "racism." The H2020 European RED-Alert project [25] is an ambitious software toolkit to support LEAs in the fight

against online propaganda, recruitment, or mobilization of members, among other terrorist operations. The latter includes powerful modules of NLP, social networks analysis (SNA), and complex event processing (CEP).

In terms of cybercrime, Iqbal et al. [26] designed a WordNet model based on named-entity recognition, semantic analysis, and similarity models to identify and extract forensically relevant information from large suspicious chat logs. Pastrana et al. [27] applied logistic regression, social network analysis, clustering, and topic extraction to forum postings for characterizing cybercriminal trends and detecting potential victims to prevent at an early stage. Bhalerao et al. [28] also explored underground forums, specifically for the discovery of cybercrime supply chains. They tested different classifiers (Facebook AI FastText (FT), logistic regression (LR), support vector machine (SVM), and gradient boosted trees (XGBoost)) to label posts within product categories (such as "malware," "botnet," and "DDoS services," among others) and categorize replies according to their type ("buy," "sell," and "other").

As described above, several approaches have employed NLP to address problems of cybersecurity, hate crime, radicalization, or cybercrime. Yet we observe that there is not a unique framework to adopt against these threats, and authors design solutions depending on the specific field, scenario, and goal of the case study. Generally, the application of NLP is not enough, and it is therefore usually complemented with other AI-based or data-oriented techniques. In this regard, the paper at hand intends to detect and monitor violent movements in Twitter, proposing a combination that we have not witnessed in the literature, employing similarity models and sentiment analysis to identify aggressive tweets, and applying clustering and social network analysis to infer groups of suspect users that write related content.

3. Data Science Life Cycle Based on NLP Models

The data science life cycle encompasses the following phases [29]: (i) business understanding, (ii) data acquisition, (iii) modeling, and (iv) deployment, and offers a high-level perspective over the actions that must be developed to build a functional data science solution. This particular section shows our proposal of application of these phases of the data science life cycle in the construction of our solution, aimed to uncover cybercrimes in social media through NLP models.

3.1. Business Understanding. The power of social networks has increased in the last few years due to the freedom of expression that people exhibit on such social platforms. Also, social networks allow a user to find peers with similar tastes and even promote the creation of groups [30]. The heyday of social networks has provoked numberless groups to appear that contain a diversity of information that is interesting for a data scientist, for example, to make inferences and detect patterns [31].

Thus, social networks become a rich source of data [32] containing information about the features of a user contained in the account profile and information about the thoughts of a user, implicitly included in the tweets and the user activity. However, some users may have a deeper interest in creating content that promotes violence such as social revolt, cyberbullying, harassment, and even conspiracy to produce harmful outcomes related to their particular interests, influencing other people's beliefs and behavior [33]. This phenomenon in information warfare is called "hostile social manipulation" [11, 34].

In this context, the objective of this research is to develop an NLP solution capable of analyzing social networks accounts promoting violent activities so that LEAs can improve their efforts in crime prevention. This solution is oriented to achieve the following specific goals:

(i) It should exhibit the relations between multiple suspicious users

(ii) It should offer an analysis of suspicious tweets in terms of similarity and polarity

(iii) It should be applied in contexts like the ones composed by Spanish-speaking countries

(iv) It should accelerate the response of LEAs to achieve cybercrime prevention.

3.2. Data Acquisition. Among all social networks, Twitter is one of the most used ones to share opinions and information and even create movements with political, social, or economical interests, becoming a big data source. Twitter exposes a great power of communication between ordinary people, which is evidenced by the increase of users' accounts and tweets over the last few years [35]. About 500 million tweets are sent on Twitter per day, and 350,000 tweets are sent per minute (https://www.omnicoreagency.com/twitter-statistics/), demonstrating that Twitter is very active in sharing opinions, and therefore, it can be very useful for local enforcement authorities to monitor unusual social behavior within cyberspace. For all these above-mentioned reasons, Twitter was the social network selected as the provider of the raw data that feed our proposal.

Additionally, tweets need to be vectorized in order to be processed, so it is also important to count on an embedding data set that contains the vector representation for most common words in a given language. It is important to have such a large number of words with their respective embeddings since this ensures that most of the words inside the tweets will have a representation of numerical vectors that can later be used by NLP algorithms. The vectorization of tweets requires using deep learning architectures, for example, continuous bag-of-words (CBOW) [36] and continuous skip-gram [37], that learn the vector representation of words from a training text. In CBOW, the order of the context words does not matter, and the words are predicted from their local context where a neighborhood parameter is defined. In skip-gram, the context is predicted from the word, and the local neighborhood parameter is randomly sampled from a uniform discrete distribution over a fixed range [38].

3.3. Modeling. The methodology of the proposed solution is shown in Figure 1, which depicts the main steps that were considered in order to guarantee a pipeline that receives the raw data composed by the gathered tweets, cleans and translates all the tweets of interest, processes the tweets through different NLP models, and obtains actionable information that may be used by LEAs to analyze the scenario of a presumable cybercrime.

3.3.1. Preprocessing. The tweets to be analyzed should be first cleaned of hashtags, mentions, and URLs, in order to avoid that the model built in the next phase gets confused with no regular words. A process of normalization should also be applied to convert all the text in lowercase with the aim of avoiding that two words with the same meaning but with different cases that may be considered as different. Additionally, emojis may be important as part of the meaning of a tweet published by a user, so these should not be removed, and conversely, these should be converted to a phrase that represents their meaning. After tweets are cleaned and all their meaningful pieces have been converted to text, these should be translated to the language used in the embedding process, for example, English language.

3.3.2. Processing in Terms of Similarity. Next, tweets must be vectorized using the embedding data set selected in Section 3.2. In this regard, each tweet is converted into a single vector that in turn constitutes the average of the vector representation of each word composing the tweet. Then, the collected tweets are represented by a matrix $T_{n\times m} = (t_1, \ldots, t_n)$, where m is the dimension of the vector and n is the number of tweets. Then, $T_{n\times m}$ is processed with the aim of building a matrix of similarities, where the element (i, j) of the matrix stores the cosine distance between tweet t_i and tweet t_j. The cosine distance $\cos(\theta)$ is the cosine of the angle θ between two vectors u and v and can be represented using the dot product and the magnitude of the vectors as observed in the following equation:

$$\cos(\theta) = \frac{u \cdot v}{\|u\|v\|}. \tag{1}$$

Additionally, and as a complementary outcome, a validation data set is built with the purpose of testing how well the similarity model is able to rank similar tweets. For this purpose, the determination of a ranking of similarities is conducted taking each tweet (t_i) and calculating a ranking according to its similarity against the remaining $n - 1$ tweets. Then, for each tweet (t_i), the most similar tweet t_s and two less similar tweets (t_p, t_q) randomly selected between the less similar tweets of the ranking are identified. Thus, a validation data set that contains for each row the following structure $\{t_i, t_s, t_p, t_q\}$ is composed. This data set needs to be reviewed and adjusted manually row by row in order to create a proper validation data set. In turn, the test of the similarity model in terms of ranking similar tweets may be done through the metrics Hits [39] and discounted cumulative gain (DCG) [40].

Hits@K is a metric that calculates the number of hits, that is, tweets found as similar to a tweet t_i by a similarity model for some K as shown in equation (2), where the term $\text{top}\,K(t_i)$ represents a set with the actual K tweets that are more similar to a tweet t_i. Iverson bracket notation is used for the term inside the sum, so dup_i represents a function that takes two possible values: 1 or 0, being 1 if the tweet found by the similarity model is in top K set or 0 in the other case.

$$\text{Hits@}K = \frac{1}{N}\sum_{i=1}^{N}[\text{dup}_i \in \text{top}\,K(q_i)]. \tag{2}$$

On the other hand, DCG@K or the discounted cumulative gain is a measurement that finds the relevance or similarity of a tweet with another, and we can observe in equation (3). This metric receives a ranked tweets list, which is denoted by rank_{dup}. The order of the tweets in the rank list is important because if a tweet has less similarity with the tweet under analysis, the DCG metric expects that such tweet is located further away from the list. The log function scales the relevance of each tweet.

$$\text{DCG@K} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\log_2\left(1 + \text{rank}_{\text{dup}_i}\right)} \cdot \left[\text{rank}_{\text{dup}_i} \le K\right]. \tag{3}$$

The principal difference between the metrics Hits and DCG is as follows: while in Hits the interest is to validate whether a tweet is in the set of similar tweets or not, in DCG the interest is also on the order that a tweet has into such set. Both metrics should be applied to validate the capability of the similarity model to find similar tweets.

3.3.3. Clustering. The matrix of similarities obtained in the previous step is then used to make clusters of similar tweets. There are different types of clustering algorithms that we can use, for instance, spectral clustering [41], Gaussian mixture [42], and K-means method [43]. Spectral clustering is a graph theory technique that uses eigenvalues to compute a graph and to find connections using edges. On the other hand, Gaussian mixture groups data that belong to a similar Gaussian distribution, and K-Means uses the Euclidean distance to build clusters. Additionally, the optimal number of clusters is determined by the Elbow method [44].

Then, each tweet in each cluster may be analyzed with a model for sentiment analysis [38] that identifies if the tweet reflects positive, negative, or neutral feelings. Different algorithms exist to make sentiment analysis such as Bernoulli or Naive Bayes, which uses the Bayes theorem [45] to interpret the meaning of a message. Another algorithm used for the same purpose is the single-layer perceptron (SLP), which is an artificial neuronal network that makes classification in a binary way using a linear separation [46]. Moreover, there are more basic algorithms like the Vader rule-based model, which is used in the VaderSentiment (https://pypi.org/project/vaderSentiment/) python library, which classifies phrases making a sum of the polarity of each word according to its semantic meaning [47]. The main outcomes obtained from these algorithms are polarity and subjectivity.

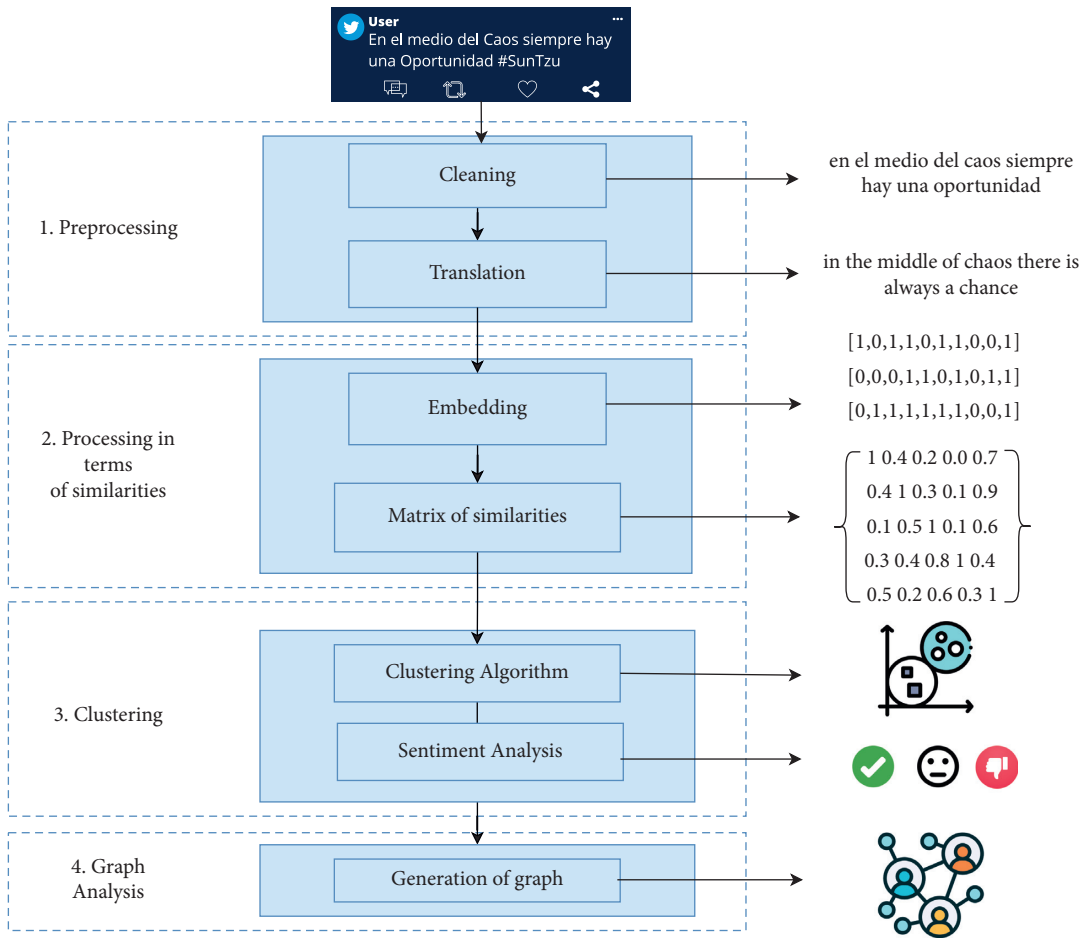

Figure 1: Methodology of the proposed solution.

Polarity allows to identify the sentiment as positive or negative in terms of aggressiveness, and it is represented by a number that lies in the range of $[-1, 1]$, where 1 depicts an extremely positive statement and -1 depicts an extremely negative statement. Tweets around -1 in polarity generally contain offensive opinions, most of them employing bad words; tweets around 1 in polarity represent positive statements; and tweets around 0 in polarity indicate neutral opinions. On the other hand, subjectivity refers to the existence of a personal opinion, emotion, or judgment in a sentence, as opposed to objectivity, which refers to factual information. Subjectivity is represented by a number that lies in the range of $[0, 1]$, where 0 means factual information and 1 means subjective opinion. Tweets around 1 in subjectivity generally refer to people who are very passionate about communicating opinions. Thus, this sentiment analysis allows extracting the polarity (negative, neutral, and positive) and the subjectivity (actual and subjective) of the tweets composing each cluster.

A word map may be built for each cluster to identify the words that are more predominant. Such word maps are generally made from the creation of a list of frequencies of the words that compose the tweets of each cluster. The words with the highest frequency will be the most predominant in the word map. At last, the cluster with the most negative polarity on average is chosen for a deeper analysis through a graph analysis.

3.3.4. Graph Analysis. Once the aggressive users are grouped in a cluster, the most aggressive users within such a cluster should also be identified. Since the level of aggressiveness of each tweet is determined by the polarity, the most violent creators of tweets can be identified through outlier detection techniques. Visualization techniques such as histograms, box plots, and scatter plots are useful for outlier detection as well as interquartile range (IQR). Finally, the presence of outliers could be validated with statistical tests such as the Grubbs, Chi-square, and Dixon Q tests.

User accounts and tweets identified previously need to be prepared and enriched before building the graph. First, accounts information such as number of followers and followings, accounts with mutual relationships (follower and following), profile picture, account creation date, Twitter ID, number of tweets sent, and average number of tweets per day could be obtained and analyzed. This information can be obtained with cyber intelligence tools such as SpiderFoot (https://www.spiderfoot.net/), Maltego (https://www.maltego.com/), or TinfoLeak (https://tinfoleak.com/). All this information is then exported to a table and is

reorganized such that each row represents a connection between two users, so it can be entered into the graph builder, for example, Gephi.

Graph theory has been traditionally employed to analyze the interactions between users and to detect communities of users [48]. So the most polarized cluster determined from the previous section is analyzed to identify accounts related between them by some type of connection (direct relation) and accounts related through a third account (indirect relation). Relations between users are represented by a directed graph where each node represents a user account, and each edge establishes the type of connection ($\mathbb{Z}$ follows B, B follows A, A follows B and B follows A) between those two user accounts.

Filters applicable in a graph are generally based on graph centrality measurements such as node degree, eigenvector centrality, or PageRank [49]. The most common centrality measurement is node degree that considers the number of neighbor nodes to determine the importance of a node in a graph; however, it does not consider the own connectivity of the neighbors. On the other hand, eigenvector centrality not only computes the degree of a node but does consider the number of connections of its neighbors. However, eigenvector centrality can introduce a hub bias when a first node that has very few connections connects to a node with many neighbors in a hub, pointing out that such a first node is apparently important in that hub, but this is not necessarily true. To eliminate this last bias, PageRank centrality considers the direction of the connections between nodes or users and assigns greater importance to nodes with a higher input degree. In this way, irrelevant users may be eliminated using some of these previously described measurements, and information on the most suspicious users may be simplified.

3.4. Deployment. The proposed solution is intended to operate as a key information system for LEAs, which can be consulted constantly to obtain valuable intelligence information. This solution should have high availability and resiliency as its operation would be essential to guarantee proactive monitoring of anomalous activities in social networks and would address in real-time actions to prevent cybercrimes.

4. Experiments

This section contains the results obtained from applying the proposal described in Section 3 in two different scenarios related to some protests that occurred in 2020 in Colombia and the United States, being the data and code available at the project repository (https://github.com/alejandrarchbold/NLP-Model-for-prevention-of-Cybercrimes). In both cases, Twitter was the social network used to provide the raw information to be processed. The gathering was done in both cases using TAGS (https://tags.hawksey.info/), which is an application focused on the collection of tweets that allows to set up and run an automatic collection using different query operators along a period of 7 days. A period of a few days may be considered short; however, the period for a collection depends on the specific campaign that is being analyzed; for example, some campaigns exist only for the day of a notable commemoration or planned event.

The embedding process was based on the use of Google News Embedding and the tool word2vec (https://code.google.com/archive/p/word2vec/), which contains generic embeddings for 3,000,000 English words; each one of them represented in 300-dimensional vectors. word2vec provides an implementation of the deep learning architectures CBOW [36] and skip-gram [37] for computing vector representations of words. This specific embedding was chosen because of its size and quality, which have made it one of the most used embedding data sets. To improve the analysis, the follower and followed accounts for the accounts included in the most aggressive cluster were extracted using the cyber intelligence tool TinfoLeak (https://tinfoleak.com/). Finally, the tool Gephi (https://gephi.org/) was used for the building of a social network graph for the cluster of interest in both scenarios.

4.1. Scenario 1: Protests against Corruption in Times of COVID-19 in Colombia. This scenario implied the gathering of 17,454 tweets containing the hashtag #Marcha15deJunio (#ProtestJune15th) that, after removing retweets, were reduced to 1,287 tweets from 880 accounts. These tweets referred to the national protest of June 15, 2020, in Colombia addressed mainly against different actions of corruption discovered during the COVID-19 quarantine, plus some national controversial cases of police abuse (https://www.lafm.com.co/bogota/en-vandalismo-acabo-marcha-por-la-vida-digna). For this scenario, the collection of tweets was done in the previous days of the protest, between May 28, 2020, and June 3, 2020.

Tweets were preprocessed and cleaned properly to be consumed by the similarity model that will be used later in the pipeline. The first step in preprocessing was to remove URLs, mentions, and hashtags. The second step was to convert the characters of all the tweets into lowercase. Then, emoticons were replaced by their meaning in words through the use of the Python library emoji (https://pypi.org/project/emoji/). After preprocessing, empty tweets were removed and a total of 1,105 tweets remained, which were translated from Spanish to English using Google API services (https://pypi.org/project/google-cloud-translate/). The purpose of this translation was to uniform the language to the one used by Google News Embedding, to be able to vectorize the Tweets.

Then, the collected tweets $(t_m = t_1, \ldots, t_n)$ were processed as indicated in Section 3 to obtain a training data set composed of a tweet (t_i), the most similar tweet (t_s) and two other randomly selected tweets (t_p, t_q) among the less similar tweets of the ranking. Thus, a data set that contains for each row the following structure $\{t_i, t_s, t_p, t_q\}$ was composed, having a total of 1,105 rows. The data set was reviewed and adjusted manually row by row to create a proper validation data set. The validation data set and the original data set were compared to verify the correctness of the model, getting the results shown in Table 2, which shows

Table 2: Metrics DCG and Hits for Scenario 1.

Interactions	DCG	Hits
1	0.576	0.576
5	0.585	0.594
10	0.591	0.615
100	0.640	0.860
500	0.658	1.000
1000	0.658	1.000

that the similarity model gets better results in the Hits metric in comparison with the DCG metrics when the training interactions increase.

Next, the collected tweets $(t_m = t_1, \ldots, t_n)$ were processed by the similarity model mentioned in Section 3 to build a matrix of cosine distances. The determination of such a matrix was done by taking each tweet (t_i) and calculating a ranking of similarities against the remaining $n - 1$ tweets. Afterward, the optimal number of clusters of tweets is determined by the elbow method for clustering (see Figure 2), where the optimal number of clusters was 4. Thus, the 1,105 tweets were split into four clusters using the PCA (principal components analysis) algorithm [50] to decompose the data variance into two components to finally create groups according to the following clustering algorithm: K-means, spectral clustering, and Gaussian mixture.

Additionally, the Calinski–Harabasz (CH) value was used to identify the most accurate cluster algorithm [51]. CH value stands for the ratio between the within-cluster dispersion and the between-cluster dispersion, where a higher CH value shows a better clustering in the data. Table 3 shows the results for the clustering algorithms K-means, spectral clustering, and Gaussian mixture, pointing that K-means is the one with the better results. Thus, K-means was selected as the clustering algorithm getting the results shown in Figure 3.

Furthermore, sentiment analysis was applied to each cluster to find the predominant polarity. Figure 4 shows the word map for the four clusters. Most of the words are related to the protests against the government in Colombia and the intention to take to the streets despite the pandemic of COVID-19. The words are also a response to what people think about the trend of that moment: racism, deliberated corruption, and threats to social leaders, among others. The sentiment analysis was made for the four clusters using two python libraries: TextBlob (https://pypi.org/project/textblob/) that employs a single-layer perceptron (SLP) algorithm and VaderSentiment (https://pypi.org/project/vaderSentiment/) that uses the Vader rule-based model. Table 4 shows the results of the execution of these two algorithms over the four preciously identified clusters. Both clustering algorithms agree that cluster 4 has higher percentage of negative tweets. Additionally, Figure 5 shows the positive, negative, and neutral tweets according to the SLP algorithm per cluster. Thus, from the polarity and subjectivity analysis, cluster 4 can be identified as the most aggressive one, as it contains the biggest amount of negative tweets.

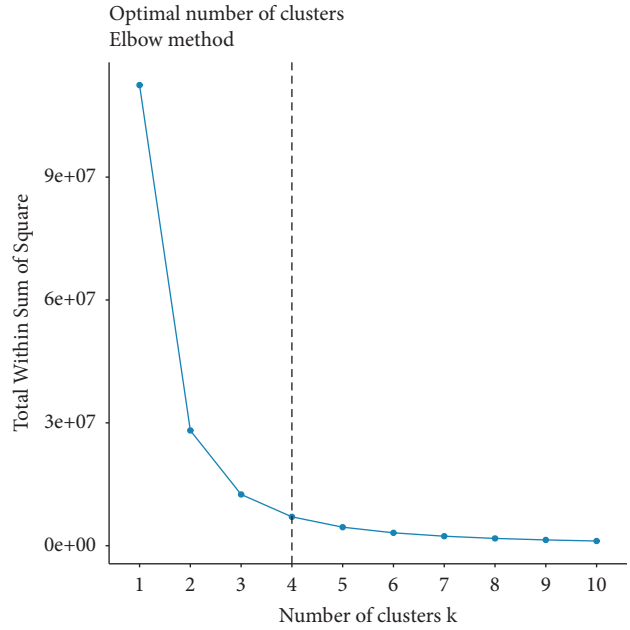

Figure 2: Optimal number of clusters with the elbow method for Scenario 1.

Table 3: Calinski–Harabasz values for different clustering algorithms for Scenario 1.

Clustering algorithm	Calinski–Harabasz value
K-means	1088.61
Spectral clustering	44.57
Gaussian mixture	1087.60

Hence, Twitter users' accounts from tweets contained in cluster 4 were extracted to get a total of 161 suspicious users. This set of suspicious users was reduced to only consider active user accounts associated with tweets with a polarity level lower than −0.3, that is, $-1 <= \text{polarity} < -0.3$, to get a total of 36 user accounts. Users associated with such accounts may be considered the authors of the most aggressive tweets, so a cyber intelligence analysis, over each identified user, was done using the tool TinfoLeak, which allowed to identify some details for each Twitter account: followers, accounts following (friends), accounts with mutual relations, profile image, account creation date, name on Twitter, full username, description of the account, Twitter ID, number of tweets sent and average number of tweets per day, number of likes, number of lists, reported location, time zone, and idiom, among others.

The information obtained from the cyber intelligence analysis performed over the 36 user accounts was used to build a relationship graph through the tool Gephi (see Figure 6). The graph was filtered by those users who had at least two relationships, to focus the analysis over users with some relation with others instead of solitary nodes. Such filter was implemented through the methodology K-core that gets a maximal subgraph where all vertices of a node are connected and have a degree of at least k [52]. Users of cluster 4 are represented as yellow nodes, while

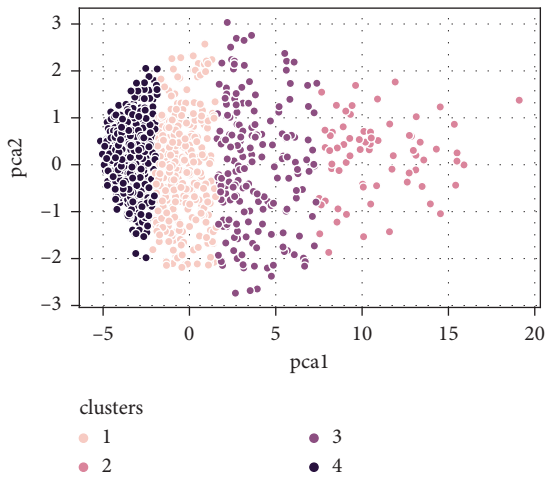

Figure 3: K-means clustering for Scenario 1.

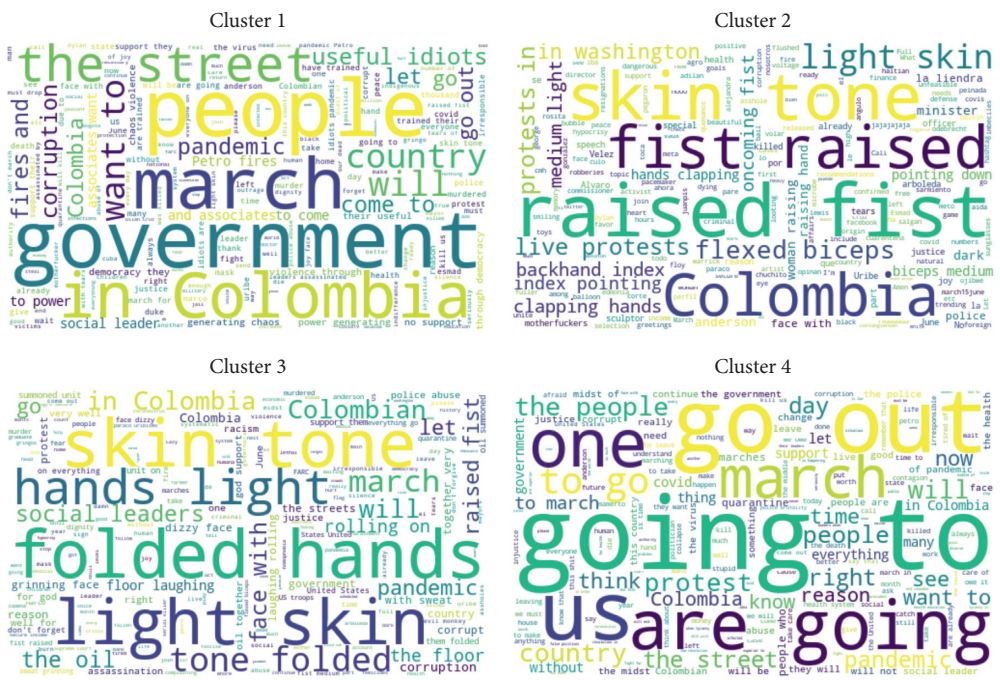

Figure 4: Word map for the clusters of Scenario 1.

Table 4: Sentiment analysis results for Scenario 1 using SLP algorithm versus Vader model.

Sentiment analysis algorithm	Cluster	Polarity			Subjectivity (mean)	Number of tweets	Number of accounts
		Negative (%)	Neutral (%)	Positive (%)			
SLP algorithm (TextBlob library)	1	29.8	42.1	27.9	0.29	368	302
	2	13.6	59.1	27.2	0.20	88	73
	3	17.3	57.1	25.5	0.22	196	167
	4	35.5	22.2	42.1	0.43	453	362
Vader model (VaderSentiment library)	1	49.4	23.3	27.1	—	368	302
	2	18.1	59.0	22.7	—	88	73
	3	40.8	36.2	22.9	—	196	167
	4	54.8	8.1	36.6	—	453	362

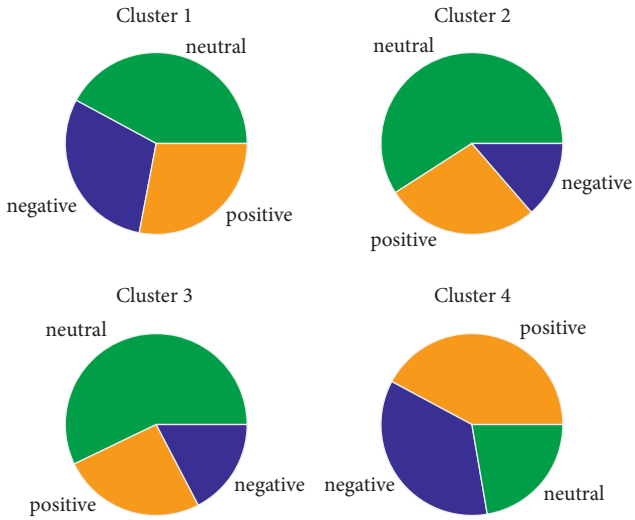

Figure 5: Polarity analysis for the four clusters of Scenario 1 using SLP algorithm.

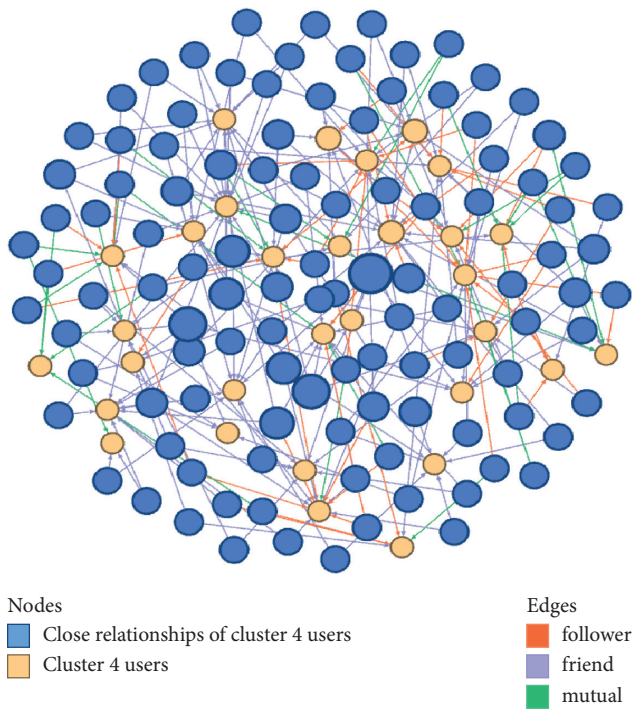

Figure 6: Neighborhood for accounts included in cluster 4 of Scenario 1.

followers, friends, and following users are represented as blue nodes. The size of the node is defined according to the number of outputs (out-degree). The color of the edges represents the relationships between nodes (follower, friend/following, and mutual). For this scenario, there were 141 nodes and 256 edges. Particularly, this graph allows to identify relations between suspicious accounts and even identify new user accounts that are related to the suspicious ones.

4.2. Scenario 2: Black Lives Matter Movement in the United States. This second scenario implied the collection of 18,741 tweets with the hashtag #blm related to the protests in the United States against racism and police abuse in the case of the death of George Floyd. The initial set of tweets was reduced to 1,287 tweets, from 1,131 users accounts, after eliminating retweets to just identify creators of content. The mentioned hashtag refers to the movement "Black Lives Matter" that aims to eradicate white supremacy and build local power to intervene in violence inflicted on black communities. The collection of tweets in this scenario was done on July 15, 2020, when a video appears showing the moments leading up to George Floyd's death (https://edition.cnn.com/2020/07/15/us/george-floyd-body-cam-footage/index.html).

The tweets were preprocessed and cleaned in the same way as the previous scenario (removal of URLs, mentions, and hashtags, conversion to lowercase, and emoticon replacement). After cleaning, a total of 1,207 tweets remained. Then, the non-English tweets were translated into English using the Google API services to be in the same language as Google News Embedding, and vectorization of all tweets was done.

Subsequently, the collected tweets $(t_m = t_1, \ldots, t_n)$ were processed to obtain a training data set composed of 1,207 rows with the structure $\{t_i, t_s, t_p, t_q\}$, where (t_i) is a tweet belonging to t_m, (t_s) is the most similar tweet, and (t_p, t_q) are two randomly selected tweets among the less similar tweets (negative tweet) of the ranking. In order to create a proper validation data set, this training data set was reviewed and adjusted manually. Then, the metrics Hits and DCG were calculated to compare the validation data set and the original data set and verify the correctness of the model, as shown in Table 5.

Then, a matrix of cosine distances was built through the application of the similarity model mentioned in Section 3 over the collected tweets $(t_m = t_1, \ldots, t_n)$. Such a matrix was done taking each tweet (t_i) and calculating a ranking of similarities against the remaining $n-1$ tweets. For this scenario, the optimal clusters of 4 were also determined by the elbow method for clustering, as observed in Figure 7. The four clusters were calculated using the PCA algorithm that identifies the principal components (pca1 and pca2) of the items of the matrix of cosine distances, and then the same clustering algorithm (K-means, spectral clustering, and Gaussian mixture) that were applied in Scenario 1 was applied.

Similarly to Scenario 1, we selected the best clustering algorithm according to the Calinski–Harabasz [51] values shown in Table 6. K-means obtains again the best results and is selected as the clustering algorithm for this current scenario, getting the results illustrated in Figure 8.

As in Scenario 1, sentiment analysis was carried out to each cluster, using the TextBlob and the VaderSentiment python libraries. Table 7 shows results obtained from both these libraries, pointing to cluster 2 as the one with the highest percentage of negative tweets. Word maps were also built to identify the most predominant words within each cluster, as depicted in Figure 9. The word maps of the "Black

Table 5: Metric DCG and Hits for Scenario 2.

Interactions	DCG	Hits
1	0.493	0.493
5	0.583	0.792
10	0.624	0.946
100	0.635	1.000
500	0.641	1.000
1000	0.642	1.000

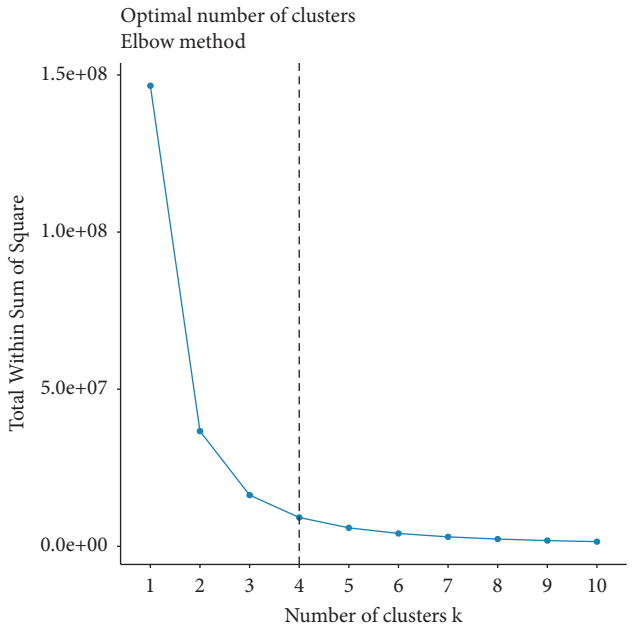

Figure 7: Optimal number of clusters with the elbow method for Scenario 2.

Table 6: Calinski–Harabasz values for different clustering algorithms for Scenario 2.

Clustering algorithm	Calinski–Harabasz Value
K-means	962.68
Spectral clustering	29.35
Gaussian mixture	962.61

Lives Matter" movement show different popular words such as racism, police abuse, and status in society. Finally, Figure 10 shows the proportion of positive, negative, and neutral tweets according to the SLP algorithm per cluster.

Tweets contained in cluster 2 are associated with 201 user accounts, so this set was reduced to 38 accounts by choosing only accounts that produced tweets with polarity between −1 and −0.3. Then, a cyber intelligence analysis was performed on those accounts through the tool TinfoLeak to obtain different features such as followers, accounts following (friends), and accounts with mutual relations, among others. Then, a full social network graph was built using Gephi (see Figure 11). Also, the graph was filtered to consider only accounts with at least two relationships through the application of a K-core filter. Cluster 2 users are represented as green nodes, while followers, friends, and following users are

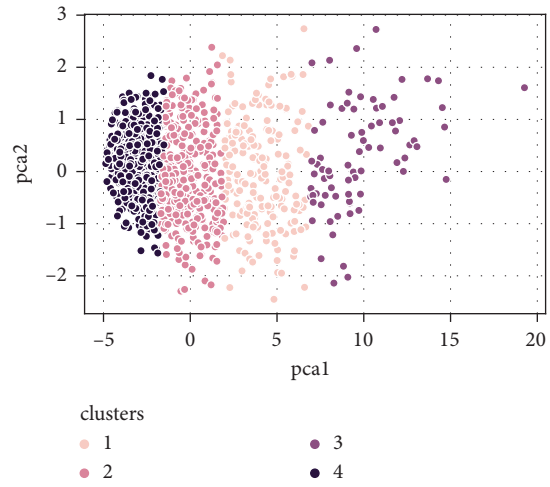

Figure 8: K-means clustering for Scenario 2.

represented as blue nodes. The size of the node is defined according to the out-degree. The color of the edges represents the relationships between nodes (follower, friend, and mutual), getting a total of 81 nodes and 124 edges. In the end, relations between suspicious nodes belonging to cluster 2 may be seen in the graph, and even new nodes that were not considered initially in cluster 2 may also pinpoint as some of them may be closely related with many suspicious nodes.

5. Application of NLP Models in a National Cyber Defense Strategy

In order to counteract the effects generated by HSM campaigns, LEAs must understand in depth how the campaigns they are facing are actually structured. One of the challenges that LEAs face is the systematic dissemination of information within that type of campaign [34]. That systematic dissemination generates large amounts of information that LEAs must process to understand the manipulation strategy [53]. There are two ways that could be efficient in containing the violent actions generated from HSM campaigns. The first is through the deployment of information operations that aim to mitigate the effects generated by disinformation actions generally used in the framework of HSM. The second is anticipating the physical points where these violent actions would happen and reinforce the security measures in those points. It is essential to identify the HSM actions in the shortest possible time; otherwise, it would be more challenging to achieve efficient containment.

The cases presented in Section 4 are examples of the initial work that an LEA analyst should develop to understand how the criminal groups organize the HSM campaigns. In both cases, NN-based NLP allows identifying key factors as similarity of information and the relationship between nodes and content polarity. All that information allows the LEAs to steer the analysis of the HSM campaign. In addition, those factors provide an analyst with information to build and support a hypothesis regarding the criminal structure behind the campaign they are facing. For example, the similarity between the collected tweets could

Table 7: Sentiment analysis results for Scenario 2 using SLP Algorithm versus Vader model.

Sentiment analysis algorithm	Cluster	Polarity			Subjectivity (mean)	Number of tweets	Number of accounts
		Negative (%)	Neutral (%)	Positive (%)			
SLP algorithm (TextBlob library)	1	13.4	65.8	20.7	0.19	82	81
	2	39.7	12.8	47.4	0.42	506	452
	3	32.9	21.4	45.6	0.37	410	355
	4	27.2	46.8	25.8	0.28	209	201
Vader model (VaderSentiment library)	1	20.7	52.4	26.8	—	82	81
	2	55.1	4.9	39.9	—	506	452
	3	41.7	14.6	43.6	—	410	355
	4	29.1	38.7	32.0	—	209	201

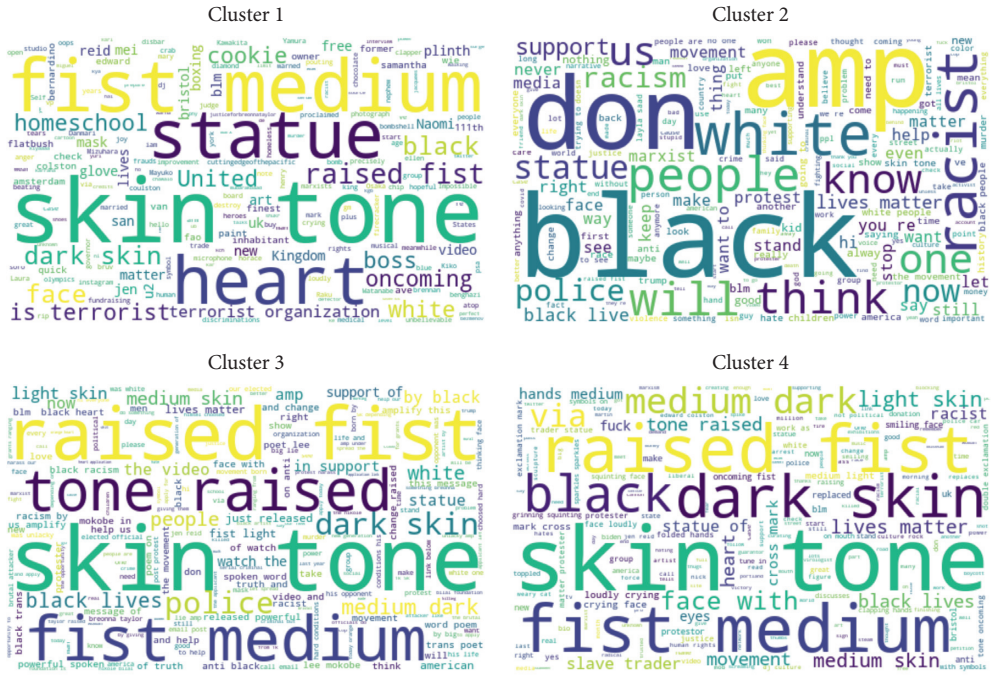

Figure 9: Word map for the clusters of Scenario 2.

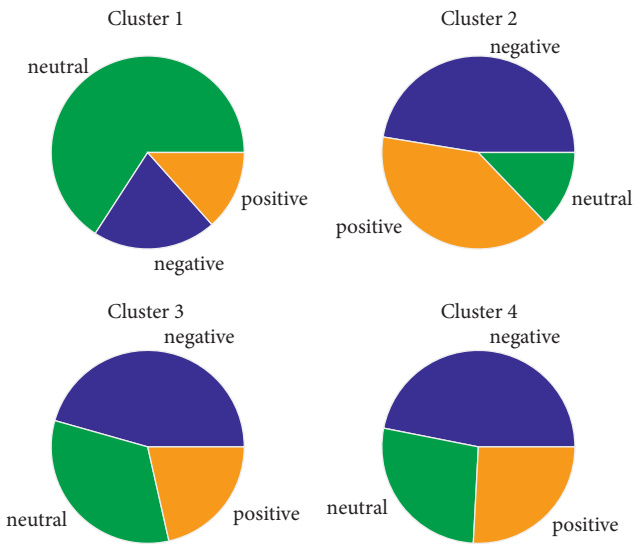

Figure 10: Polarity analysis for the four clusters of Scenario 2 using SLP algorithm.

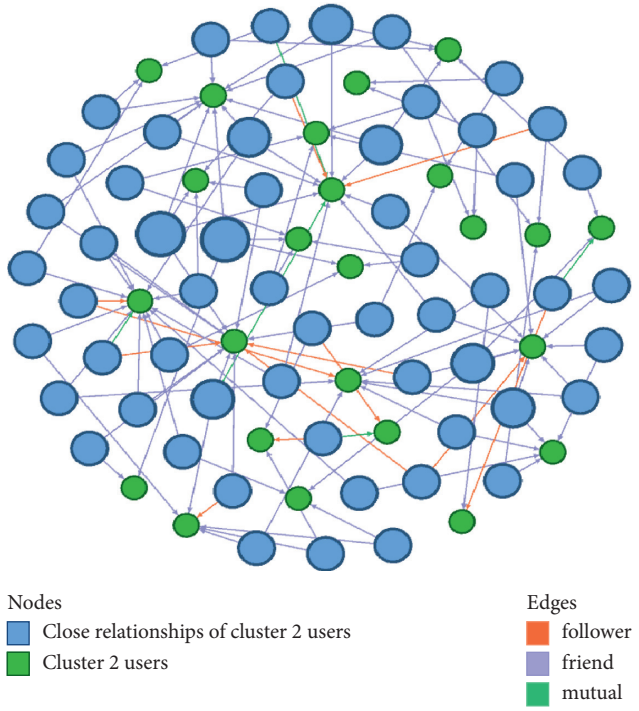

Figure 11: Neighborhood for accounts included in cluster 2 of Scenario 2.

indicate the rising of virtual communities generating and sharing potentially hostile information. The result of such analysis will allow orienting the operational efforts of LEAs to prevent and detect criminal actions behind the HSM campaigns.

The tweets collected in the two scenarios described in Section 4 include information related to two potential HSM campaigns. Only the application of NLP for the analysis of the information does allow to identify the criminal structure behind these types of campaigns. However, NLP is crucial in reducing analysis time. That time reduction would allow an LEA to better understand the structure of the HSM campaign they face. On the one hand, that understanding would allow deploying containment measures in less time, reducing the impact generated by HSM campaigns. On the other hand, the information analyzed and complemented with other means such as human intelligence or signals intelligence would allow linking people participating in manipulation actions, which would facilitate their prosecution.

6. Conclusions and Future Work

Deep learning and particularly NLP have proven their potential in the support of cybersecurity labors and particularly in the detection of cybercrimes. The adoption of NN-based NLP solutions by LEAs would strengthen a national cyber defense strategy reducing considerably the time of attention to cybersecurity incidents and providing LEAs with the capacity to detect and prevent HSM.

In this regard, the paper at hand proposed an NLP-based solution that uses a similarity model, implemented using deep learning architectures, to identify clusters of tweets and then determine their level of polarity to identify its aggressiveness. The most aggressive cluster is analyzed through a review of the relations between the nodes composing the cluster. Our proposal was applied in two different scenarios related to protests that occurred in 2020 in Colombia and the United States, obtaining a graph with suspected users and their respective relationships.

As future work, we plan to develop experiments gathering tweets for a longer period, for example, one month before and after the protest, which would allow us to seek some relation between the behavior exposed by suspect users the day of the protest and other events that occurred in close dates. This would allow us also to develop a deeper analysis of the hostile social manipulation in scenarios of interest and determine their evolution over time.

We also plan to extract more information related to the Twitter user accounts that belong to the most aggressive cluster through posts published in Twitter along a period previous and after the protest. This activity could also be correlated with activity from other social networks accounts to support the graph analysis phase of our proposal, allowing us to make a deeper analysis of advanced patterns adopted by specialized threats.

Data Availability

The data and code used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was partially funded by the Spanish Government (grant nos. FPU18/00304 and RYC-2015-18210) and cofunded by the European Social Fund. Also, this work has been supported by the Unit of Research and Innovation at the University of Rosario (Colombia) through the project "IV-TFA043–Developing Cyber Intelligence Capacities for the Prevention of Crime."

References

[1] S. Makridakis, "The forthcoming artificial intelligence (ai) revolution: its impact on society and firms," *Futures*, vol. 90, pp. 46–60, 2017.

[2] M. N. Prakash, O. M. Lucila, and W. C. Wendy, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[3] J. Ibañez, S. Rocha, D. D. Díaz-López, J. Pastor-Galindo, and F. Gómez, "C3-sex: a conversational agent to detect online sex offenders," *Electronics*, vol. 9, no. 11, 2020.

[4] J. Pastor-Galindo, M. Zago, P. Nespoli et al., "Spotting political social bots in twitter: a use case of the 2019 Spanish general election," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2156–2170, 2020.

[5] M. Hernandez, C. Pinzón, D. O. D Díaz- López, J. Garcia, and R. Pinto, "Open source intelligence (osint) in a colombian context and sentiment analysis," *Revista vínculos*, vol. 15, no. 2, pp. 195–214, 2018.

[6] O. Araque, G. Zhu, and A. Iglesias Carlos, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowledge-Based Systems*, vol. 165, pp. 346–359, 2019.

[7] O. Araque and C. A. Iglesias, "An approach for radicalization detection based on emotion signals and semantic similarity," *IEEE Access*, vol. 8, pp. 17877–17891, 2020.

[8] A. Roy, Y. Park, and S. H. Pan, "Learning Domain-specific Word Embeddings from Sparse Cybersecurity Texts," 2017, https://arxiv.org/abs/1709.07470.

[9] J. Pereira-Kohatsu, L. Q. Quijano-Sánchez, F. Liberatore, and M. Camacho-Collados, Detecting and monitoring hate speech in twitter," *Sensors*, vol. 19, no. 21, p. 4654, 2019.

[10] M. Wright, "Cyberbullying victimization through social networking sites and adjustment difficulties: the role of parental mediation," *Journal of the Association for Information Systems*, vol. 19, no. 2, pp. 13–123, 2018.

[11] M. J. Mazarr, A. Casey, A. Demus, and S Harold, "Hostile social manipulation present realities and emerging trends," Technical report, RAND Corporation, Santa Monica, CA USA, 2019.

[12] J. Pastor-Galindo, P. Nespoli, F. G. Mármol, and G. M. Pérez, "The not yet exploited goldmine of osint: opportunities, open challenges and future trends," *IEEE Access*, vol. 8, pp. 10282–10304, 2020.

[13] K. Tamura and K. Matsuura, "Improvement of anomaly detection performance using packet flow regularity in industrial control networks," *IEICE-Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 102.A, no. 1, pp. 65–73, 2019.

[14] N. Chambers, B. Fry, and J. McMasters, "Detecting denial-of-service attacks from social media text: applying nlp to computer security," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 1626–1635, New Orleans LA US, January 2018.

[15] R. P. Khandpur, T. Ji, S. Jan, G Wang, C. T. Lu, and N. Ramakrishnan, *Crowdsourcing Cybersecurity: Cyber Attack Detection Using Social Media*, Association for Computing Machinery, New York, NY, USA, 2017.

[16] A. Ritter, E. Wright, W. Casey, and T. Mitchell, "Weakly supervised extraction of computer security events from twitter," in *Proceedings of the 24th International Conference on World Wide Web, WWW '15*, pp. 896–905, Republic and Canton of Geneva, CHE, Florence, Italy, May 2015.

[17] D. Kong, L. Cen, and H. Jin, "Autoreb: automatically understanding the review-to-behavior fidelity in android applications," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15*, pp. 530–541, Association for Computing Machinery, New York, NY, USA, October 2015.

[18] X. Liao, K. Yuan, X. F. Wang, Z. Li, L. Xing, and R. Beyah, "Acing the ioc game: toward automatic discovery and analysis of open-source cyber threat intelligence," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 755–766, Association for Computing Machinery, New York, NY, USA, October 2016.

[19] M. U. S. Khan, A. Abbas, A. Rehman, and R. Nawaz, "Hateclassify: a service framework for hate speech identification on social media," *IEEE Internet Computing*, vol. 25, no. 1, pp. 40–49, 2021.

[20] B Gambäck and U. K. Sikdar, "Using convolutional neural networks to classify hate-speech," in *Proceedings of the First Workshop on Abusive Language Online*, pp. 85–90, Association for Computational Linguistics, Vancouver, Canada, August 2017.

[21] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP, 2017*, pp. 467–472, INCOMA Ltd, Varna, Bulgaria, September 2017.

[22] J. Qian, M. ElSherief, E. Belding, and W. Y. Wang, "Leveraging intra-user and inter-user representation learning for automated hate speech detection,"vol. 2, pp. 118–123, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, Association for Computational Linguistics, New Orleans, Louisiana, June 2018.

[23] M. Nouh, J. R. C. Nurse, and M. Goldsmith, "Understanding the radical mind: identifying signals to detect extremist content on twitter," in *Proceedings of the 2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 98–103, IEEE, Shenzhen, China, July 2019.

[24] H. Chen, "Sentiment and affect analysis of dark web forums: measuring radicalization on the internet," in *Proceedings of the 2008 IEEE International Conference on Intelligence and Security Informatics*, pp. 104–109, IEEE, Taipei, Taiwan, June 2008.

[25] M. Florea, C. Potlog, and P. Pollner, "Challenges in Cybersecurity and privacy - the European research landscape," *chapter Complex project to develop real tools for identifying and countering terrorism: real-time early detection and alert system for online terrorist content based on Natural Language processing, Social Network Analysis, Artificial Intelligence and Complex Event Processing*, River Publishers, Denmark, Europe, pp. 181–206, 2019.

[26] F. Iqbal, B. C. M. Fung, M. Debbabi, R. Batool, and A. Marrington, "Wordnet-based criminal networks mining for cybercrime investigation," *IEEE Access*, vol. 7, pp. 22740–22755, 2019.

[27] S. Pastrana, A. Hutchings, A. Caines, and P. Buttery, "Characterizing eve: analysing cybercrime actors in a large underground forum," in *Michael Bailey, Thorsten Holz, Manolis Stamatogiannakis and Sotiris Ioannidis*, pp. 207–227, Springer International Publishing, NewYork, NY, USA, 2018.

[28] R. Bhalerao, M. Aliapoulios, I. Shumailov, S. Afroz, and D. McCoy, "Mapping the underground: supervised discovery of cybercrime supply chains," in *Proceedings of the 2019 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–16, IEEE, Pittsburgh, PA USA, November 2019.

[29] G. Ericson, W. Rohm, and J. Martens, *Team Data Science Process Documentation*, Microsoft Azure, Technical Report, 2017, https://docs.microsoft.com/en-us/azure/architecture/data-science-process/overview.

[30] D. Tambini and Media Freedom, Hoboken, NJ, USA, 2021, https://books.google.com.co/books?id=J2I9EAAAQBAJ&dq=D.+Tambini.Media+Freedom.+Wiley,+2021.&source=gbs_navlinks_s.

[31] T. Toivonen, V. Heikinheimo, C. Fink et al., "Social media data for conservation science: a methodological overview," *Biological Conservation*, vol. 233, pp. 298–315, 2019.

[32] M. James, N. Thompson, K. Lee, K. W. Wong, and B. A. Salih, "Unlocking social media and user generated content as a data

source for knowledge management," *International Journal of Knowledge Management*, vol. 16, pp. 101–122, 2020.

[33] E. Gutierrez, "Connect to divide: Social media in 21st century Warfare," Technical report, Air Command and Staff College, Montgomery, Alabama, 2020.

[34] M. J. Mazarr, R. M. Bauer, A. Casey, S. A. Heintz, and L. J. Matthews, "The Emerging Risk of Virtual Societal Warfare: Social Manipulation in a Changing Information Environment," Technical Report, RAND Corporation, Santa Monica, CA, USA, 2019.

[35] N. Blagus and S. Žitnik, "Social media comparison and analysis: the best data source for research?" in *Proceedings of the 2018 12th International Conference on Research Challenges in Information Science (RCIS)*, pp. 1–10, IEEE, Nantes, France, May 2018.

[36] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, ICLR, Vienna, Australia, 2013.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2, NIPS'13*, pp. 3111–3119, Curran Associates Inc, Red Hook, NY, USA, October 2013.

[38] E. Jacob, *Introduction to Natural Language Processing*, MIT press, Cambridge, MA, USA, 2019.

[39] N. Imafuji and M. Kitsuregawa, "Finding a web community by maximum flow algorithm with hits score based capacity," in *Proceedings of the 18th International Conference on Database Systems for Advanced Applications DASFAA*, pp. 101–106, IEEE, Kyoto, Japan, March 2003.

[40] C. Renjifo and C. Craig, "The discounted cumulative margin penalty: rank-learning with a list-wise loss and pair-wise margins," in *Proceedings of the 2012 IEEE International Workshop on Machine Learning for Signal Processing*, pp. 1–6, IEEE, Santander, Spain, September 2012.

[41] V. F. Martinez and E. F. Martinez, "Spectral clustering for sensing urban land use using twitter activity," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 237–245, 2014.

[42] D. An, X. Zheng, C. Rong, T. Kechadi, and C. C. Chen, "Gaussian mixture model based interest prediction in social networks," in *Proceedings of the 2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 196–201, Vancouver, Canada, December 2015.

[43] A. Likas, N. Vlassis, and J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[44] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, "The determination of cluster number at k-mean using elbow method and purity evaluation on headline news," in *Proceedings of the 2018 International Seminar on Application for Technology of Information and Communication*, pp. 533–538, IEEE, Almaty, Kazakhstan, October 2018.

[45] I. Rish, "An empirical study of the naive Bayes classifier,"vol. 3, pp. 41–46, in *Proceedings of the IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, vol. 3, pp. 41–46, IBM, New York, NY, USA, 2001.

[46] N. Xu, W. Mao, and G. Chen, "A co-memory network for multimodal sentiment analysis," in *Association for Computing Machinery*SIGIR '18, New York, NY, USA, 2018.

[47] C. Hutto and E. Gilbert, "VADER: a parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the International AAAI Conference on Web and Social Media*, pp. 216–225, 2014, https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[48] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Elsevier, Amsterdam, Netherlands, 1976.

[49] J. Pastor-Galindo, F. Gómez, and G. Martínez, "Botter: A Framework to Analyze Social Bots in Twitter," 2021, https://arxiv.org/abs/2106.15543.

[50] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.

[51] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-Simulation and Computation*, vol. 3, no. 1, pp. 1–27, 1974.

[52] T. Luczak, "Size and connectivity of the k-core of a random graph," *Discrete Mathematics*, vol. 91, no. 1, pp. 61–68, 1991.

[53] J. Pastor-Galindo, F. G. Gómez, and G. Martínez, "Nothing to hide? on the security and privacy threats beyond open data," *IEEE Internet Computing*, vol. 25, no. 4, pp. 58–66, 2021.