Sparse Shield: Social Network Immunization vs. Harmful Speech

Alexandru Petrescu
University Politehnica of Bucharest
apetrescu0506@stud.acs.upb.ro

Ciprian-Octavian Truică
University Politehnica of Bucharest
ciprian.truica@upb.ro

Elena-Simona Apostol
University Politehnica of Bucharest
elena.apostol@upb.ro

Panagiotis Karras
Aarhus University
panos@cs.au.dk

ABSTRACT

With the rise of social media users and the general shift of communication from traditional media to online platforms, the spread of harmful content (e.g., hate speech, misinformation, fake news) has been exacerbated. Harmful content in the form of hate speech causes a person distress or harm, having a negative impact on the individual mental health, with even more detrimental effects on the psychology of children and teenagers. In this paper, we propose an end-to-end solution with real-time capabilities to detect harmful content in real-time and mitigate its spread over the network. Our main contribution is Sparse Shield, a novel method that out-scales existing state-of-the-art methods for network immunization. We also propose a novel architecture for harmful speech mitigation that maximizes the impact of immunization. Our solution aims to identify a set of users for which to move harmful content at the bottom of the user feed, rather than censoring users. By immunizing certain network nodes in this manner, we minimize the negative impact on the network and minimize the interference with and limitation of individual freedoms: the information is not hidden but rather not as easy to reach without an explicit search. Our analysis is based on graphs built on real-world data collected from Twitter; these graphs reflect real user behavior. We perform extensive scalability experiments to prove the superiority of our method over existing state-of-the-art network immunization techniques. We also perform extensive experiments to showcase that Sparse Shield outperforms existing techniques on the task of harmful speech mitigation on a real-world dataset.

CCS CONCEPTS

• **Theory of computation → Social networks**; • **Human-centered computing → Social network analysis**; • **Computing methodologies → Supervised learning by classification**.

KEYWORDS

network immunization; harmful speech detection; preventive immunization; counteractive immunization

ACM Reference Format:
Alexandru Petrescu, Ciprian-Octavian Truică, Elena-Simona Apostol, and Panagiotis Karras. 2021. Sparse Shield: Social Network Immunization vs. Harmful Speech. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3459637.3482481

1 INTRODUCTION

In recent years, we have noticed an increase in the online presence of the population, and the current events have also helped this growth. With the increase in the number of users, there is also an increase in harmful content that is spread on the social network with the explicit intent to target individuals or groups of people. This ill-intended content causes distress or harm, having a negative impact on individuals' mental health [8, 36], with even more detrimental effects on the psychology of kids and teenagers [4]. To mitigate harmful content, there is a need to increase the interaction of moderators with social media posts. Sometimes, this approach is not plausible, as automated content can be created by bots [27]. Thus, we propose a new architecture to mitigate the spread of harmful content, offering a novel solution for the task of network immunization.

In this paper, we tackle the issues of harmful speech detection and network immunization using real-world Twitter datasets. We have chosen to collect a large dataset from the social network Twitter because *i)* it has a large number of active users; *ii)* its structure can be formalized as a directed graph by generating edges between two users based on either user interaction, i.e., follower-followee, or content interaction, i.e., liking, commenting, or retweeting a post; *iii)* weights can be added using different heuristics to analyze the spread of information; and *iv)* the post can have a maximum length of 280 characters forcing the users to be as concise as possible.

For the harmful speech detection task, we use an existing labeled dataset to build and benchmark multiple models. We use a binary classification approach that takes into consideration the number of harmful symbols, which can be split into multiple bins such as race, religion, etc.

For the immunization task, we use the Twitter API to collect tweets and build a graph structure. We define the network immunization problem as follows. For a given directional weighted graph and a given *immunization budget* k of graph nodes that will be *immunized*, i.e., have the rank of the harmful content lowered so that it will not be reached without specific intention, select k nodes to immunize in order to minimize the spread of harmful information. We distinguish two problem variants: in *preventive* immunization,

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482481

we do not know the sources of harmful content, and we immunize the network in advance; in *counteractive* immunization, we know those sources yet cannot select them for immunization.

To handle these problems, we propose Sparse Shield, a scalable algorithm for network immunization that excels at preventive immunization and also provides a foundation for counteractive immunization. Immunization is performed only when harmful content is detected and only by lowering the rank of that particular post in the feed. Each post has a rank on the feed of a user; by lowering it, we minimize the chance that harmful content will appear on the top of the feed, but it can still be found if the given user consume a lot of content. We apply immunization only when harmful content is detected; if it is overused, it may *i)* infringe on the freedom of speech, and *ii)* alter the structure of the influence graph.

In our experiments, we use the following simplified presumptions: *i)* harmful content is something rare in our network; *ii)* information is spreading in the same way regardless of its nature; *iii)* active users, the ones who interact with the posts, spread the information; *iv)* we can ignore the inactive users.

To show our method’s efficiency, we compare it with four state-of-the-art methods: *i)* Random which selects node at random to be blocked; *ii)* Degree heuristic [12] which selects the highest degree nodes to be immunized; *iii)* DAVA [44] which uses a dominator tree-based algorithm for network immunization; and *iv)* NetShield [11] which computes a *vulnerability* value equal to the dominant eigenvalue of the network and then builds a priority queue to immunize based on the budget.

In summary, our contributions are:

- We perform an in-depth analysis of current state-of-the-art Transformer and Deep Learning-based models for the task of detecting harmful speech.
- We propose a new architecture for network immunization of harmful speech and apply it on a real-world Twitter dataset.
- We propose Sparse Shield, a novel *preventive* network immunization algorithm that outperforms the state-of-the-art methods, i.e., NetShield, DAVA, etc.
- We propose two *counteractive* solutions based on Sparse Shield, i.e., Sparse Shield Plus, Sparse Shield Seedless, that perform well on both small and high budget immunization.
- We do extensive experiments on the mitigation of harmful speech in social networks on the worst outcome possible scenarios using a real-world Twitter dataset.

This paper is structured as follows. In Section 2, we analyze the current literature and present the state-of-the-art approaches used for harmful speech detection and network immunization. In Section 3, we present our architecture for harmful speech mitigation in social networks and we describe and analyze our novel algorithm for preventive network immunization called Sparse Shield. In Section 4, we present *i)* the real-world dataset collected from Twitter, *ii)* the experimental results of our harmful speech detection benchmark that shows which model performs better on our dataset, and *iii)* Sparse Shield performance for the task of harmful speech mitigation in social networks. Also, we empirically prove the superiority in the scalability of our network immunization method over the state-of-the-art algorithms and strategies. Finally, in Section 5, we draw our conclusions regarding our proposed pipeline and new algorithm and hint at future directions for this solution.

2 RELATED WORK

Harmful speech detection is tackled by both Machine Learning and Natural Language communities as a text classification task that requires inside knowledge about the use of language. On the other hand, network immunization is tackled by the Network Analysis and Graph Mining communities and deals with proposing strategies for stopping the spread of information within a network structure. To the best of our knowledge, these two communities have never come together and work on a task such as harmful speech mitigation in social networks that can be seen as a context-aware network immunization problem. Thus, this section is divided into two parts: *i)* harmful speech detection and *ii)* network immunization.

2.1 Harmful Speech Detection

In the current literature, harmful speech datasets are relatively small and hard to align in a multi-class classification style [15]. More general categories such as "toxicity" or "offensive" are identified well, but the more subtle ones such as "threat" or "severe toxic" are harder to find. For a more robust detection and a better understanding of the nature of the harmful content, some solutions use a multi-class representation [7, 38]. Banko et al. [7] use 13 classes and 4 super-classes of harmful content which helps them adjust how fine-grained the detection of harmful content can be. Sharma et al. [38] propose a fine-grained approach that employs 3 super-classes based on the degree of harmfulness and 12 classes. The proposed dataset provides good accuracy even for simple classifiers such as Naïve Bayes or Random Forest, i.e., accuracy >70%.

Waseem et al. [40] propose a dataset containing classes based on $\{Directed, Generalized\} \times \{Explicit, Implicit\}$ to focus on the typology rather than on the content. Using a logistic regression classifier, they conclude that the best results are obtained using character n-grams of lengths up to 4, along with gender as an additional feature.

Sap et al. [37] focus on the source of the offensive content and their target. When the text comes with the racial profile of the person that posted it, the label is adjusted as it is proved that the users of different ethical groups consider different things offensive. Using neural attention architecture initialized with GloVe vectors, they manage to determine the false positive rates differ across groups for several toxicity labels.

In the current literature, the majority of harmful speech detection solutions apply for classification Transformers or Deep Learning architectures. Alonso et al. [3] show that RoBERTa (Robustly Optimized BERT Pretraining Approach) [22] model provides very good results for detecting harmful speech in Twitter-based content with minimal text processing. The authors obtain high F1 scores, and when analyzing the wrongly classified tweets, they observe that more subtle things were misclassified. Zampieri et al. [43] also obtain good results with multiple deep approaches such as BiLSTMs for the tasks at SemEval-2019. On the subtle side of harmful content, Lemmens et al. [20] use the Dutch versions of BERT and RoBERTa with good results on hateful metaphors on Dutch LiLaH corpus consists of approximately 36K Facebook comments.

In the case of multiple languages or cross-language harmful speech detection, we can also see Zero-shot approaches [29] providing good results, sometimes better even than Transformers such as BERT (Bidirectional Encoder Representations from Transformers) [14]. Markov et al. [25] also find BERT to be able to pick up better explicit harmful content when dealing with English or other single-language datasets.

Qian et al. [33] talk about the hard truth that the models might not remain up-to-date as the hot topics of social-media change. To alleviate catastrophic forgetting, they propose to use Variational Representation Learning (VRL) [19] along with a memory module based on Load-Balancing Self-Organizing Incremental Neural Network (LB-SOINN) [28].

One of the issues of deep models might be the high number of falsepositives. Markov and Daelemans [24] achieve better results for cross-domain hate speech detection using non-deep models such as Ensembles. In our experiments, we also obtain good results with balanced benchmarks using Ensembles.

In conclusion, it is important to do a qualitative analysis not only a quantitative one, as Mosca et al. [26] also state. They also analyze what were the most harmful topics, how they were distributed, and how they fit in the bigger picture of the social network.

2.2 Network Immunization

For the smaller graph structures, Li et al. [21] observe that the Degree heuristic does not provide results as good as other methods that also consider the network structure. They conclude that a good method to upgrade the Degree heuristic is by also using eigenvalues. Yoshida and Yamada [42] also arrive at the same conclusion and propose the use of network properties that are leveraging the structures of communities. Ghalmane et al. [16] propose a solution for network immunization that integrates the non-overlapping community structure. The authors propose structures that have properties of real-world networks, although there are under 15K nodes. They obtain good results, and we can see that the bigger the number of non-overlapping communities, the lower the performance will be.

Ahmad et al. [1] propose a method that leverages the combinatorial properties of nodes that outperforms NetShield [11] while improving resource cost. Peng et al. [30] present an approach that models real-world graphs that are build using a smartphone social network. They propose an immunization method based on a BFS (Breadth-First Search) algorithm that ranks each node. In this work, the authors have also considered the degree and the number of messages sent over a fixed period to contribute to the rank of a node. Logins and Karras [23] analyze multiple types of preventive and counteractive network immunization, i.e, NetShield [11], DAVA [44], etc. They concluded that the behavior of the algorithms is correlated with the network's structure and properties.

Zhang and Prakash [44] propose a good and almost salable method for counteractive network immunization in large networks. The method leverages the tree-like stature of subgraphs by building dominator trees out of the infected nodes. They conclude that data-aware approaches based on the construct of dominator trees perform best in most cases. However, preemptive approaches utilizing spectral network properties are better for networks with a power-law degree distribution [2].

Ren et al. [34] introduce the concepts of the excepted eigenvalue (EE) and excepted fraction of infected node (EF) to quantify the spread strength and influence of disease or viruses. The aim is to minimize the EE and EF values of the remaining network with an algorithm based on the characteristics of degree and largest eigenvalue in uncertain networks and obtain good results against other spectral-based methods in really small graphs.

3 PROPOSED SOLUTION

Figure 1 presents the architecture of our solution. By listening to the Twitter-Stream, we can detect, in real-time, harmful content and stop its spread or minimize the impact, based on the case. By immunizing a node, we propose lowering the rank of the specific post in the feed of the selected nodes or if the source itself is selected to the 1st-degree connections.

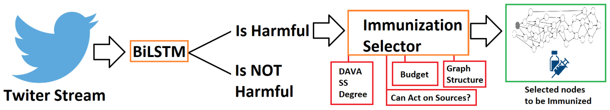


Figure 1: Proposed Solution for Network Immunization

The proposed system has the purpose to lower the chance of a user reaching harmful content as soon as the feed is opened. By lowering the rank of that content, we increase the chance that users who just want to casually browse Twitter for a few moments are not impacted by ill-indented content while not removing it. Lowering the rank of a post means that it will require more time to reach other users and followers if they are not expressly looking for it by browsing the feed.

When dealing with the immunization of nodes within a social network, we must be careful not to overuse the system and block the same users too often. To achieve this, we need to take into consideration multiple factors before selecting an algorithm. Thus, if the proposed network immunization strategy is used too much, it can have the following negative impacts, which we want to avoid:

- *i)* Social impact: the strategy can become a censorship method that can infringe on the liberties and rights of the individual.
- *ii)* Computational impact: the influence network will change as the strategy blocks nodes and, for each new iteration, it needs to be recalculated.

3.1 Harmful Speech Detection

For this task, we are leveraging state-of-the-art methods such as Transformers or Bi-LSTM in conjunction with some of the best pre-trained word embeddings. By having already trained models, we are fine-tuning the harmful speech detection algorithms to our Twitter dataset with no prior preprocessing required. To encode the textual content we use two word embeddings, i.e., Glove and FastText, and three context-aware embeddings, i.e, BERT, RoBERTa, and XML-RoBERTa. For classification, we implement the following models: BiLSTM, Classification using Transformers, Voting Ensembles, and Stack Ensemble. After analyzing several datasets and also considering the findings from other related work solutions, we concluded that for this task we are using binary classification. The reason is that, although general harmful speech categories, e.g., "toxicity", "offensive", are identified with high accuracy, the more subtle ones, e.g., "threat", "severe toxic", are harder to find.

Text Representation. *Glove* (Global Vectors for Word Representation) [31] is a global log-bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization and local context window methods. This embedding is suitable for large vocabularies, such as the ones used in social-media-based datasets, and comes with multiple pre-trained representations in different vector spaces. Also, a dimensional reduction using t-SNE [39] can dramatically improve performance, lower the system requirements, and not affect the performance [10].

FastText [9] is a word embedding similar to Word2Vec that takes the representation to the next label by moving to a char n-gram based model instead of a word-based model with existing pre-trained models. Improvements are addressing the Continuous Bag of Words, CBOW, and the log probability.

BERT [5] proves the importance of bidirectional pre-training by surpassing the previous state-of-the-art approach, which was a left-to-right approach. Instead, BERT uses the self-attention mechanism to unify these two stages, encoding a concatenated text pair with self-attention, and including bidirectional cross attention between two sentences. The pre-trained model comes with a generally-trained embedding layer on a very large Corpus and for general use. The model will go only under the fine-tuning step to be adapted to the task.

RoBERTa [22] is a retraining of BERT with improved training methodology, 1000% more data, and compute power. To improve the training procedure, RoBERTa removes the Next Sentence Prediction (NSP) task from BERT's pre-training and introduces dynamic masking so that the masked token changes during the training epochs. Larger batch-training sizes were also found to be more useful in the training procedure.

XML-RoBERTa [13] is a Transformer-based pre-trained multilingual masked language model able to determine the correct language just from the input ids. It does not require language tensors to understand which language is used.

Classification Methods. *LSTM* (Long Short-Term Memory) [35] is a Neural Network architecture that has good results on text classification. By using LSTM in a bidirectional architecture, it offers better results as the max-pooling mechanism constrains the model to capture the most useful features produced by the *BiLSTM* encoder [32].

Classification using Transformers is done by adding a dense layer for determining the correct class. The input of the dense layer is the hidden layer of the Transformer.

Voting and Stack Ensemble use SVM, Trees, and Regression for classification. The difference between voting and stacking is how the final aggregation is done. In voting, user-specified weights are used to combine the classifiers, whereas stacking performs this aggregation by using a blender/meta classifier.

3.2 Network Immunization Methodology

For the network immunization module, we propose a new algorithm: Sparse Shield. For comparison, we use as baselines the following heuristics that will be applied on all nodes, excluding the infected ones: *i)* Selecting Random Nodes to be Immunized, and *ii)* Selecting the Highest Degree Nodes Possible to be Immunized [12] We compare Sparse Shield with two state-of-the-art network immunization strategies to prove its superiority w.r.t. scalability: *i)* Data-Aware Vaccine Allocation over Large Networks (DAVA) [44], and *ii)* NetShield [11].

To test the effectiveness of these algorithms, we are running simulations under the Independent Cascade Model [18], i.e., the information flows over the network through the cascade. Using this model, nodes are either active (i.e., the node is already influenced by the information in diffusion) or inactive (the node is unaware of the information or not influenced).

The main objective of this module is to *save a node*, which means that the node will no longer be reached by the harmful content through the immunization of the selected nodes. In the case of Sparse Shield, if we immunize an infected node, that node will be counted towards the infected nodes.

DAVA [44] is a network immunization algorithm that requires that the sources of the infection are known apriori. The algorithm creates dominator trees from the initial node structure using the spreading probability as the weight of the edges. Based on the dominator tree, the nodes that are not on the first layer will be removed, and this process is repeated based on the budget.

In the original paper, the authors propose two more heuristics *i)* DAVA-prune that speeds up the process by only re-weighting the nodes that are involved in the first layer of the dominator tree, and *ii)* DAVA-fast, which will run the dominator with the whole budget for immunization instead of 1, thus skipping the loop. For our experiments, we use DAVA-prune as it offers the same results as DAVA, but in less time.

NetShield [11] is a network immunization algorithm that makes the following assumptions: *i)* a node 'vulnerability' value can be obtained by computing the dominant eigenvalue of the network from the adjacency matrix, and *ii)* using the node vulnerability value, a priority queue to immunize based on a budget can be built.

The method does not require knowing the source of the infection and can be pre-applied to the network as it will rank the nodes to be immunized based only on the structure of the graph. This makes the method scale very well as the main time-consuming task is obtaining the eigenvalue. The rest is just a selection of nodes that is comparable with the Degree algorithm. Furthermore, we can also specify to NetShield the known sources of the infections. Thus, the algorithm can run only when it is needed on a given node that spread harmful content.

3.3 Influence Graph

As we are using the Independent Cascade model, we must define the probability of v to spread the information from u to obtain the weight of an edge. Thus, we are modeling the graph under the assumption that the spread of information is based on a network built from twitter-stream by combining two probabilities (Equation 1). We call this graph the Influence Graph. The two probabilities required to build this graph are:

1) The passive probability (Equation 2) which is always 0.5 since this is the base for all edges. We are building an edge (u, v) only if v interacted with any of the content from u over the span of 2 weeks.

It is only "a coin toss" because we want to model the probability to spread any type of content, harmful or not.

2) The active probability (Equation 3) which represents how influential a node is, equally over its edges. This probability tries to model the "influencer" property of a social network node, by raising the probability of spreading information from more popular nodes.

$$p(u,v) = p_{passive}(u,v) + p_{active}(u,v) \tag{1}$$

$$p_{passive}(u,v) = \begin{cases} 0.5, & \text{if } (u,v) \in E \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$p_{active}(u,v) = 0.5 \cdot \frac{|\{v|(u,v) \in E\}|}{|V|} \tag{3}$$

3.4 Sparse Shield

NetShield [11] solves the problem of network immunization by computing the largest eigenvalue and then applying some priority queue heuristics using this value. This method is valid and produces good results, but it is not optimal in terms of scalability when dealing with large graphs with very few edges, as it happens when modeling social media. To compute the eigenvalues, we must first build the adjacency matrix A from the social graph. This matrix is generated from the nodes of the undirected and unweighted social graph $G = (V, E)$. Due to the social media graph structure, we obtain a very large sparse matrix $A \in \mathbb{R}^{|V| \times |V|}$ needed to compute the eigenvalues. Thus, the graph G meets the following two properties:

- Sparse [6] - due to the fact that $\|V\| \simeq \|E\|$ and $\|V\|$ is large;
- Symmetrical - due to the undirected adjacency matrix.

Sparse Property. Let $u, v \in V$ be 2 arbitrary nodes with $u \neq v$. We define the adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$ for the undirected representation of G as $A_{u,v} = 1 \wedge A_{v,u} = 1$ if $(u,v) \in E$ and $A_{u,v} = 0 \wedge A_{v,u} = 0$ otherwise.

Relationships between nodes in a social network can be defined in one of the following ways:

- Directional: a user can access the content of another user, but not vice-versa.
- Unidirectional: both users involved can see each other's content.

In this paper, we are considering a graph-based approach for tweets where there is only the directional relationship given by the "follow" action. By running the network analysis over Twitter, either using the APIs or listening to the Twitter Message Stream, we can observe that some users have highly disproportionate ratios between the number of followers and the number of people they follow. This is natural due to the amount of content one user can consume. Moreover, in the resulting graph, some nodes have the property of "social influencers", which means that their content can reach many other nodes. While each social platform has its own minimum number of influencers, the influencer/non-influencer ratios are similar between them. Thus, for each generated social network graph, the number of nodes with a high degree will be much lower than the rest of the nodes.

3.4.1 *Sparse Shield Algorithm.*

The pseudo-code for Sparse Shield is presented in Algorithm 1. The algorithm receives as input the influence graph $G = (V, E)$, the immunization budget k, and the hyper-parameter $\alpha > 0$ which acts as the priority multiplier for immunization. The hyper-parameter has only positive values, otherwise, we revert the priority queue Q. The higher the value of the parameter, the more importance is given to the neighbors of the selected node, and the probability of selecting a neighbor in the next iterations is increased.

The first part of the algorithm builds the adjacency matrix A from the undirected and unweighted social graph G (Lines 1 to 4). Using the adjacency matrix A, the algorithm computes the maximum eigenvalue and its corresponding eigenvector using the Arnoldi algorithm (Line 5). Based on the maximum eigenvalue and eigenvector, we compute each node's priority score and create the priority queue Q (Lines 6 to 9). For a given budget k, the algorithm *i)* determines the lowest priority node u for immunization and update Q and the list of immunized nodes *immunized* (Lines 12 to 19), *ii)* computes the neighbors of u (Lines 20 to 23), *iii)* updates Q with the neighbors v of u by updating for them a priority score p_v to p'_v (Lines 24 to 28), and *iv)* updates the budget k (Line 29). When the entire budget is spent, the algorithm returns the list of immunized nodes *immunized* (Line 30).

To compute the algorithm's time complexity, we define k as the budget and i as the number of infected nodes. The complexity is $O(k \cdot |V|^2)$, and it is computed as follows:

- The complexity of building the adjacency matrix in sparse form is $O(|E|)$;
- The complexity of using the Arnoldi algorithm and extracting the dominant value is $O(|V|)$;
- The complexity of building the priority queue is $O(|V|)$;
- The construction of the immunized list of nodes is $O(k \cdot |V|^2)$.

We also propose two variants of Sparse Shield for counteractive immunization:

- Sparse Shield Plus: removes the infected nodes from G adding a complexity $O(|E| + |V|)$, but keeping the total complexity $O(k \cdot |V|^2)$ since we are dealing with a sparse matrix;
- Sparse Shield Seedless: increases the number of nodes with the number of infected nodes i to be able to remove them from the selected nodes resulting in a complexity of $O((k + i) \cdot |V|^2)$.

With the above configuration, the best way to compute the maximum eigenvalue is the Arnoldi algorithm, which finds an approximation to the eigenvalues and eigenvectors of general (possibly non-Hermitian) matrices by constructing an orthonormal basis of the Krylov subspace, which makes it particularly useful when dealing with large sparse matrices. Any standard method to compute or represent the adjacency matrix will result in expanding a sparse matrix unnecessary, which in turn will require more resources in terms of RAM and CPU.

4 EXPERIMENTS

We use Python 3 for implementing our solution. The code is publicly available online on GitHub at https://github.com/AskingAlexander/SparseShield_NIvsHS We use the following hardware configurations for the experiments:

Algorithm 1: Sparse Shield Algorithm

```
Input  : the influence graph $G = (V, E)$,
         the immunization budget $k$, and
         the priority multiplier $\alpha$
Output : the list of immunized nodes $immunized$
/* Create the sparse adjacency matrix for the
   undirected and unweighted graph G                    */
$A \leftarrow O_{|V| \times |V|}$;
foreach $(u, v) \in E$ do
    $A_{u,v} \leftarrow 1$;
    $A_{v,u} \leftarrow 1$;
/* Compute the maximum eigenvalue $\lambda$ and its
   corresponding eigenvector $\mathbf{w}$ using Arnoldi */
$\lambda, \mathbf{w} \leftarrow Arnoldi(A, 1)$;
/* Compute each node's priority score $p_u$ and
   create the priority queue $Q$                        */
$Q \leftarrow \emptyset$ ;
foreach $u \in V$ do
    $p_u \leftarrow \alpha \cdot \lambda \cdot \mathbf{w}_u^2$;
    $Q \leftarrow Q \cup \{(u, p_u)\}$;
/* Build the list of immunized node given a
   budget $k$                                           */
$immunized \leftarrow \emptyset$;
while $k > 0$ do
    /* Get the lowest priority node for
       immunization and update $Q$ and $immunized$      */
    $u \leftarrow None$;
    $p_u \leftarrow \infty$;
    foreach $(v, p_v) \in Q$ do
        if $p_v \leq p_u$ then
            $u \leftarrow v$;
            $p_u \leftarrow p_v$;
    $Q \leftarrow Q \setminus \{(u, p_u)\}$;
    $immunized \leftarrow immunized \cup \{(u, p_u)\}$;
    /* Get the neighbors of $u$                         */
    $neighbors \leftarrow \emptyset$;
    foreach $edge\ e \in E$ do
        if $u \in e$ then
            $neighbors \leftarrow neighbors \cup \{v | v \in e\}$;
    /* Update $Q$ with the neighbors of $u$             */
    foreach $node\ v \in neighbors$ do
        if $v \notin immunized$ then
            $Q \leftarrow Q \setminus \{(v, p_v)\}$;
            $p'_v \leftarrow p_v - \alpha \cdot \mathbf{w}_u \cdot \mathbf{w}_v$;
            $Q \leftarrow Q \cup \{(v, p'_v)\}$;
    /* Update the budget                                */
    $k \leftarrow k - 1$;
return $immunized$;
```

- To build and benchmark the models for the harmful speech detection task, we use Google Colab Pro with the High Memory GPU machine option that can leverage 16GB of VRAM.
- To perform network immunization, we use a Windows 10 machine with 64GB of DDR4-3 200 MHz RAM, and an 8-core 16-thread 4.3GHz AMD Razor Processor.

4.1 Influence Graph

We build our Influence Graph using a real-world Twitter dataset. We collect two weeks of data from the Twitter Stream to reflect a true network of information spreading, considering only English tweets that use any kind of mention/retweet. After collecting the raw data, we aggregate it in the following way:

- Let x be a tweet from the collected dataset. From x we are only interested in: *i)* $y = x['user']['id']$ the user unique identifier, *ii)* $z = x['entities']['user\_mentions']$ the other Twitter users mentioned in the text of the tweet, and *iii)* $c = x['text']$ the content of the tweet.
- For each y, we build a list of all distinct z entries and define $Z(y)$ as the list that contains all the nodes obtained by $y \cup z$.
- The final directed graph is built the following way $G = (V, E)$ where $V = \{v | v \in y || v \in z, distinct(v)\}$ and $E = \{(v, u) | u, v \in V, v \in Z(u)\}$.
- We associate for each user y the content of the tweet c.

Thus, we model the social network as a graph as follows: Let $G = (V, E)$ with V a set of vertices and E a set of directed edges in which $\forall e \in E$, e connects u and v which means that u can influence v and this is computed by the gathered data if v interacted (liked, commented or retweeted), at least once, with a post from u.

The idea can be applied to any social media platform in which we can model the network as a directed graph, and we can create the Influence Graph as stated above.

Twitter has about 187 million (M) active users worldwide. The top 4 most-followed people on Twitter have about 100 M - 130 M followers, which means that also selecting sub-graphs of Twitter will result in huge degree variations between users.

For our experiments, we start from the base graph configuration ($G4$) and then build smaller graphs by removing the nodes with the smallest degree until reaching a satisfying number of nodes. This is because we want to cover the cases when the information will spread from the most influential nodes as we try to minimize the impact of harmful content over the network. All the graph configurations can be found in Table 1.

Removing nodes in the order $G4 \gg G3 \gg G2 \gg G1$ changes the root (i.e., the node with the highest degree) and the structure of the graph. Moreover, by removing a high degree node, we also remove its leaves (i.e., the nodes with the smallest degree), thus the distribution of degrees remains almost the same.

4.2 Harmful Speech Detection

For this task, we use the "Hateful Symbols or Hateful People" dataset [41] which has 25K tweets creating a vocabulary of 6330 words after pre-processing. The records are classified in 2 ways:

- The degree of hateful symbols used, from 0 to 6;
- The degree of hateful speech used, from 0 to 6.

Table 1: Graph Configurations

Code	Nodes	Edges	Root Node
G1	3.8K	85.6K	1 409 798 257
G2	47K	0.63M	2 228 960 582
G3	0.88M	2.8M	2 228 960 582
G4	2.79M	4.55M	1 409 798 257

For our approach, we are combing the various degrees into a single one by the following formula $IsHarmful = (HateSymbol + HateSpeech > 0)$. Using this approach, the final dataset remains balanced. We also note that usually, hateful symbols are used together with hateful speech, but the reverse is not applicable. We use the 80%-20% ratio to split the dataset into the training and the testing sets.

For the BiLSTM setup, we build the models using 2 pre-trained word embeddings, i.e., Glove300D and Fasttext300D (300 vector-space). Using hyper-parameter tuning, we obtain the best parameters as follows. For BiLSTM, we set $BATCH\_SIZE = 64$, $N\_EPOCHS = 50$, $num\_hidden\_nodes = 32$, $num\_layers = 2$, and $dropout = 0.24$. While for the Transformers, the best hyper-parameters for fine-tuning are: $BATCH\_SIZE = 64$, $N\_EPOCHS = 10$, $weight\_decay = 0$, $early\_stopping\_delta = 0.01$, and $learning\_rate = 2e - 5$. Fine-tuning the Transformers takes more memory and time than training the BiLSTM model.

The results can be seen in Table 2. We observe that Transformers models obtain an accuracy of over 84%. The BiLSTM models obtain the overall best results, with an accuracy of 93.4% when using FastText and 93.28% when using GloVe. This small difference between the accuracy of word embeddings is a direct impact of word representations learned from the training dataset. The overall best model is BiLSTM with FastText. The out of the box ensemble model performs very well and is recommended when the training is done using cloud-based platforms.

Table 2: Twitter Classification Results

Model	Accuracy	F1	Precision	Recall
BiLSTM FastText	**93.40**	**92.86**	**92.84**	**92.97**
BiLSTM GloVe	93.28	92.85	92.76	92.93
BERT	84.40	86.00	87.40	87.90
RoBERTa	85.70	86.70	86.70	87.70
XML-RoBERTa	84.30	85.90	87.40	88.00
Voting Ensemble	92.97	84.03	83.01	87.06
Stack Ensemble	92.93	83.01	84.04	85.06

4.3 Scalability

One of the main advantages of Sparse Shield is the improved cost in terms of time, memory, and CPU usage. This allows our solution to outperform algorithms on the same datasets.

Figure 2a presents a runtime comparison between NetShield and Sparse Shield. We observe that our solution outperforms considerably NetShield w.r.t. the size of the graph. Due to the high memory requirements for running NetShield, we are not able to run this algorithm for G3 and G4, but we can estimate its performance using multiple runs on the smaller graphs. Figure 2b presents the improvement of Sparse Shield over NetShield in terms of memory consumption. For G3 and G4, we estimated the NetShield required memory using the number of allocation errors obtain during objects resource allocation requirements. In terms of CPU usage, we also see a significant boost (Figure 2c) which means that we can also leverage multi-threading in future work.

Figure 3 presents the runtime performance for the immunization of a graph. For G1, we have a few seconds difference between the tested algorithms. The gap in performance time increases with the size of the graph. Thus, for the entire Influence Graph, i.e., G4, the difference in runtime increases to hours. We observe that Sparse Shield performs better than DAVA, but it is outperformed by the simpler heuristics Random and Degree.

Figure 4 presents the performance between Sparse Shield and DAVA on different graph sizes w.r.t. various budgets. Both methods scale linearly. DAVA scalability is influenced by both the size of the graphs and the budget. Since Sparse Shield has only to rank the nodes based on the eigenvalue, it remains invariant to the number of nodes to immunize. Thus, it scales only with the size of the graph. Taking into consideration a large number of nodes and the size of the budget, it becomes unjustified to consider running DAVA to immunize a social network graph structure.

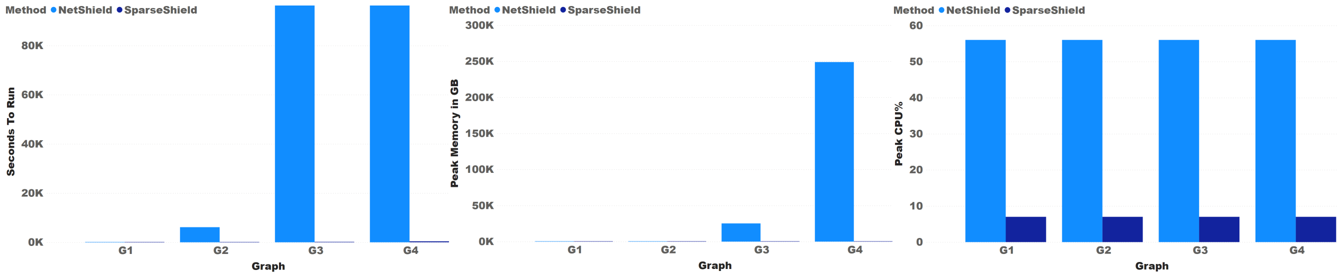
4.4 Network Immunization of Harmful Speech

We simulate the propagation of harmful speech using the Independent Cascade model and try to present the worst-case scenario with the next experiment. For this, we are going to randomly sample nodes from the ones with the highest degree in the network to be our harmful speech information sources. We use a combination of both preventive and counteractive immunization to determine the best choices for this case. For the test cases when the infection source can not be immunized, we test the two variants of Sparse Shield, i.e., Sparse Shield Plus and Sparse Shield Seedless.

4.4.1 Saved nodes vs. budget. For the first set of experiments, we are using the following graph configuration: *i)* graph is G1; *ii)* out of the top-15 nodes with the highest degree 10 nodes are selected at random to be infected; *iii)* for each set of infected nodes, we are performing 1 500 random selections. This simulation is performed 250 times using the Independent Cascade model.

In Figure 5, we can see that DAVA, counteractive immunization, behaves almost as well as Sparse Shield, which employs preventive immunization. By considering the Sparse Shield variants that do counteractive immunization, we see a significant decrease in performance than DAVA, making them not so reliable. We can also convert Sparse Shield to a counteractive method by immunizing the nodes that are directly connected to the infected nodes selected for immunization. But, in this case, the budget increases dramatically as it is directly correlated to the structure of the Influence Graph.

For the second set of experiments, we compare a small budget ($k \in \overline{1, 10}$) with a high budget ($k \in \overline{110, 910}$) when selecting at random 100 nodes from the set of infected nodes. This simulation is performed 250 times using the Independent Cascade model. Figure 6 presents the results for this comparison. For a high budget, Figure 7 depicts the ratio between saved and infected. Analyzing

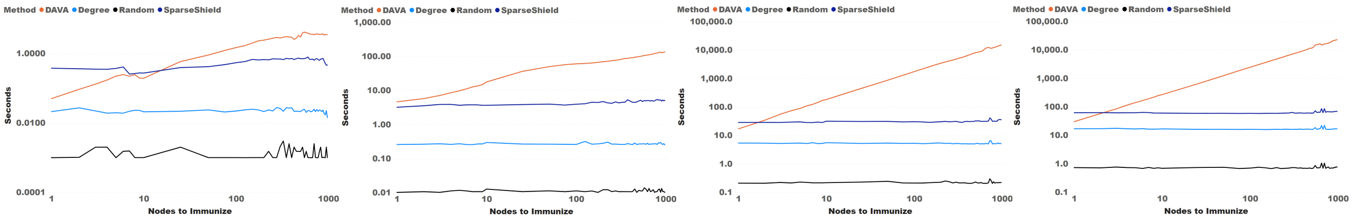


(a) Runtime

(b) Memory

(c) CPU%

Figure 2: Performance evaluation



(a) G1

(b) G2

(c) G3

(d) G4

Figure 3: Runtime for computing the nodes to be Immunized

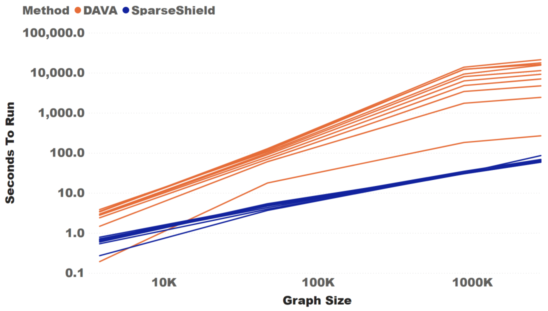


Figure 4: Runtime comparison for various budgets

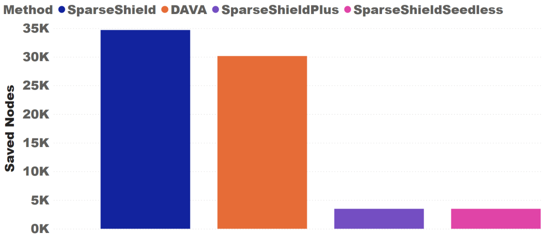
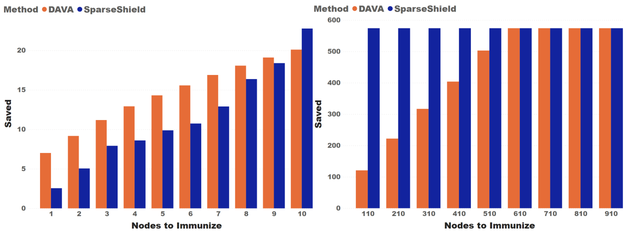


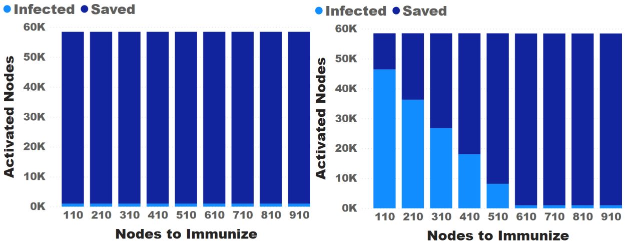
Figure 5: Total nodes saved



(a) Small budget

(b) High budget

Figure 6: Saved nodes



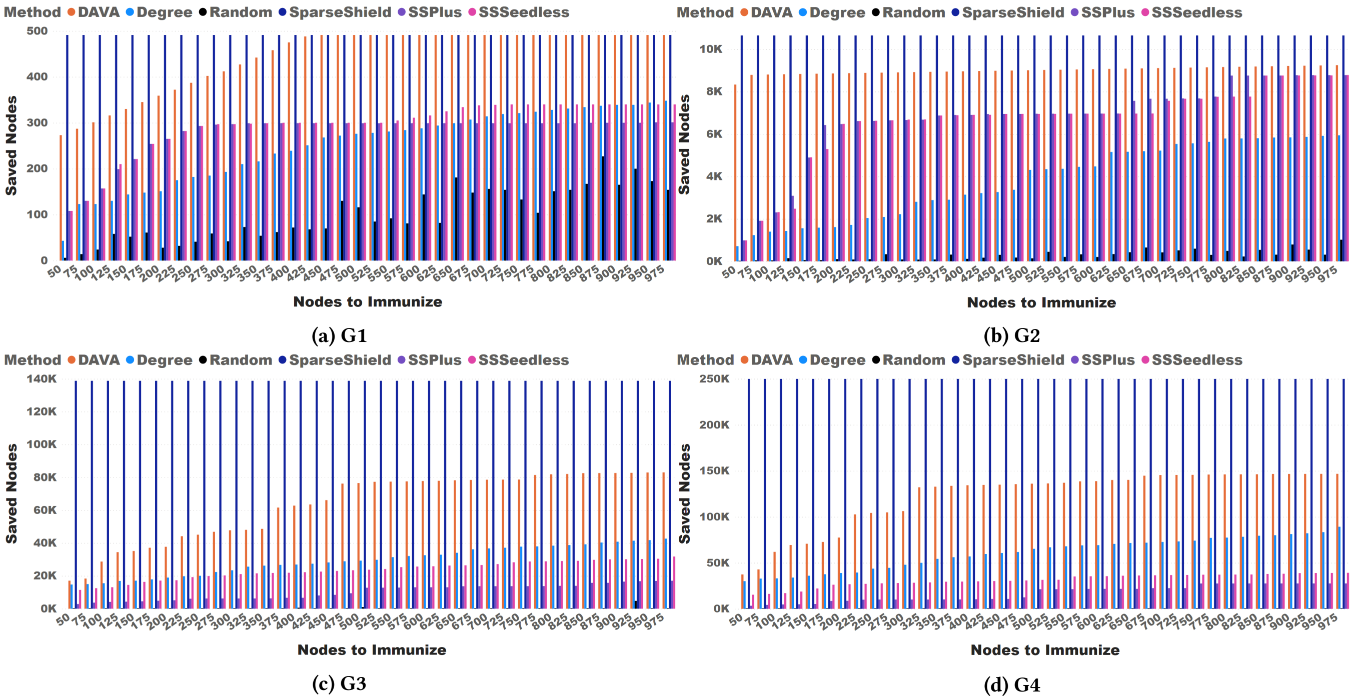
(a) Sparse Shield

(b) DAVA

Figure 7: Infected vs. saved nodes for a high budget

the results, we conclude that DAVA performs better with a smaller immunization budget, while Sparse Shield is optimal for bigger budgets.

4.4.2 Sparse Shield variants vs. others. For this set of experiments, we want to determine how preventive immunization compares to counteractive immunization. This set of experiments is performed on all the graphs. We use the node with the highest degree as the source of spreading harmful content. We perform 1 000 simulations using the Independent Cascade model. We observe that Sparse



(a) G1

(b) G2

(c) G3

(d) G4

Figure 8: Saved nodes benchmark

Shield, a preventive method, is outperforming all other tested algorithms (Figure 8).

Sparse Shield obtains the overall best performance using preventive immunization regardless of the employed graph. DAVA has the overall best performance among the counteractive algorithms. For G1 and G2, where we have different sources of infection, the performance from best to worst is as follows: Sparse Sheild, DAVA, Sparse Shield Plus (SSPlus), Sparse Shield Seedless (SSSleedless), and Degree. For G3 and G4, where we have the same source of infection, we observe the following behavior in performance, from best to worst: Sparse Sheild, DAVA, Degree, Sparse Shield Seedless, and Sparse Shield Plus. We conclude that Sparse Shield Plus works better than Sparse Shield Seedless on smaller graphs, while the reverse happens on larger graphs.

5 CONCLUSION

We propose a new end-to-end real-time-capable architecture for harmful speech mitigation in social networks, leveraging state-of-the-art methods and introduce a new network immunization method, Sparse Shield, and apply it to real-world data.

Our solution offers two types of network immunization against harmful content, i.e., preventive and counteractive immunization. When we can block the sources or high degree nodes, we have preventive immunization. In this case, we can employ Sparse Shield to obtain a ranking of the nodes and select the top-*k* nodes for immunization. As Sparse Shield is independent of the sources of the infection, it can run in the lower activity hours of the network. Thus, ensuring that the performance of the existing machines which host the environment is not affected. When we know the sources but cannot act on them, we can use counteractive immunization. Aside from strong scalability and strong preventive immunization results, Sparse Shield also offers a foundation for effective counteractive immunization. Yet, for a small budget, DAVA provides the best counteractive immunization outcome.

We have seen in the experiments that both NetShield and DAVA consume a lot of resources when *i)* the network is large, and *ii)* it is required to immunize a large number of nodes. For these cases, the cost-efficient solution is Sparse Shield which efficiently acts on the sources of the harmful speech. Although Degree offers reduced performance, it can immunize the network very fast. It is important to consider the time in which a network immunization algorithm provides a solution in the real world; if the reaction is not within minutes, the information may have already been spread far beyond the scenarios in which the methods are accurate.

To detect the harmful content, we employ a BiLSTM classifier that encodes tweets using word embeddings. Using directly the model is memory-efficient and can offer real-time detection. Furthermore, it can be retrained periodically offline to improve the accuracy of detection. Using the content-based analysis of the tweets, we can pick a method to immunized the nodes and lower the rank of the harmful post for the selected nodes rather than on the whole network; thereby, we minimize the impact while minimizing the infringement on users’ freedoms.

In the future, we plan to design new immunization approaches that integrate content and network structure in their analysis, in the spirit of influence spread models that consider the content of propagated messages [17]. We also aim to include the nodes’ rank to fine-tuning the models and improve classification performance.

REFERENCES

[1] Muhammad Ahmad, Sarwan Ali, Juvaria Tariq, Imdadullah Khan, Mudassir Shabbir, and Arif Zaman. 2020. Combinatorial trace method for network immunization. *Information Sciences* 519 (2020), 215–228. https://doi.org/10.1016/j.ins.2020.01.037

[2] Muhammad Ahmad, Juvaria Tariq, Mudassir Shabbir, and Imdadullah Khan. 2017. Spectral Methods for Immunization of Large Networks. *Australasian Journal of Information Systems* 21 (2017), 1–18. https://doi.org/10.3127/ajis.v21i0.1563

[3] Pedro Alonso, Rajkumar Saini, and György Kovacs. 2020. TheNorth at SemEval-2020 Task 12: Hate Speech Detection Using RoBERTa. In *The 14th Workshop on Semantic Evaluation*. ICCL, 2197–2202.

[4] Uttara M. Ananthakrishnan and Catherine E. Tucker. 2021. The Drivers and Virality of Hate Speech Online. *SSRN Electronic Journal* (2021), 1–32. https://doi.org/10.2139/ssrn.3793801

[5] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Learning for Hate Speech Detection in Tweets. In *The 26th International Conference on World Wide Web Companion*. ACM, 759–760. https://doi.org/10.1145/3041021.3054223

[6] Jack Bandy and Nicholas Diakopoulos. 2021. More Accounts, Fewer Links. In *The ACM Conference on Human-Computer Interaction*, Vol. 5. ACM, 1–28. https://doi.org/10.1145/3449152

[7] Michele Banko, Brendon MacKeen, and Laurie Ray. 2020. A Unified Taxonomy of Harmful Content. In *The 14th Workshop on Online Abuse and Harms*. ACL, 125–137. https://doi.org/10.18653/v1/2020.alw-1.16

[8] Michał Bilewicz and Wiktor Soral. 2020. Hate Speech Epidemic. The Dynamic Effects of Derogatory Language on Intergroup Relations and Political Radicalization. *Political Psychology* 41, S1 (2020), 3–33. https://doi.org/10.1111/pops.12670

[9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. https://doi.org/10.1162/tacl_a_00051

[10] David M. Chan, Roshan Rao, Forrest Huang, and John F. Canny. 2018. T-SNE-CUDA: GPU-Accelerated T-SNE and its Applications to Modern Data. In *The 30th International Symposium on Computer Architecture and High Performance Computing*. IEEE, 330–338. https://doi.org/10.1109/CAHPC.2018.8645912

[11] Chen Chen, Hanghang Tong, B. Aditya Prakash, Charalampos E. Tsourakakis, Tina Eliassi-Rad, Christos Faloutsos, and Duen Horng Chau. 2015. Node Immunization on Large Graphs: Theory and Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 28, 1 (2015), 113–126. https://doi.org/10.1109/TKDE.2015.2465378

[12] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. 2003. Efficient Immunization Strategies for Computer Networks and Populations. *Physical Review Letters* 91, 24 (2003), 247901. https://doi.org/10.1103/PhysRevLett.91.247901

[13] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *The 58th Annual Meeting of the Association for Computational Linguistics*. ACL, 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *The 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[15] Paula Fortuna, Juan Soler, and Leo Wanner. 2020. Toxic, Hateful, Offensive or Abusive? What Are We Really Classifying? An Empirical Analysis of Hate Speech Datasets. In *The 12th Language Resources and Evaluation Conference*. ERLA, 6786–6794.

[16] Zakariya Ghalmane, Mohammed El Hassouni, and Hocine Cherifi. 2019. Immunization of networks with non-overlapping community structure. *Social Network Analysis and Mining* 9, 1 (2019), 45:1–45:22. https://doi.org/10.1007/s13278-019-0591-9

[17] Sergei Ivanov, Konstantinos Theocharidis, Manolis Terrovitis, and Panagiotis Karras. 2017. Content Recommendation for Viral Social Influence. In *The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 565–574. https://doi.org/10.1145/3077136.3080788

[18] David Kempe, Jon Kleinberg, and Eva Tardos. 2015. Maximizing the Spread of Influence through a Social Network. *Theory of Computing* 11, 1 (2015), 105–147. https://doi.org/10.4086/toc.2015.v011a004

[19] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*. 1–14.

[20] Jens Lemmens, Ilia Markov, and Walter Daelemans. 2021. Improving Hate Speech Type and Target Detection with Hateful Metaphor Features. In *The 4th Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*. ACL, 7–16. https://doi.org/10.18653/v1/2021.nlp4if-1.2

[21] Xianghua Li, Jingyi Guo, Chao Gao, Leyan Zhang, and Zili Zhang. 2018. A hybrid strategy for network immunization. *Chaos, Solitons & Fractals* 106 (2018), 214–219. https://doi.org/10.1016/j.chaos.2017.11.029

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692

[23] Alvis Logins and Panagiotis Karras. 2019. An Experimental Study on Network Immunization. In *The 22nd International Conference on Extending Database Technology*. OpenProceedings, 726–729. https://doi.org/10.5441/002/edbt.2019.97

[24] Ilia Markov and Walter Daelemans. 2021. Improving Cross-Domain Hate Speech Detection by Reducing the False Positive Rate. In *The 4th Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*. ACL, 17–22. https://doi.org/10.18653/v1/2021.nlp4if-1.3

[25] Ilia Markov, Nikola Ljubešić, Darja Fišer, and Walter Daelemans. 2021. Exploring Stylometric and Emotion-Based Features for Multilingual Cross-Domain Hate Speech Detection. In *The 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. ACL, 149–159.

[26] Edoardo Mosca, Maximilian Wich, and Georg Groh. 2021. Understanding and Interpreting the Impact of User Context in Hate Speech Detection. In *The 9th International Workshop on Natural Language Processing for Social Media*. ACL, 91–102. https://doi.org/10.18653/v1/2021.socialnlp-1.8

[27] Mariam Orabi, Djedjiga Mouheb, Zaher Al Aghbari, and Ibrahim Kamel. 2020. Detection of Bots in Social Media: A Systematic Review. *Information Processing & Management* 57, 4 (2020), 102250. https://doi.org/10.1016/j.ipm.2020.102250

[28] Jose L. Part and Oliver Lemon. 2017. Incremental online learning of objects for robots operating in real environments. In *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*. IEEE, 304–310. https://doi.org/10.1109/DEVLRN.2017.8329822

[29] Andraž Pelicon, Ravi Shekhar, Matej Martinc, Blaž Škrlj, Matthew Purver, and Senja Pollak. 2021. Zero-shot Cross-lingual Content Filtering: Offensive Language and Hate Speech Detection. In *The EACL Hackashop on News Media Content Analysis and Automated Report Generation*. ACL, 30–34.

[30] Sancheng Peng, Guojun Wang, Yongmei Zhou, Cong Wan, Cong Wang, Shui Yu, and Jianwei Niu. 2019. An Immunization Framework for Social Networks Through Big Data Based Influence Modeling. *IEEE Transactions on Dependable and Secure Computing* 16, 6 (2019), 984–995. https://doi.org/10.1109/TDSC.2017.2731844

[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *The 2014 Conference on Empirical Methods in Natural Language Processing*. ACL, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[32] Alexandru Petrescu, Ciprian-Octavian Truica, and Elena-Simona Apostol. 2019. Sentiment analysis of events in social media. In *The 15th International Conference on Intelligent Computer Communication and Processing*. IEEE, 143–149. https://doi.org/10.1109/ICCP48234.2019.8959677

[33] Jing Qian, Hong Wang, Mai ElSherief, and Xifeng Yan. 2021. Lifelong Learning of Hate Speech Classification on Social Media. In *The 2021 Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, 2304–2314. https://doi.org/10.18653/v1/2021.naacl-main.183

[34] Yizhi Ren, Mengjin Jiang, Ye Yao, Ting Wu, Zhen Wang, Mengkun Li, and Kim-Kwang Raymond Choo. 2018. Node Immunization in Networks with Uncertainty. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering*. IEEE, 1392–1397. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00193

[35] Devendra Singh Sachan, Manzil Zaheer, and Ruslan Salakhutdinov. 2019. Revisiting LSTM Networks for Semi-Supervised Text Classification via Mixed Objective Function. In *The 2019 AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI, 6940–6948. https://doi.org/10.1609/aaai.v33i01.33016940

[36] Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and Psychological Effects of Hateful Speech in Online College Communities. In *The 10th ACM Conference on Web Science*. ACM, 255–264. https://doi.org/10.1145/3292522.3326032

[37] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The Risk of Racial Bias in Hate Speech Detection. In *The 57th Annual Meeting of the Association for Computational Linguistics*. ACL, 1668–1678. https://doi.org/10.18653/v1/P19-1163

[38] Sanjana Sharma, Saksham Agrawal, and Manish Shrivastava. 2018. Degree based Classification of Harmful Speech using Twitter Data. In *The 1st Workshop on Trolling, Aggression and Cyberbullying*. ACL, 106–112.

[39] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 11 (2008), 2579–2605.

[40] Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *The 1st Workshop on Abusive Language Online*. ACL, 78–84. https://doi.org/10.18653/v1/W17-3012

[41] Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*. ACL, 88–93. https://doi.org/10.18653/v1/N16-2013

[42] Tetsuya Yoshida and Yuu Yamada. 2017. A Community Structure-Based Approach for Network Immunization. *Computational Intelligence* 33, 1 (2017), 77–98. https://doi.org/10.1111/coin.12082

[43] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *The 13th International Workshop on Semantic Evaluation*. ACL, 75–86. https://doi.org/10.18653/v1/S19-2010

[44] Yao Zhang and B. Aditya Prakash. 2015. Data-Aware Vaccine Allocation Over Large Networks. *ACM Transactions on Knowledge Discovery from Data* 10, 2 (2015), 1–32. https://doi.org/10.1145/2803176