This project has received funding from the European Union's Horizon 2020 research and innovation programme under **grant agreement No 780355**

European Commission

FANDANGO

FANDANGO DELIVERABLE

Deliverable No.:	D4.4
Deliverable Title:	Source Credibility Scoring, Profiling and Social Graph Analytics Prototypes
Project Acronym:	Fandango
Project Full Title:	FAke News discovery and propagation from big Data and artificial intelliGence Operations
Grant Agreement No.:	780355
Work Package No.:	WP4
Work Package Name:	Fake News Identifiers, machine learning and data analytics
Responsible Author(s):	David Martín, Gustavo Hernández, Federico Álvarez
Date:	31.07.2019
Status:	V1.0
Deliverable type:	REPORT
Distribution:	PUBLIC

REVISION HISTORY

VERSION	DATE	MODIFIED BY	COMMENTS
V0.1	20.05.2019	David Martín (UPM), Gustavo Hernández (UPM)	First draft, contribution 1.1
V0.2	20.05.2019	David Martín (UPM), Gustavo Hernández (UPM)	Internal review
V0.3	20.06.2019	David Martín (UPM), Gustavo Hernández (UPM)	contributions, 2 and 3
V0.4	27.06.2019	Thodoris semertzidis (CERTH)	Contributions by CERTH. Section State-of-the-Art
V0.5	5.07.2019	Gustavo Hernández (UPM)	Contributions to section 3
V0.6	12.07.2019	David Martín Gutierrez (UPM)	Contibutions to section 4
V0.7	15.07.2019	Panagiotis Stalidis	General Review and CERTH contributions
V0.8	18.07.2019	David Martín	General Review
V0.9	20.07.2019	Gustavo Hernández	Conclusions and general review.
V1.0	22.07.2019	Gustavo Hernández (UPM)	Final Version

TABLE OF CONTENTS

Y

ABBREVIATIONS

ABBREVIATION	DESCRIPTION
H2020	Horizon 2020
EC	European Commission
WP	Work Package
EU	European Union
API	Advanced Programming Interface
GCN	Graph Convolutional Network
NLP	Natural Language Processing
NN	Neural Network
DL	Deep Learning
GPU	Graphical Process Units
CNN	Convolutional Neural Networks

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

This deliverable is an official document of the FANDANGO project funded by the European Union's Horizon 2020 (H2020) research and innovation programme under grant agreement No 780355. It is a public report that describes the technical requirements for FANDANGO platform and services.

This deliverable is divided into three main chapters: *Firstly*, is a complete review of the literature for the FANDANGO subjects (graph analysis for multiple purposes) from multiple technical perspectives (Neural Networks, also referred to as Graph Convolutional Networks and Network approaches). *Secondly*, it reports the activities performed, the algorithms selection, its implementation and the results obtained so far. *Last*, the future lines for the next period are outlined.

This document reports two main topics of the project: On the one side, it provides the State-of-the-art of reliable entities profiling, source scoring and media analysis. By entities, it is understood each of the components in the data model. This data model was drafted in *D2.1 - Data lake integration plan*, and extended in *D3.1 - Data model and components* and/or *Project Progress Periodic Reports*. To sum up, these entities are: "Organization", "Person", "Article", "Media"; "Claim","ClaimReview";"Fact".

On the other side, this deliverable is focused on reporting the activities performed aiming at both profiling the entities for the aforementioned ambition. This deliverable is associated to Task 4.4. and consequently, it contains the actions made to profiling entities from a graph based perspective. The idea behind is that the relationships between the diverse entities can be modelled as graphs (i.e. nodes, representing the entities, and edges that express the relationship) to apply complex graph processing algorithms.

Therefore, in this report, the main techniques applied for entities profiling, source credibility and social media analysis are described. This is a continuous evolving process. Initially, extended approaches were implemented, and the results obtained are illustrated in this document. By the time of this report, efforts are being devoted to implement novel techniques based on Graph Neural Network techniques.

As the data lake evolves so will its documentation, becoming more descriptive and precise during the lifespan of the project. Therefore, this deliverable is also greatly complemented by content defined on sections of deliverables D2.1 and D3.1 to define more detailed data structures available in each repository and its conventions.

INTRODUCTION

FANDANGO's final goal is to integrate an information verification system, able to analyze different typologies of data, correlated with a news like images, video, text and metadata (source, author, etc). To achieve such a goal, several different tasks and functionalities need to be developed and integrated to provide a final tool where users can navigate and obtain useful results.

In particular, it has to be underlined how to match the tools and solutions presented with the user requirements to provide the information that they might require in the proper manner. Based on the user requirements, the analysis and definition of technical, operational and functional specifications.

From the user perspective, they would expect a type of support system that provides them with a "scoring" of trustworthiness. Actually, They do not want a system to make a hard decision (i.e. "fake", "real"), but a type of information that support their ideas, providing some guidelines, however, at the end the decision is going to be made by themselves (journalists).

According to the task, FANDANGO shall provide the mechanisms to profile the mentioned entities, serving scorings that will be used as relevant indicators for the decision making process. In this sense, this task will analyze the information coming from the preprocessing module, taking into account the diverse relationships that exist among the sources and the update process of scoring, then these relationships will yield to reliable indicators of the article "trustworthiness".

PLATFORM OVERVIEW

PLATFORM GENERAL DESCRIPTION

In order to achieve all the requested made by end-users in Deliverable 2.3, a new design for fandango's platform architecture has been created. In this new proposal, all the tasks and functionalities have been divided into two main workflows, called online and offline workflow.

The offline workflow is intended as all the processes to ingest data needed to build most of the machine learning, graph models and official data lake for Fandango. In fact the ground truth will be taken from the ingestion process and from the annotation system as well as information to build graphs about authors and organizations/publishers.

The diagram contains all the technologies involved and a full connection among them.

On the top of the diagram there are data typologies that are necessary to have all the services integrated in the platform. Kafka is used in order to handle asynchronous returns in the results, since images and videos analysis are much more time consuming than a prediction score calculation in terms of text and graph analysis. At the end, a final document will be ready to be stored on the no-sql database ElasticSearch.

The data ingestion is intended as a continuous process, the more information the system can hold and add, the better performances in terms of relations and correlations among articles, authors, publishers, topics and entities will be.

The online flow describes all the services/modules interconnections to be integrated in the web-application.

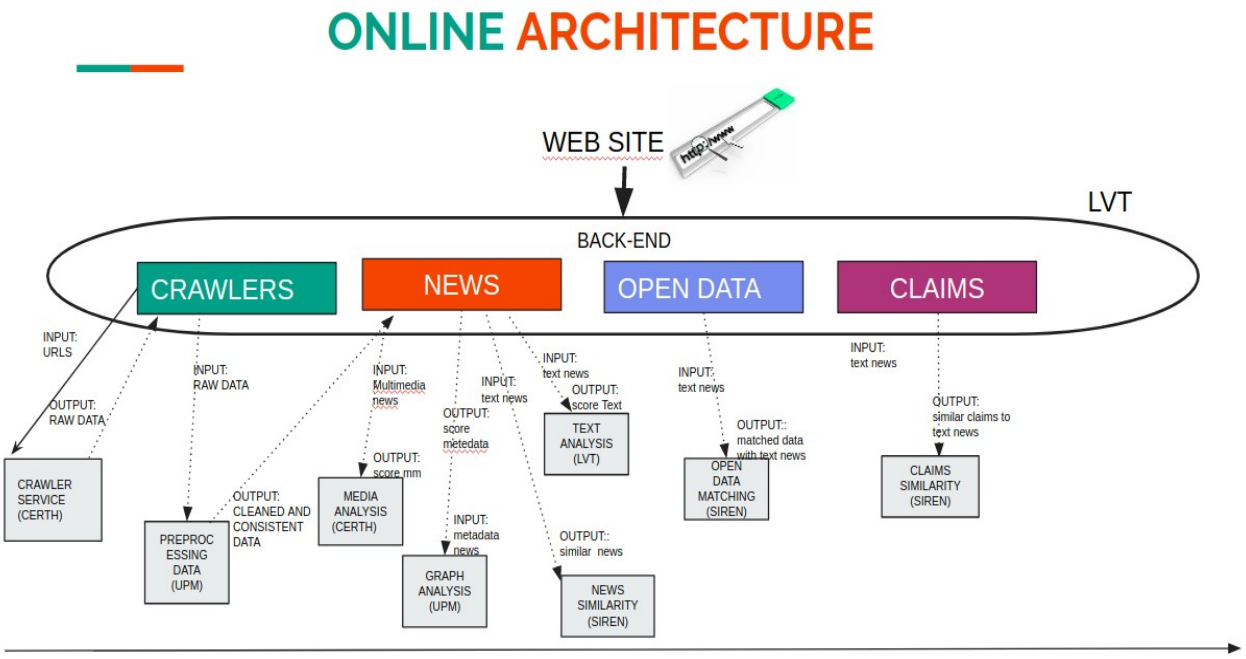


Figure 1 General Overview of FANDANGO online services

Through the web-app several processing flows to analyze the trustworthiness of a news it can be initiated: the first of these flows executes sequentially the **Crawler** service, **Pre-Processing**, **Text Analysis**, **Meta Data (Graph) Analysis**, **Open Data Matching** and **Similar News**. The second processing flow retrieves the most similar claims by means of **Entity service** and **Claim Similarity** service. Finally, the last processing flow is executed in order to analyze images or videos by means of the **Media Analysis** service**.**

PLATFORM NON-FUNCTIONAL ASPECTS

The platform set up to support the previously described analytics, both from a batch and an online perspective, enables scalable pipelines and processes. The state of the art of the project is implementation phase, relying on a development environment which benefits from a cloud infrastructure on MS Azure. At this point data is ingested in bulk mode too, in order to collect all of the available raw data to archive historical data for the further analysis. The final amount of raw data to ingest and to implement for the use-cases is still to be defined, as well as the system doesn't have specific SLAs to grant.

Thanks to this configuration the Fandango architecture, which is composed by scalable tools and services, is comfortable to achieve all of the desired performances that the final product must satisfy.

From a technical aspect all of the services are deployed as simple instances that can work in parallel, as well as the rest of the processes. In order to allow many simultaneous users/developers to access the

system and set-up/run the services, the platform is delivering the right match of resources in terms of RAM and CPUs. Whenever some fault is detected or more service instances need to be run, exceeding the amount of computational resources, the cluster can be extended both in scale out/scale up mode.

All of the issues that can concern the storage dedicated to memorize raw data as well as the output of the services, plus the disk storage for the final results is fulfilling the current requirements, since the system can benefit from the hard disk of the Elasticsearch cluster, local storage within the containers, the persistence layer (relying on disk storage) for the retention in Kafka and, above all for intermediate results Azure Blob Storage, the MS Azure object store, can be used.

Since this is a complex system, composed by several services and tools, many quantitative aspects of the way the whole platform performs, depending on the actual use-cases in terms of processes and storage. Setting a proper prototyping during this phase of the project, can help identifying the perfect hardware provisioning for the final production environment.

1. STATE OF THE ART

Section Summary: This section presents a complete overview of the trending techniques for source credibility, fact checking and back-trace propagation of news. Section 1.1 presents the Deep Learning approaches whereas section 1.2. outlines network graph (graph theory) approaches.

Fake news recognition has a particular growing interest as the circulation of misinformation on-line increase. This problem, in Fandango, will be faced in two different ways; on one hand, considering text and title of an article and analyzing its syntax and some linguistic features. On the other hand, building a graph to collect and connect authors and organizations in order to apply some graph analysis. In the following two paragraphs both approaches will be described in detail.

1.1. GRAPH CONVOLUTIONAL NETWORKS APPROACH

Graphs have been used to denote information in diverse areas such as communication networks, biology, linguistics and social sciences. Modelling the interactions between entities as graphs has enabled researchers understand the various network systems in a systematic manner.

The problem of node classification concerns the use of the already labelled nodes of a network and its topology for the purpose of predicting a label for an unclassified node of this network.

Concerning the network topology features, they can be either directly or indirectly extracted from the adjacency matrix of the network and therefore their computation is bounded from the graph size, irrespective of the category of the classification method. The method and the developed system in this report introduce the idea that even when the network undergoes analysis with multiple subsets of the aforementioned input matrix, the graph embedding that are extracted can be as meaningful as if the whole matrix were used.

In the next sections, the work in[1] is extended: the tool used is a Graph Convolutional Network (GCN), a special neural network that performs operations on the full *adjacency matrix* of the input graph (or the Laplacian of the graph). In order to clarify, the graph adjacency matrix, is a diagonal matrix that contains binary values: being 1 in the position of two nodes that are related (linked) and 0 in nodes without direct link relationship. The output is a vector representation of each graph node that can be used in various clustering or classification applications.

Nevertheless, the requirement of any algorithm to work with a full adjacency matrix or a Laplacian of a graph bounds the applicability of the algorithm to the available computational resources (RAM, GPU etc.).

The current report the following assumption (researched in[2]) is evaluated: the training of a GCN with multiple subsets of the full data matrix is possible and converges to the full data matrix training scores.

The experimental results in^2 validate that similar classification accuracy can be achieved in the same computational resources as the other methods, at a cost in the number of training epochs. This permits the processing of large graphs (e.g., in the order of millions of nodes) as input, providing a scalable approach in the node classification task, which was otherwise impossible for this kind of neural networks.

In real-world problems, semi-supervised classification would make sense in large datasets that are difficult to annotate. Therefore the barrier of the limited resources is dropped allowing for the approach of this report to apply in large graphs using subgraphs of the original graph.

Additionally, by applying several methodologies for formulating subgraphs of better quality, it is demonstrated that the use of specific community detection algorithms lead to considerable improvements.

1.1.1. LITERATURE REVIEW

The problem under investigation emerges from the fields of graph embedding and node classification. As stated in[3], the task of node embedding concerns the mapping of each node in a new space preserving a proximity measure. The proximity of nodes can be represented by the similarity of their corresponding

1 Kipf, Thomas N and Welling, Max. Semi-supervised classification with graph convolutional networks. ICLR 2017.
2 G. Palaiopanos, P. Stalidis, N. Vretos, T. Semertzidis, P. Daras, 'Embedding Big Data in Graph Convolutional Networks', 25th International Conference on Engineering, Technology and Innovation (ICE/IEEE ITMC 2019), Sophia Antipolis Innovation Park, France, 17-19 June 2019.
3 Cai, Hongyun, Vincent W. Zheng, and Kevin Chen-Chuan Chang. "A comprehensive survey of graph embedding: Problems, techniques, and applications." IEEE Transactions on Knowledge and Data Engineering 30.9 (2018): 1616-1637.

vectors. Deep learning (DL) based graph embeddings can be applied either on the whole of a graph or on the nodes of the graph.

The problem includes feature extraction from nodes for the subsequent classifying operation on them, such as in[4] and[5]. The relevant solutions and algorithms aim to label nodes based on the information they encode. The input can be either grid-like data (image, a video frame, sequence of video frames, text) or interconnected raw data (graphs). 'Geometric deep learning: going beyond Euclidean data[6] provides an explanatory list of the geometric aspect of this research.

Convolutional Neural Networks (CNN) have proven popular for graph embedding in the area of geometric deep learning. In^3, the methods are distinguished in two groups. The first contains CNNs that work in Euclidean domains and reformat input graphs to fit it. For example in[7] perform a node-ordering creating a sequence and then uses the CNN to learn a neighborhood representation. In the second group, the deep neural models operate in non-Euclidean domains (a leading example is graphs). Each approach builds a convolutional filter which is applied in the input graph and can be either spectral[8910] or spatial, i.e., a neighborhood matching[1112].

In Semi-supervised classification with graph convolutional networks[1] the authors provide a solution scalable in the number of edges. The next sections of this report extend this classification system in the case of million scale (order) of nodes managing to have it as input in the GPU and achieving the same or higher accuracy, at the same time.

1.1.2. Methodology: Graphs and Convolutional Neural Networks CNNs

4 Bhagat, Smriti, Graham Cormode, and Irina Rozenbaum. "Applying link-based classification to label blogs." International Workshop on Social Network Mining and Analysis. Springer, Berlin, Heidelberg, 2007.

5 Lu, Qing, and Lise Getoor. "Link-based classification." Proceedings of the 20th International Conference on Machine Learning (ICML-03). 2003.

6 Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine, 34(4), 18-42.

7 Niepert, Mathias, Mohamed Ahmed, and Konstantin Kutzkov. "Learning convolutional neural networks for graphs." International conference on machine learning. 2016

8 Defferrard, Michaël, Xavier Bresson, and Pierre Vandergheynst. "Convolutional neural networks on graphs with fast localized spectral filtering." Advances in neural information processing systems. 2016

9 Henaff, Mikael, Joan Bruna, and Yann LeCun. "Deep convolutional networks on graph-structured data." arXiv preprint arXiv:1506.05163 (2015).

10 Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203.

11 Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. IEEE Transactions on Neural Networks, 20(1), 61-80

12 Zhang, C., Zhang, K., Yuan, Q., Peng, H., Zheng, Y., Hanratty, T., ... & Han, J. (2017, April). Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In Proceedings of the 26th International Conference on World Wide Web (pp. 361-370). International World Wide Web Conferences Steering Committee.

The methodology explored in this work follows[2] and suggests a solution to the following graph-based learning problem: for each node *i* of a graph *G*, the prediction of a label y_i from a predefined set of labels is needed. In order to make this prediction one uses the available features x_i defined for each node in the form of observed numerical attributes and the weighted edges of the graph, usually represented by an adjacency matrix *A*. The main assumption is that, when predicting the output y_i for node *i*, the attributes and connectivity of nearby nodes provide useful side-information or additional context for the training of the above CNN.

A limitation in processing datasets (particularly the corresponding adjacency matrix) as large as millions of entities in the available resources cannot be satisfied using the existing deep learning models. In spite of the fact that feeding the model with all possible subgraphs would at some point contain the whole of the network information, this is not practically achievable, as the number of these increases exponentially with the number of nodes in the graph. To face this, the current approach uses a smaller number of subgraphs in the form of submatrices of the adjacency matrix. It is shown experimentally that, given a fixed number of nodes as size, it suffices to use the minimum number of subgraphs of this size in order to approximate the accuracy rate of different approaches, as long as every node is included in some subgraph and the computation remains efficient.

The developed system is based on the work 'Semi-supervised classification with Graph Convolutional Networks (GCN)[1], where the authors introduce a layer-wise propagation rule:

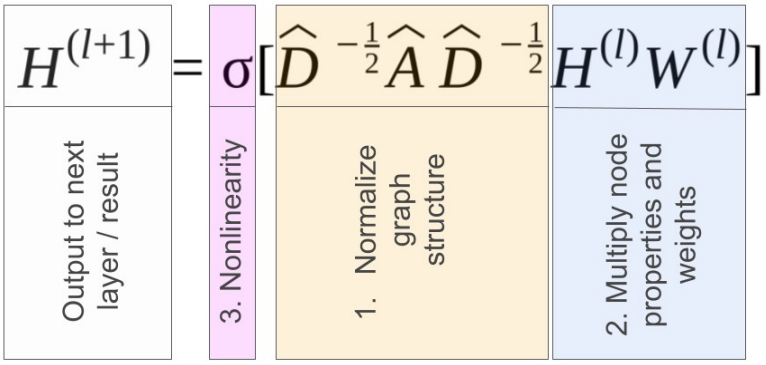


Figure 2: Layer rule

- $\hat{A} = A + I$ is the adjacency matrix of the undirected graph G with added self-connections,
- D is the degree (diagonal) matrix and is positive semidefinite[1]. $\hat{D}$ is the normalized version of D.
- $W^{(l)}$ is a layer-specific trainable weight matrix
- σ() denotes an activation function, such as ReLU.
- $H^{(l)}$ is the Nxd real matrix of activations in the l^{th} layer, N is the number of nodes and d the number of features per node.

This rule allows for neural networks to operate directly on graphs. This is motivated from first-order approximation of spectral graph convolutions, and in their work they demonstrate how a multi-layer Graph Convolutional Network model can be used for embedding network structure information.

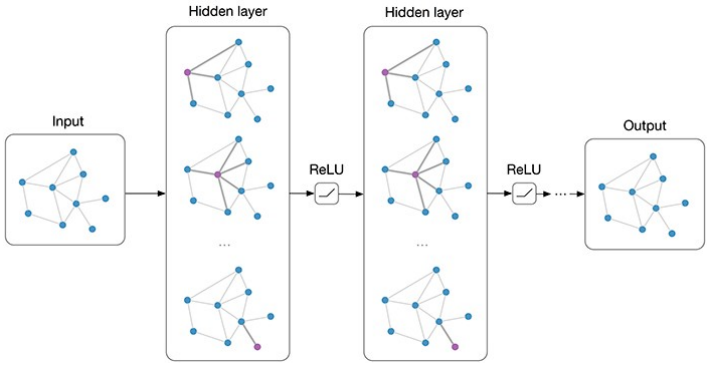


Figure 3: Graph convolutional neural network

Following the previous equation, each node embedding includes information about the number and quality of all the neighbors of the node (also called 'first order neighbors'). Recursively, for the ith layer, the embedding information contains the information embedded in the *(l-1)th* layer.

In the current case, the data will be arranged in subgraphs which leads to the use of submatrices S_i of the adjacency matrix *A*. The selection strategy is investigated along with the way that the sampling procedure impacts the classification performance.

1.1.3. DATASETS

The first experiments were carried out for semi-supervised document classification in a citation network dataset called 'Cora'. Afterwards, spammer classification was performed in a social network graph dataset called 'Social Spammer' in a fully and semi-supervised manner.

- The Cora dataset[13] consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.

Dataset	Type	Nodes	Edges	Classes	Features	Label rate[a]
Cora	Citation network	2,708	5,429	7	1,433	0.052
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001
Social Spammer	Social network	5,607,447	858,247,099	2	4	0.05

[a]Rate of known labels in the semi-supervised classification experiments

Table 1: DATASETS

13 Cora Dataset. url: https://linqs.soe.ucsc.edu/data, LINQS, 2015.

- NELL is a dataset extracted from the knowledge graph introduced in [14]. The pre-processed version[15] of the data is used [1]. A knowledge graph is a set of entities connected with directed, labeled edges (relations). Entity nodes are described by sparse feature vectors. The number of features in NELL were extended by assigning a unique one-hot representation for every relation node, effectively resulting in a *61,278-dimensional* sparse feature vector per node. The semi-supervised task here considers the extreme case of only a single labeled example per class in the training set. The binary, symmetric adjacency matrix from this graph was constructed by setting entries $A_{ij}=1$, if one or more edges were present between nodes *i* and *j*.
- Social Spammer dataset[16] is a social network graph dataset originally published for the task of identifying the spammer users based on their relational and non-relational features. It contains 5.6 million users of the Tagged.com social network website and 858 million links between them. Each user has 4 features and is manually labeled as "*spammer*" or "*not spammer*". Each link represents an action between two users and includes a timestamp and one of 7 anonymized types of link. For the training set 4.5 million users are randomly selected and 0.5 million users are used for validation purposes. The testing set is comprised by the remaining 607 thousand users. With this dataset, there were experiments implemented both in a fully and a semi-supervised manner. For the semi-supervised experiments, 95% of the training set nodes was randomly selected and their label was deleted.

1.1.4. EXPERIMENTAL PARAMETERS

The graph convolution layer from the *Keras* implementation[17] is used and the same experimental setup as in[18] is followed.

For the Cora dataset, a 5-layer model is employed: an input layer and alternating Dropout and GCN layers. The first GCN layer uses 16 hidden units and the second GCN layer has 7 hidden units. The activation function of the first layer is *ReLU* and a constraint of L2 regularization of 5x10-4 is applied on the kernel weights. Both the Dropout layers mask 0.5 of the input features. The second GCN layer is the last layer of the model, so the activation function applied is the *softmax*. For the training process an Adam solver with a learning rate of 0.001 is used. The loss function is categorical cross-entropy. The node features are pre-processed by applying standard normalization and the *localpool* renormalization trick from^6 is used for the neighborhood features. The model is trained for 500 epochs.

For the NELL dataset, the only differences are that 64 hidden units are used in the first GCN layer (instead of 16), the dropout is reduced from 0.5 to 0.1. The L2 constraint is changed from 5e-4 to e-5.

The Social Spammer dataset: it is much larger, so a larger model with more parameters is employed. It consists of 3 GCN layers and a fully connected classification layer. Before each layer a dropout of 0.1 is

14 Zhang, C., Zhang, K., Yuan, Q., Peng, H., Zheng, Y., Hanratty, T., ... & Han, J. (2017, April). Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In Proceedings of the 26th International Conference on World Wide Web (pp. 361-370). International World Wide Web Conferences Steering Committee.

15 Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., & Mitchell, T. M. (2010, July). Toward an architecture for never-ending language learning. In Twenty-Fourth AAAI Conference on Artificial Intelligence.

16 Fakhraei, Shobeir and Foulds, James and Shashanka, Madhusudana and Getoor, Lise. Social Spammer Dataset. Proceedings of the 21th Acm Sigkdd international conference on knowledge discovery and data mining, pages 1769--1778, ACM, 2015, url: https://linqs-data.soe.ucsc.edu/public/social_spammer

17 Thomas Kipf website. url: https://github.com/tkipf/keras-gcn. Github 2016.

18 Kipf, Thomas N and Welling, Max. Semi-supervised classification with graph convolutional networks. ICLR 2017.

applied but no weight regularization was performed. The Adam solver is applied here as well, but with the default Keras learning rate of 0.01. Since the number of spammers is much smaller than the number of non-spammers, the ratio of spammer to non-spammer users is calculated and the loss function was modified to give higher weight to the under-represented class in order to have a more balanced learning.

For the semi-supervised experiments, the loss is computed only from the nodes that have a 'known' target label. As noted in[6], the initialization and dropout schemes of Keras and Tensorflow (used in the original paper) are different, so the reported accuracy of the original GCN model might present slight differences than the one reported by this report.

The tried experimental approach followed the stages below:

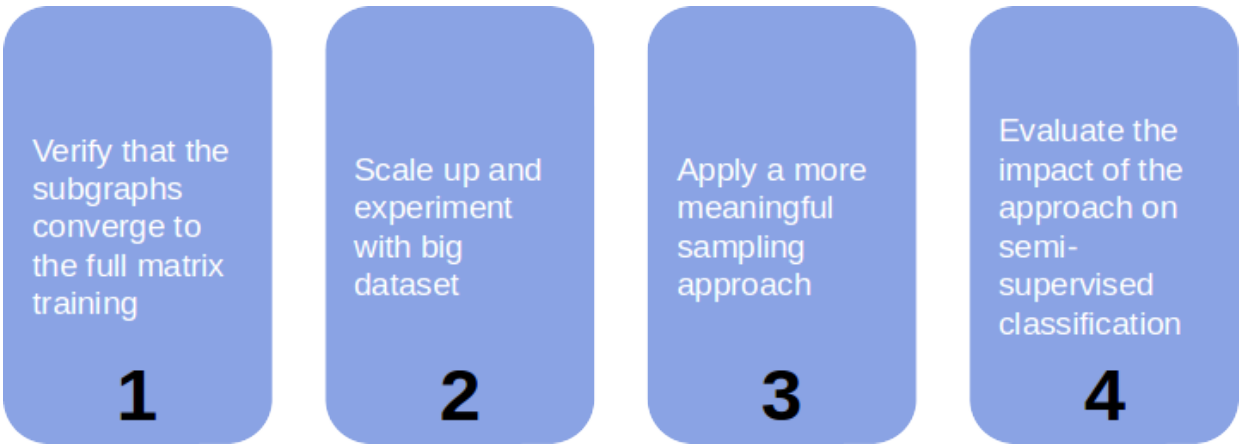


Figure 4 Experimental Stages

1.1.5. SAMPLING STRATEGY

In order to split the original graphs to smaller subgraphs of the same size (K nodes) that would fit in the GPU, a random split methodology is initially used. To avoid dividing by zero while calculating the loss, at least one node with a known label in each subgraph is added. The rest of the nodes are randomly picked so that all nodes belong to exactly one subgraph. The creation of random, same-sized subgraphs is repeated in each training epoch.

This procedure of fragmentation is very likely to create subgraphs with nodes that have very sparse connectivity. This phenomenon leads to subgraphs that practically do not contain any of the inherent connectivity features of the original network. Even though it is assumed that after numerous training epochs an adequate number of densely connected subgraphs will be fed to the system allowing for the extraction of the inherent network features, a smarter way of creating the subgraphs may lead to faster training times.

1.1.5.1. STAGE 1: SAMPLING VERSION OF BASELINE PROBLEMS & RESULTS

In order to test and measure the performance of training the Graph Convolutional neural Network with random subgraphs, two experiments were conducted with splits of the graph into 2 and 8 subgraphs (Table 2, accuracy is shown in percentages). In the case of randomly splitting Cora dataset into two subgraphs, the GCN model reaches classification score only 1-2% lower than the GCN's accuracy with the whole of the input.

The 'Planetoid' and 'GCN original method' results are drawn from 'Semi-supervised classification with graph convolutional networks', noted in[7]. The baseline result is from a two layer

MLP model, where all configuration parameters are the same as the GCN model. The 'GCN whole input' result is the accuracy score that is achieved in the run of the GCN model where the input is the whole of the graph adjacency matrix.

SUMMARY OF RESULTS FOR DIFFERENT SIZES OF RANDOM SUBGRAPHS IN TERMS OF CLASSIFICATION ACCURACY

Method	Cora	NELL
Baseline	61.10	39.41
Planetoid	75.70	61.90
GCN original method	81.50	66.00
GCN Whole input	81.20	43.60
GCN 2 Subgraphs	79.30	41.68
GCN 8 Subgraphs	78.35	39.41
GCN 25 Subgraphs	60.20	33.54

Table 2

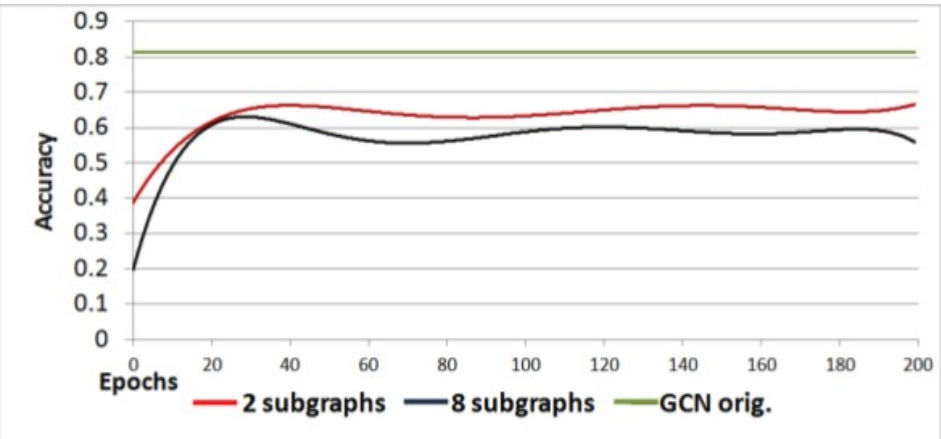


Figure 5 Evolution of accuracy during raining of GCN model on the Cora dataset for the whole input and subgraphs.

Figure 5 shows the evolution of accuracy percentage during training of the GCN model on the ``Cora'' dataset for the case of taking the input as a whole (green), for the case of splitting it into 2 subgraphs (red) and into 8 subgraphs (blue).

Normally all existing permutations of the nodes should be included in the subgraphs for the connectivity configuration to be properly embedded in the training phase. Nevertheless, the above experiments demonstrate that the CNN encodes the graph information with approximately the same quality without the need of all subgraphs for training.

The rest of the results in Table 2 show that splitting the input into more subgraphs for training worsens the accuracy of the network, a fact which is due to the random fragmentation and the consequent unsystematic information contained in them.

1.1.5.2. Stage 2: Scaled-UP Problem (Big Data): Results

For the case of large graph-based datasets (millions of nodes), training is impossible with the whole of the data because memory cannot accommodate their graph representation (GCN model requires O(n2) parameters). E.g. in the case of 'Social Spammer' dataset, n = 1013 vertices and dataset cannot be handled by the existing resources. On the other hand, the suggested idea of subgraphs (of size n = 105) allows the graph to be stored in memory and used for training. This procedure requires a greater number of iterations, but the aggregate duration is of the scale of several hours.

Table III reflects the testing metrics for the node classification problem in the previously mentioned dataset. The employed neural network contains 3 layers and a fully connected classification layer.

FULLY SUPERVISED NODE CLASSIFICATION USING NODE FEATURES VS SUBGRAPHS

Method	Fully-supervised	
	Accuracy (%)	AUROC
Baseline method	81.97	0.8072
Fakhraei et al	-	0.8170
Random S-GCN 16	81.52	0.7927
Random S-GCN 64	82.33	0.8115
Random S-GCN 256	85.59	0.8134
Random S-GCN 1024	84.31	0.8120

Table 3

As illustrated therein, the GCN method is compared to a multi-layer perceptron (MLP) as a baseline method with 4 layers of 256 hidden units. This method uses solely the features of the nodes without considering any of their interconnections. The next model for comparison is those trainings of Fakhraei et al[19] that contain graph-structured features.

For the current GCN applied on subgraphs of the input, after varying the number of hidden units, the most suitable one was found to be 256, where the fully-supervised variant gives AUROC 0.8134 slightly outranking the baseline and approximating the Fakhraei et al result. The rest of the subgraph selections are marginally lower in performance. Therefore, the learning effect of the MLP is equivalent to training the GCN using randomly chosen subgraphs. Consequently, a more strategic selection of the subgraphs and their structure is expected to boost the classification accuracy.

1.1.5.3. Communities in Graphs

For the purpose of formulating subgraphs in an organized way and inspired by the ideas of 'Curriculum learning'[20], a preprocessing of the dataset was developed with the goal to assist the learning process

19 Fakhraei, Shobeir and Foulds, James and Shashanka, Madhusudana and Getoor, Lise. Collective spammer detection in evolving multi-relational social networks. Proceedings of the 21th Acm Sigkdd international conference on knowledge discovery and data mining, pages 1769--1778, ACM.

20 Bengio, Yoshua and Louradour, Jerome and Collobert, Ronan and Weston, Jason. Curriculum learning. Proceedings of the 26th annual international conference on machine learning, pages 41-48, 2009, ACM.

embed the interconnections of the graph-based input.

The chosen solution path was the application of community detection algorithms which divide the graph into 'communities': they partition the node set into groups with high concentrations of edges in the interior of each group and low concentrations (sparse connectivity) between every pair of groups. The input to the GCN network is now samples drawn from these communities.

The formal definitions of communities are presented in 'Community detection in graphs'[21], where communities are described as separate entities with their own autonomy. Hence, they can be used for the training phase because it makes sense to evaluate them independently of the graph as a whole and are expected to share common properties in terms of classification e.g., to have the same label.

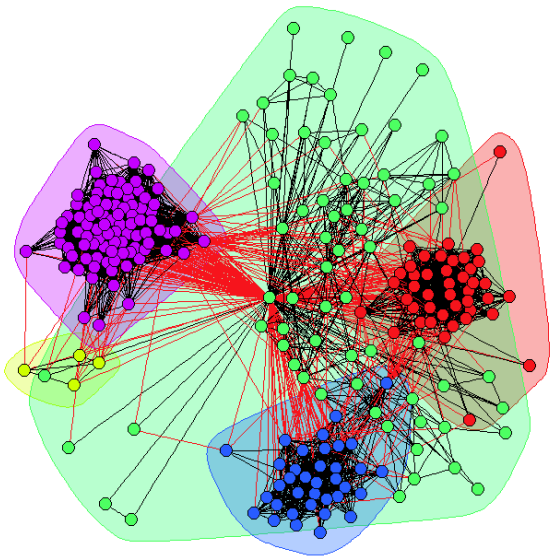


Figure 6 Communities in a graph

To first test this theory the *'Label Propagation'*[22] algorithm for community detection in large-scale networks is applied, using the *'iGraph'*[23] library, where vertices are initially given unique labels (e.g., their vertex labels). At each iteration, a sweep over all vertices, in random sequential order, is performed: each vertex takes the label shared by the majority of its neighbors. If there is no unique majority, one of the majority labels is picked at random. In this way, labels propagate across the graph: most labels will disappear, others will dominate. The process reaches convergence when each vertex has the majority label of its neighbors. Communities are defined as groups of vertices having identical labels at convergence. The time complexity of each iteration of the algorithm is $O(|E|)$, E being the edge set of the graph, and the number of iterations to convergence appears independent of the graph size, or growing very slowly with it, in the case of sparse graphs.

21 Fortunato, Santo. Community detection in graphs. Journal of Physics reports, 486, 3-5, pages 75-174, 2010, Elsevier

22 Raghavan, Usha Nandini, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks." Physical review E 76.3 (2007): 036106.

23 Csardi, Gabor, and Tamas Nepusz. "The igraph software package for complex network research." InterJournal, Complex Systems 1695.5 (2006): 1-9, url: https://igraph.org/python/.

When the Label Propagation method is applied in the Social Spammer dataset, the outcome is a very small number of communities containing the majority of the nodes and the remainder of the nodes are assigned to singleton communities. This community detection is deemed very poor for the purposes of this report and therefore the Louvain[24] algorithm was then implemented. Initially, all vertices of the graph are put in different communities. The first step consists of a sequential sweep over all vertices. Given a vertex i, one computes the gain in weighted modularity coming from putting i in the community of its neighbor j and picks the community of the neighbor that yields the largest increase, as long as it is positive. At the end of the sweep, one obtains the first level partition. In the second step communities are replaced by supervertices and two supervertices are connected if there is at least an edge between vertices of the corresponding communities. In this case, the weight of the edge between the supervertices is the sum of the weights of the edges between the represented communities at the lower level. The two steps of the algorithm are then repeated, yielding new hierarchical levels and supergraphs. At some iteration, modularity cannot increase any more, and the algorithm stops. As it will be shown below, it supplies partitioning's that result in a more credible output, at the expense of higher computational time (in comparison to the previous method): $O(|V|\log|V|)$,where *V* is the set of graph nodes.

For both methods of community detection that are used, the subgraphs are created firstly by randomly picking a community. Then the subgraph was filled with members of this community up to the preset size. If the community got exhausted, members of the next community were used. Consecutive subgraphs were created in the same manner, continuing with the already selected community.

In the experiments that follow, subgraph sizes of 50%, 12.5% and 4% of the dataset size (number of nodes) were tested.

1.1.5.4. Stage 3: Sampling Method in Big Data & Results

Following up the previous ideas, the current step exploits the benefits of a more sophisticated arrangement of the nodes in subgraphs (than simple random selection). The methodology used is to perform community detection on the graph in order to detect groups of nodes that contain denser connections in their interior and looser links to the other groups. Then, using these communities as the focus point of the selection process, an algorithm is devised to create subgraphs that are contained entirely or in their biggest part in the communities.

FULLY SUPERVISED NODE CLASSIFICATION USING NODE FEATURES VS SUBGRAPHS

Method	Fully-supervised	
	Accuracy (%)	AUROC
Baseline method	81.97	0.8072
Fakhraei et al [25]	-	0.8170
Random S-GCN 16	81.52	0.7927
Random S-GCN 64	82.33	0.8115
Random S-GCN 256	85.59	0.8134
Random S-GCN 1024	84.31	0.8120
LabelPropagation S-GCN 256	85.95	0.8185
Louvain S-GCN 256	**92.11**	**0.8187**

Table 4

24 Blondel, Vincent D., et al. "Fast unfolding of communities in large networks." Journal of statistical mechanics: theory and experiment 2008.10 (2008): P10008

As shown in Table IV, the first algorithm employed for community formulation is Label Propagation: there are fewer than 10 communities containing more than 80% of the nodes, while the remaining 20% of the nodes are assigned to singleton communities (containing a single member). Despite this uneven composition, the GCN model is able to create better embeddings than the random-sampling case, a fact that is depicted by the improved accuracy and AUROC scores (0.8185 in comparison to the previous 0.8134 AUROC metric for the supervised case).

Using Louvain algorithm as a community method provides a more balanced subgraph composition providing samples which lead to a raise in AUROC (0.8187). The testing accuracy reaches 92.11% outperforming all previous methods.

1.1.5.5. Stage 4: Semi-Supervised Classification in Big Data & Results

The last experiment is the semi-supervised training which yields the following results (Table V).

FULLY AND SEMI-SUPERVISED NODE CLASSIFICATION

Method	Fully-supervised		Semi-supervised	
	Accuracy (%)	AUROC	Accuracy (%)	AUROC
Baseline method	81.97	0.8072	70.49	0.6421
Fakhraei et al	-	0.8170	-	-
Random S-GCN 16	81.52	0.7927	75.50	0.7080
Random S-GCN 64	82.33	0.8115	79.13	0.7516
Random S-GCN 256	85.59	0.8134	87.94	0.7777
Random S-GCN 1024	84.31	0.8120	77.68	0.7719
LabelPropagation S-GCN 256	85.95	0.8185	91.13	0.7802
Louvain S-GCN 256	**92.11**	**0.8187**	**93.25**	**0.8023**

Table 5

In figure 6 there is the validation accuracy visualised at each training epoch.

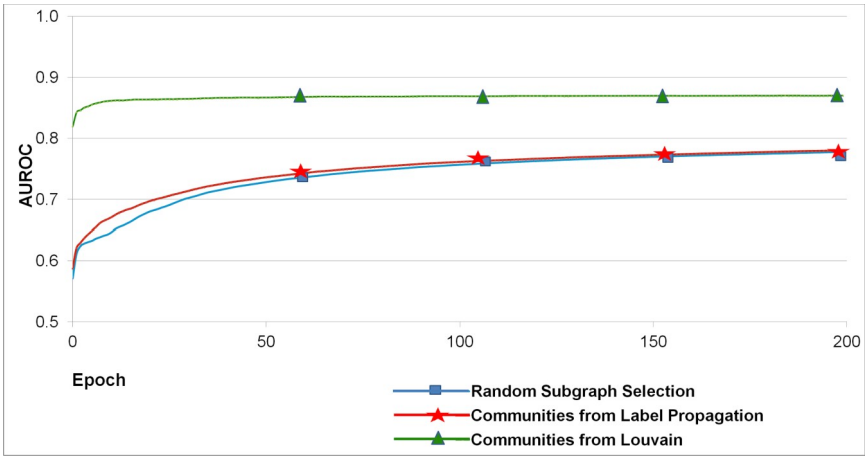


Figure 7 Training accuracy (evolution with respect to epochs) for the semi-supervised classification task using three different methods

Comparing the training process of the three sampling methods, it can be observed that sampling from Louvain communities not only produces better classification but also accelerates the learning process in comparison to sampling in both the Random and Label Propagation methods (higher convergence rate).

1.1.6. CONCLUSIONS

This report presented a method that allows semi-supervised node classification in large datasets using community detection algorithms to create the input samples for the used neural network. The existing graph convolutional networks are based on spectral graph convolutions or the full adjacency matrix of the input graph and thus prove ineligible when the size of the graph-data matrix is large, which is the case when dealing with big data (millions of nodes). The idea of using subgraphs, in order to feed the graph in the model, allows to scale up to millions of entities.

It is shown experimentally that, given a fixed number of nodes as size, it suffices to use the minimum number of subgraphs of this size for the training of a GCN in order to approximate the accuracy rate of baseline approaches, as long as every node is included in some subgraph; this computation remains efficient and lets the GCN process large datasets.

Moreover, the more targeted selection of subgraphs results in node embeddings superior to embeddings created from random subgraphs. The training time required to reach a certain classification quality is reduced when using targeted subgraphs compared to random ones, as well.

1.2. NETWORK APPROACHES

In this subsection, the main approaches covering the FANDANGO topics from a Network perspective (graph theory) are covered. We use the term Network to those proposals based on pure graph theory to make a difference with respect to the literature proposals that cover these topics using novel Deep Learning methods.

1.2.1. SOURCE CREDIBILITY STATE-OF-THE-ART

Most of the methods covered here, and in general, for the mentioned tasks are based on supervised learning: Learning a text / image / graph classifier from a bunch of labelled data provided for training. However, even novel methods for research still rely on fundamental aspects of previous research. In fact, challenges have been stated and reported in several previous works[25][26]. These works were considered to define the data-model in FANDANGO.

25 BJ Fogg, Jonathan Marshall, Othman Laraki, Alex Osipovich, Chris Varma, Nicholas Fang, Jyoti Paul, Akshay Rangnekar, John Shon, Preeti Swani, et al. 2001. What makes web sites credible?: a report on a large quantitative study. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 61–68. ACM.

26 BJ Fogg and Hsiang Tseng. 1999. The elements of computer credibility. In Proceedings of the SIGCHI conference on Human Factors in Computing Systems, pages 80–87. ACM

The task of automated fact-checking can be considered as one of the most challenging tasks in Natural Language Processing (NLP). Perhaps, the most difficult task is to design trustworthiness indicators associated with sources of information. Additionally, the complexity inherent in creating and connecting logical arguments makes even harder to find a reasonable solution. This is a basis to communicate and defend opinions (or claims within this context), to understand new problems and to perform scientific reasoning[27]. An alternative to face with this issue is known as argumentation mining methods, which are supposed to better understanding text structures and relations among entities, i.e., processing raw text in natural language to recover inferential structure[28]. However, the main drawback of such approaches is given by the lack of validation in real-world applications.

Nakamura et al[29] performed a classification survey for users of diverse age ranges by understanding how they identify the reliable websites. With the results obtained, they built a graph based ranking method to help users in judging the trustworthiness of search results retrieved by a search engine. An investigation conducted by Stanford University revealed important aspects, especially visual, considered by volunteers when facing with websites (to decide whether are trustworthy or not)[30]

The writing style also play a key role in the determining process of bias of information, and consequently on the credibility level. However, the procedure to evaluate the credibility of websites by users is impacted only by the number of heuristics they are aware of, biasing the human evaluation with respect to a limited and specific set features.

Additionally, the position in the main page of Search engines is a key factor in the decision making process. The higher ranked a website is when compared to other retrieved websites the more credible people judge a website to be[31]. Popularity is considered as another major credibility parameter[32]. In[33], it is proposed to integrate a recommendation engine into a Web Credibility Evaluation System (WCES). Shas et all[34] have centered their effort in extract features from focusing on the user's feedback. Among the most relevant features, it can be found: 1) the quality of the design of the website and 2) how well

27 Gil Rocha, Henrique Lopes Cardoso, and Jorge Teixeira. 2016. ArgMine: A framework for argumentation mining. Computational Processing of the Portuguese Language-12th International Conference, PROPOR. 13–15.
28 Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. ACM Trans. Internet Technol. (TOIT) 16, 2 (2016)
29 Satoshi Nakamura, Shinji Konishi, Adam Jatowt, Hiroaki hshima, Hiroyuki Kondo, Taro Tezuka, Satoshi Oyama, and Katsumi Tanaka. 2007. Trustworthiness Analysis of Web Search Results. Springer Berlin Heidelberg, Berlin, Heidelberg
30 Brian J Fogg, Cathy Soohoo, David R Danielson, Leslie Marable, Julianne Stanford, and Ellen R Tauber. 2003. How do users evaluate the credibility of web sites?: a study with over 2,500 participants. In Proceedings of the 2003 conference on Designing for user experiences, pages 1–15. ACM.
31 Julia Schwarz and Meredith Morris. 2011. Augmenting web pages and search results to support credibility assessment. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM
32 Katherine Del Giudice. 2010. Crowdsourcing credibility: The impact of audience feedback on web page credibility. In Proceedings of the 73rd ASIS&T Annual Meeting on Navigating Streams in an Information Ecosystem-Volume 47, page 59. American Society for Information Science
33 Xin Liu, Radoslaw Nielek, Paulina Adamska, Adam Wierzbicki, and Karl Aberer. 2015. Towards a highly effective and robust web credibility evaluation system. Decision Support Systems, 79:99–108.
34 Asad Ali Shah, Sri Devi Ravana, Suraya Hamid, and Maizatul Akmar Ismail. 2015. Web credibility assessment: affecting factors and assessment techniques. Information Research, 20(1):20–1

the information is structured. In[35], the trustworthiness of a web source based on the information extracted from the source (i.e., applies fact-checking to infer credibility) is estimated by using a (KBT) method. Triples correctness are extracted to determine the trustworthiness of the source.

35 Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. 2015. Knowledge-based trust: Estimating the trustworthiness of web sources. Proc. VLDB Endow., 8(9):938–949.

2. REQUIREMENTS

Section Summary: this section presents the requirements, firstly from the perspective of an end-user and what they expect to see from this module. Based on this, it presents a set of technical and functional requirements and the outputs that they expect to obtain from it.

From an end-user perspective, the analysis of text can support them to understand inconsistencies in the manner of writing, the semantic incoherence and related indicators of abnormalities. However, an exhaustive analysis must also include high-level descriptors. As an example, not only the text is valuable, but also who wrote, where has the article / news been written, and even more deep analysis on metadata information. This ambition is generally tackled by using network approaches, which consists on express the entities in the news environment as a graph of connected nodes that are joint according to the relation among them. In other words, what network approaches do is to implement the data model into a mathematical formulation that allows to perform analysis to extract relevant patterns and hidden relationships to support journalists and end users in the decision making process.

Therefore, the expected output for end users in this context are values “scoring” the degree of reliability or fakeness of an entity. More information on the data model can be found in deliverable D2.2, however, for the sake of simplicity, let’s summarize it into 9 type of entities: Author, Publisher, Article, Media_image, Media_Video, topic, Entity, Claim and claim_reviewers. To clarify the entities topic and Entity refer to the type of objects found in the news content. As an example, a news over US presidential elections has a topics: elections, Donald Trump, Democratic Party, Republican parties…

Fake news recognition has a particular growing interest as the circulation of misinformation on-line increase. This problem would be faced by building a graph to collect and connect authors and organization in order to apply some graph analysis.

2.1. GOAL

This task is directly linked to WP4.4 and the main goal is to provide a reliability score for the aforementioned entities based on the information available, the hidden and expressed relationships that a particular entity (i.e. author) might have.

This proposal will be faced from diverse perspectives. Initially, traditional graph processing techniques will be employed. As an example in this group is the successful PageRank algorithm. Furthermore, the research will be focused on applying novel techniques based on Deep Learning on graphs.

2.2. PRECONDITION (USER REQUIREMENTS ADDRESSED)

The main challenge to successfully apply the mentioned techniques is the need of having updated labelled datasets. For this ambition, the End users (journalists) will support technical teams in the appropriate classification of data sources.

2.3. CHOSEN APPROACH

The current approach considers the use of the background created with the support of end users to express the graph as a network of:

- publishers
- authors
- articles
- and additional entities according to FANDANGO's data model.

Moreover, a graph is a combination of two main components: nodes (vertices) and edges (links), where in this case, the different elements involved in the data model such as authors publishers or articles are represented as nodes whereas the connections that they have among them are denoted as edges. Subsequently, different centrality algorithms such as PageRank or Eigenvector Centrality algorithms are used to compute the influence of the different nodes in the network based on their connections.

Furthermore, combining the different scores of the rest of the analyzer, a weighted metric is computed to support the credibility of either an author or a publisher. For instance, whether the articles associated to an author have a high probability of being fake and the corresponding images to such article were detected as fake as well, it is clear that the credibility of such author should be low. Moreover, if such author has a large centrality score according to the PageRank algorithm it means that his/her fakeness influence in the network is relevant and therefore, its final credibility score must be lower.

Finally, a sigmoid function is used to keep the range between 0 and 1 since the pageRank algorithm range is above it and then it is multiplied by 100 to attach the meaning of percentage. A similar procedure is investigated as well for publishers in order to manage the computation of their credibility measure.

2.3.1. TECHNOLOGIES INVOLVED IN THE PROCESS

For the initial analysis, the most suitable, scalable and fast technology for graph signal processing is Neo4J, and therefore is chosen as the core technology to create the graph and perform the personalized queries. This technology is integrated with powerful tools for advance processing tasks. An adaption to python programming language is done for some related tools such as Natural Language Processing, in the processing tasks libraries such as *fuzzywuzzy*, *nltk* and *numpy* will be employed to speed up processing tasks, cleaning information and adapting it to the proposed models.

2.3.2. PROCESS OUTPUT

The user will receive as result a set of scores for the related entities including both the publisher of the article and the set of authors associated to it.

An example of the output provided by the analyzer is presented below in a JSON format. Four keys form the final solution including:

a) **identifier**: the unique identifier of the article. This value is provided during the preprocessing step, where the "texthash" is used for this purpose.
b) **author**: this field contains a list of Elasticsearch identifiers associated to the authors.
c) **publisher**: this field contains a list with the corresponding Elasticsearch unique identifier of the publisher.
d) **authorRating**: this field will provide the credibility score of the authors and it is represented as a list of percentages in the range (0, 100), where the higher the value is, the higher the credibility of such author will be.
e) **publisherRating**: this field contains the final credibility score associated to the publisher of the article. It is presented as a list of values in the range (0, 100) as well indicating the trustworthiness of the publisher.

{"**identifier**": "Media plays a very important role in news articles. Images and videos are often used as corroboration to the story in an article. It is very important for the Fandango project to be able to automatically evaluate media and mark them as authentic or manipulated. Additionally, even if a media file is authentic, the system should also check if the specific media file coincides with the information existing in the other modalities of the article. In order to make this evaluation, the Fandango system will provide for a number of different descriptors that will help understand the time frame when the media was captured, the location that is depicted and the objects that are displayed.", "**author**": ["MICT-GoBu09LVEn7SCPg"], "**publisher**": ["MYCT-GoBu09LVEn7SiNn"], "**publisherRating**": [10.25], "**authorRating**": [21.45]}

3. PROBLEM STATEMENT

Section Summary: This section presents the problem from a mathematical formulation. The objective is to introduce the reader in the techniques adopted by FANDANGO so far and its notation is relevant to understand where the results come from.

From a conceptual point of view, we can define a graph as a set of vertices (nodes) and edges (relationships). From a mathematical point of view, a graph *G* consists of a pair of sets *(V,E)* , where *V* refers to the collection of vertices and *E* indicates the set of edges.

Moreover, a data model needs to be mapped into a graph representation. To do so, we first need to define the concept of *labeled Graphs*. More specifically, labelling a Graph *G = (V,E)* means a mapping $\alpha : V \rightarrow A$, where *A* indicates the label set. Moreover, we need to do the same for the set of edges, so that we need to define a mapping $\beta : E \rightarrow B$, being *B* the label set in this case.

As it is described in previous deliverables such as D.3.1, FANDANGO has created a potential data model for representing the fake news detection by ensembling the different entities that are involved in such process including authors, publishers and articles, claims and so on. Furthermore, the set of edges $V = \{v_1, v_2, ... v_n\}$ represents the semantic connection among the entities. Therefore, the different edges are also provided by the designed FANDANGO's data model.

In the first approach that was investigated, the label set *A* is composed by three main elements: *Author*, *Organization* and *Article*. Similarly the label set *B* is composed by the relationships among the entities including: *IS_AFFILIATED_TO, WAS_PUBLISHER_BY, WAS_WRITTEN_BY.* As it is observed, this is a simplification of the general data model described in D.3.1.

Moreover, the investigation of the final trustworthiness score is computed as a combination of different and powerful graph algorithms that have the capability of analyzing the influence and the similarity among the different elements of the network.

To do so, the main idea is to research and detect possible hidden patterns that occurred when publishing fake news in the media market including anonymous authors, unsecured domains, or even the timeline and the registration of the domain or its PageRank. In many scenarios, these features arises as potential data to be used in the analysis. Therefore, using centrality algorithms to measure ranks as well as the degree of power in the network together with data mining techniques such as clustering are proposed to solve the problem.

Along the following sections, the explanation of the different techniques that are employed to get the final scores will be presented. In particular, in Section Source Credibility Scoring we will present a first

approach to model the degree of trustworthiness regarding authors and organizations. Then, in the Source Profiling section a description of the selected methods and approaches to cluster the different media sources will be explained as well. Then, we will present the technology that was considered to solve the problem and the results that were achieved based on this first approximation. Next, the implementation of the service will be described with details with the purpose of facilitating the understanding of the component and its integration in FANDANGO's platform.

Finally, as it was remarked in previous documents and deliverables, the goal of this component is to provide a scoring to measure the trustworthiness of both authors and publishers. By using this set of scores, the end-users and journalists may take advantage of them to make a final decision. Hence, the aim of the artificial intelligence components or services is not to make decisions without the consideration of the end-users but to support them in choosing the most optimal or adequate decision.

2.1. Source Credibility Scoring and Profiling

Due to the exponential growth of the internet, the huge number of fake-news propagation has been increasing every year. Therefore, measuring the credibility of the sources may arise as a potential application to be considered. As it is suggested in [36], many global indicators as the so-called PageRank from Google or the Alexa Rank are not public anymore, so that, generating new trustworthiness indicators for websites arises as a new opportunity to be explored using different modalities and techniques such as visual, text analysis or graph analysis which is the modality that is proposed in this document.

Moreover, there exists many fact-checking tools such as the one presented in[37] , which are naturally divided in two main families: the ones that works over natural language claims such as the research described in this paper[38], and another family of tools that works directly using knowledge-bases including the recent study described by Esteves et al[39].

On the other hand, source profiling consists of analyzing the information available for the different sources and then determine a set of clusters or profiles where similar sources are ensembled in the same cluster[40]. More specifically, profiling the source arises as a potential tool to be used to complement the credibility of a certain source in the sense that:

Having a collection of sources, $S=\{s_1, s_2, \ldots, s_k\}$ associated to a set of labels $Y=\{y_1, y_2, \ldots, y_k\}$ indicating the level of trustworthiness, then, an unknown source $s_{\{k+1\}}$ is matched to a corresponding cluster based on the similarity between the cluster centroid and such source. Therefore, using the profiling procedure, the credibility score is improved without manually label every new source that is introduced in the network.

In the next sections, we are going to explain all the steps that are used to obtain the credibility or trustworthiness score from the mapping of the articles into Neo4j to the computation of the graph

36 "Belittling the Source: Trustworthiness Indicators to Obfuscate Fake ...." 3 Sep. 2018, https://arxiv.org/abs/1809.00494. Accessed 11 Jul. 2019.

37 "Detection and Resolution of Rumours in Social Media: A Survey." 3 Apr. 2017, https://arxiv.org/abs/1704.00656. Accessed 11 Jul. 2019.

38 "Automated Fact Checking: Task formulations, methods and future ...." 20 Jun. 2018, https://arxiv.org/abs/1806.07687. Accessed 11 Jul. 2019.

39 "Toward Veracity Assessment in RDF Knowledge Bases." 15 Mar. 2018, https://dl.acm.org/citation.cfm?id=3177873. Accessed 11 Jul. 2019.

40 Cypher Language. Available: https://neo4j.com/developer/cypher-query-language/

algorithms.

2.1.1 Mapping Fandango Entities into Neo4j

Firstly, the article data needs to be mapped into Neo4j in order to form the graph database and to produce the corresponding scores. To do so, we make use of the native query language that Neo4j employs named as Cypher which allow us to store and retrieve data from the graph database. An example to create an *Article* node is presented below:

```
CREATE (i:Article {uuid:$id, headline:$headline, body:$articleBody,
    images:$images, videos:$videos, dateCreated:$dateCreated, dateModified:$dateModified,
    datePublished:$datePublished, authors:$authors, publisher:$publisher,
    sourceDomain:$sourceDomain, fakeness:$fakeness, language:$language})
```

One remarkable step is to avoid duplicates in the graph database. Since the identifiers of all the entities involved in FANDANGO are unique, we need to indicate to Neo4j that the nodes of the type Article must be unique as well using the identifier as the key to check it. The Cypher query in this case is the following:

```
CREATE CONSTRAINT ON (i:Article) ASSERT (i.uuid) IS UNIQUE
```

Thus, using similar expression for the different types of nodes of our graph database, we avoid suffering for duplicate nodes and relationships. Then, the relationships among the different nodes are computed using the labels of the nodes that are desired to be mapped and selecting the name of the relationship between them as the following Cypher query dictates:

```
MATCH(i:Article) UNWIND i.authors as author MATCH(a:Author) WHERE author =
 a.uuid MERGE (i)-[:WAS_WRITTEN_BY]-(a)
```

By employing the expression above, we are capable of associating the set of authors that wrote the corresponding article using the relation *WAS_WRITTEN_BY*. Once we have applied these queries for the different entities that we want to relate, the graph database is updated with such information. In the figure below, an example of the output of this process is represented.

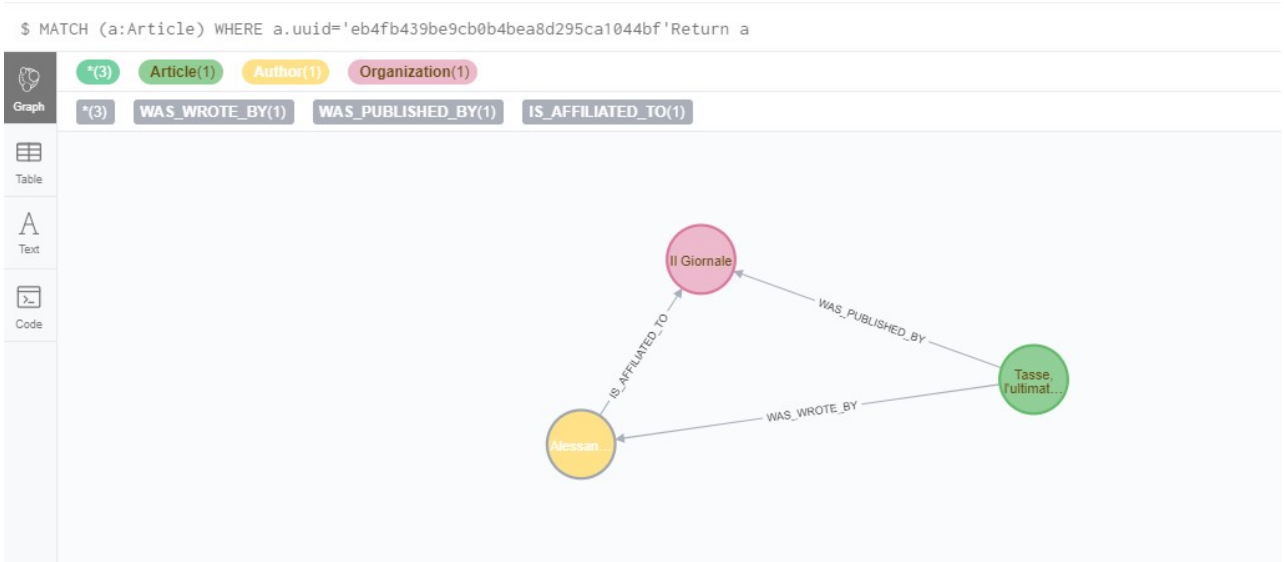


Figure 8. Example of the Neo4j web interface after relating three nodes: Article (green), Organization (red) and Author (Yellow)

2.1.2 Source Credibility Scoring: Graph Analysis

In this section, the description of the process needed to compute the credibility scores using graph analytics is presented. First of all, there are different algorithms and features that are required in the analysis:

1. Centrality Algorithms: This family of algorithms seeks to measure the power of a certain node n_i in a network *G* based on the set of connections or vertices *V* that are associated to it.
2. Clustering Algorithms: This family of algorithms are in charge of computing a distance measurement that indicates the similarity among the different elements of *G*.
3. Control parameters and features: The set of features will contribute to determine the level of trustworthiness or credibility in a positive or negative way depending on the control parameter. For example: if a certain publisher has anonymous authors associated to the articles, the feature is the name of the author and the control parameter associated to such feature will be negative since it is anonymous.

The final score $\mu \in [0,100]$where a *value of close to zero indicates a low level of trustworthiness*, whereas a value close to 100 will indicate the opposite. Moreover, the initial value is set up to 50 which indicates that there is not enough information to determine the trustworthiness.

As it was introduced before, the credibility score is a combination of different algorithms and procedures. Firstly, two powerful centrality algorithms are used to measure the impact of a certain node in the Network: ArticleRank[41] and Eigenvector[42]. Therefore, we will denote the centrality partial score with the Greek letter. Thus, this partial score consists of a weighted sum of the aforementioned algorithms:

$$\varphi = \omega_1 C_1 + \omega_2 C_2$$

Where the weights ω_1, ω_2,associated to the centrality measurements vary over time according to the number of elements available in the network. Moreover, an additional constraint is imposed to prevent noise in the final solution so that, the sum of both must be equal to one. In addition, both metrics are normalized every time in order to keep the value of in the proper range.

Considering this partial score, we are capable of measuring the impact of the node in *G*. However, this is not enough to measure the level of trustworthiness since a non-trusted node may have a similar value of than a trusted one. Therefore, we need to find additional information to capture the credibility in a more adequate way. To do so, we need to define some additional features that we will denote with the Greek letter $Y = \{1, 2, \ldots, m\}$. In addition the control parameter set will be denoted as $\Lambda = \{1, 2, \ldots, m\}$. By employing this set of features, we can improve the investigation of the credibility.

According to previous investigations, there exists many meta-data information associated to the websites domains that are useful to determine the trustworthiness of a certain source including:

1. Anonymous authors associated to the piece of news.
2. The country where the domain is registered
3. The created date of the source

41 Gong, C., Li, W., & Yi, P. (2019). Rank-based analysis method to determine OWA weights and its application in group decision making. International Journal of Intelligent Systems, 34(7), 1685-1699.

42 Agryzkov, T., Tortosa, L., Vicent, J. F., & Wilson, R. (2019). A centrality measure for urban networks based on the eigenvector centrality concept. Environment and Planning B: Urban Analytics and City Science, 46(4), 668-689

4. The domain type of the website (.edu, .org, .com etc.)
5. The amount of articles posted in the network.

As expected some of the information is not available a priori since it is not captured by Crawlers such as the created date or the country of the domain. To mitigate this problem, we have employed WhoisIP[43] , a powerful tool that retrieves meta-data given a domain or a url. Moreover it provides a Python library which was used to collect the information.

Thus, after calculating , the algorithm to obtain the final credibility score for a certain node nilabelled as Organization is the following:

1. Compute the difference between the amount of anonymous and non-anonymous authors and obtain the value of 1 as:

   where Pand R are the sets of non-anonymous and anonymous authors respectively.

$$\gamma_1 = \lambda_1 \sqrt{N} \left(\sum_{i=1}^{P} a_i - \sum_{j=1}^{R} a_j \right) s.t. P + R = N$$

2. Retrieve the country of the Publisher using WhoisIP. If the country is available λ_2 will be positive, if not it will be negative. Finally $\gamma_2 = a\,\lambda_2$

2. Retrieve the created date of the source using WhoisIP as well. In this case, the total number of years that the source has been active is used as the third feature:

$$\gamma_3 = \lambda_3 \sqrt{T}$$

   Where *T* indicates the number of years that the source has been providing news. The hypothesis behind this feature is that a new source domain cannot be trusted as much as one that has more years of experience.

4. Then, we take the suffix of the source domain (.org, .edu) and we compute its value as:

$$\gamma_4 = \lambda_4 b$$

   Where *b* is the associated weight to the corresponding suffix. For instance, .edu must have a parameter *b* greater than .com since the former is associated to academic purposes.

5. Using the training label that indicates the trustworthiness of the source, we compute 5 as the square root of the total number of articles. If the label indicates that the source is non-trusted, we assume that all the articles associated to it are non-trusted and then, such value must be negative and vice versa. So that $\gamma_5 = \lambda_5 \sqrt{\{N\}}$, where *N* refers to the total number of articles written by this source.

6. Once all the features are computed, we need to put everything together in order to retrieve the score for a particular source:

$$\mu = 100\, sigmoid$$

   Where a sigmoid function is used to maintain the range in the interval between 0 and 1. Finally, the output of the sigmoid function is multiplied by 100 to keep the format of the trustworthiness score.

In the case of computing for authors the number of features is reduced. However, some of the features are directly inherited from the source domain were they have published. Therefore, to compute

43 Available online: https://www.ultratools.com/tools/ipWhoisLookup

the credibility score of a certain author we follow the next steps:

1. Check the name: if the name is anonymous $\gamma_1 = \lambda_1$, otherwise the value is the same but negative.
2. Compute the average mean of the scores of the articles that such author has written and multiply it by the total number of articles that he/she has associated:

$$\gamma_3 = \lambda_3 \sqrt{N} \left(\sum_{i=1}^{P} a_i - \sum_{j=1}^{R} a_j \right) s.t. P + R = N$$

Where in this a_i and a_j are articles and *P* and *R* are the set of fake and non-fake articles. As expected *N* refers to the total number of articles written by a certain author.

3. The last feature γ_4 consists of the trustworthiness score of the source where the author has written the article.

3. The final score for a certain author is computed in a similar way as the one obtained for a publisher, so that:

$$\mu = 100\ sigmoid$$

Furthermore, we have observed that all the features depend on a set of control parameters denoted as λ_i. The influence of these values is really important, the higher the value is, the sooner the score will converge to its bounds. On the other hand, if these values are very small, the scores will require a large number of iterations to converge to either 0 or 100. During this period, several experiments were carried out with different values and it was defined that $0 < \lambda_i < 0.5 \; \forall \; i \in R$. Some of these experiments are depicted in the following figures:

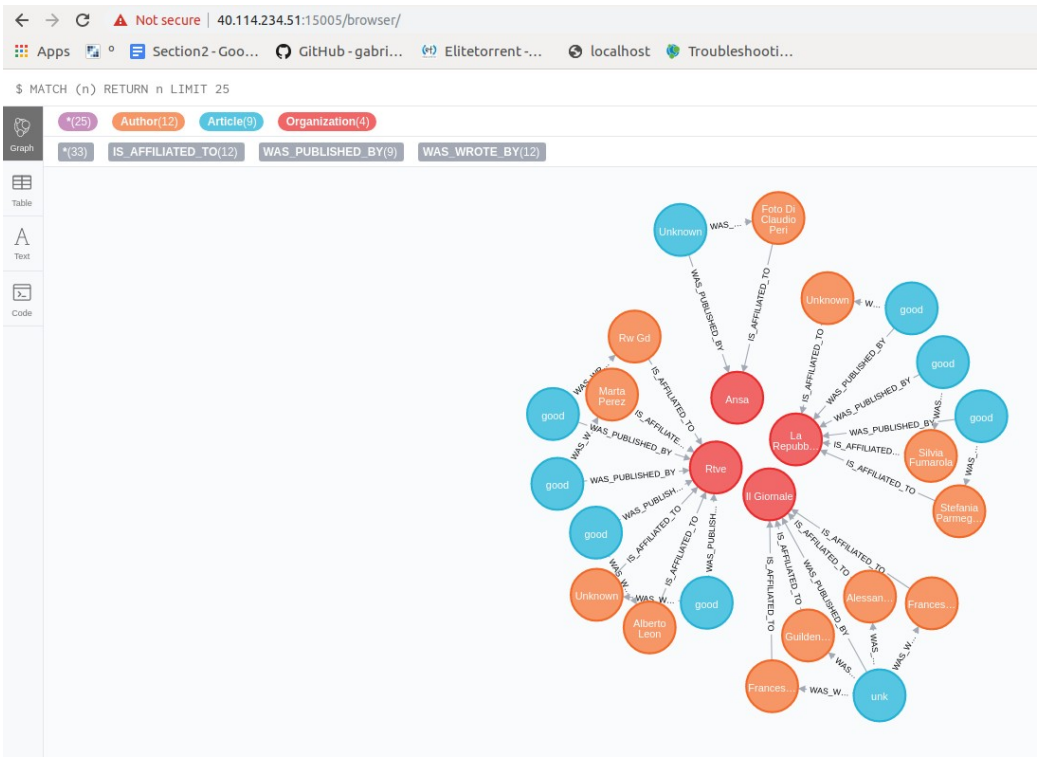


Figure 9 General overview of Neo4j Dashboard with the experiments performed for 4 main data sources (red circles), including RTVE (www.rtve.es), Il Giornale (www.ilgiornale.it), La Repubblica (www.larepubblica.it) and ANSA (www.ansa.it). Orange circles indicate the author and blue circles represent the scoring respectively.

This procedure is very useful to implement having a set of labels associated to the organizations since the scores can be evaluated to measure the quality of the model. However, whether an unseen source or author is introduced into the network, not always is going to have this meta-data and therefore, not always is going to be easy computing. To mitigate this problem, a clustering procedure is employed in order to measure the similarity between the new node and the rest of the nodes of the same type associated. Once the similarity is computed, the unseen node automatically receives an average score of the similar nodes. In this first approach we have employed the so-called Jaccard similarity metric which is defined as the size of the intersection divided by the size of the union of two sets.

Thus, If the unseen node is an Organization, is computed as the average of the scores of the k-organizations more similar to it. On the other hand, if the unseen node is an Author, is calculated as the average of the scores of the authors associated to the same source.

In future releases, the introduction of Graph Convolutional Neural Networks to promote the investigation of the credibility score will make the process more accurate and robust against errors.

2.3. Social Graph Analytics

The scope of this task is to analyze the influence of social networks in spreading fake content through the Internet. There are different methodologies and procedures that can be useful in order to detect user/accounts that are potential candidates to be bots or to spread non-trusted information or biased propaganda:

1. Clustering users or accounts using different features from the profiles that can be collected directly from the social network.
2. Analyze the sentiment of the users based on a topic such as climate change, migration or the European budget.
3. Analyze the bigrams, n-grams associated to the tweets from a particular topic.

There exists different social networks that can be analyzed to the aforementioned purposes such as Facebook, Twitter, Instagram or Reddit. Facebook arises as the best candidate since is one of the most popular social networks that is employed not only by people but also by companies and media-content providers. However, its API is not open so that, it is not possible to access to the data and the user information. Consequently, it was decided to analyze Twitter, which contains an open but limited API. More specifically, the API is bounded to a number of calls in windows of fifteen minutes, which makes complex the gathering process of tweets. On the other hand, Twitter provides an enterprise solution for commercial and business-oriented developments.

Moreover, in the last few years, with the exponential growth of social networks and media content, many researchers in both social media and data science fields have analyzed the spread of fake news and propaganda via Twitter in order to understand if there exists relevant hidden patterns in the accounts that provides misinformation. The fundamental drawback lies in the time consuming that monitoring the activity of hundreds of thousands of twitter accounts implies. In addition, the set of bots are creating social networks with the capability of generating automatically posts that may reach to a large audience by doing retweets among them. Then, a certain user may see this feed in their network and he/she might be deceived into believing false facts and claims.

Therefore, Machine/ Deep learning techniques play a crucial role in order to automatically detect fraud spread of news and also anomalies in the social network. As it is suggested in different articles[44][45] , there exists relevant information that may be considered in the analyzed and that can be

44 Singh, N., Sharma, T., Thakral, A., & Choudhury, T. (2018, June). Detection of fake profile in online social networks using machine learning. In 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE) (pp. 231-234). IEEE

collected directly from the profile of the account such as the created date, the number of followers or the description stored on each user profile of users.

Along this section, a full-description of the first approaches that have been followed to analyze Twitter is presented including the methodology that was employed and the first results that were obtained during the experimental phase.

Furthermore, the first step that was required to do this research is collecting a potential dataset of tweets in some domains where FANDANGO is focusing on including climate change, migration and the European budget. In particular in this first approach, the investigation was centered upon the climate change topic but it will be generalized to the rest of the domains.

2.3.1. User Account Clustering

The scope of this section is to present the methodology that has been used to perform the clustering using a set of users from Twitter. As it was mentioned before, we have focused the experiments in analyzing the climate change domain, therefore, in order to collect specific tweets from this topic, we have used different *hashtags* that are related to it such as *#ClimateChange, #climateaction* or #globalwarming. These tags are provided by RiteTag, which is a website that informs about different trend topics in Twitter.

In the following image, a block diagram regarding the main steps that are considered in this analysis is presented. As it is depicted in the figure, after collecting the data, a feature extraction stage is performed to compute relevant features that are representative for the clustering analysis. Then, the clustering analysis is applied. In particular we used a powerful clustering algorithm named as Spectral clustering. The description of the different steps will be described later on.

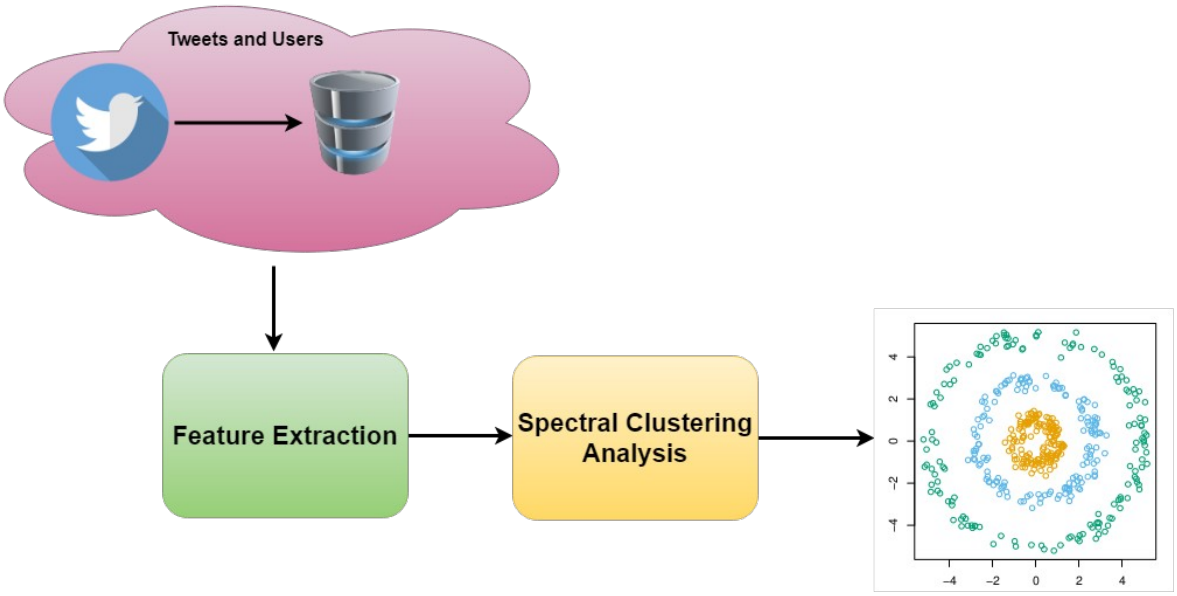


Figure 10 Block Diagram representing the main steps that are followed to compute the clustering analysis of twitter accounts

Once the data is collected, the set of features that may have relevance in the clustering is defined. In particular we have considered the following features regarding the users:

45 Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. (2014, August). Using sentiment to detect bots on twitter: Are humans more opinionated than bots?. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 620-627). IEEE Press.

1. Text description of the profile
2. Number of followers of the account
3. Number of people followed by the account

Moreover, we have analyzed as well some characteristics of the tweets of these users:

1. Amount of favorites
2. Amount of retweets
3. Boolean indicating if the tweets contains url's
4. Boolean indicating if the tweet is a response
5. Boolean indicating if the tweet contains hashtags
6. Boolean indicating if the tweet is a retweet

Many of the aforementioned features can be computed directly from either Tweet object or the User object. However, in the case of the description, an additional feature extraction stage is required to extract features from this short piece of text.

Thus, to compute the feature extraction regarding text, we have used the so-called Term-Frequency times inverse document-frequency denoted as *Tf-IDF*[46] . Basically, this descriptor can be analyzed as a product of two fundamental statistics, term frequency and inverse document frequency.

Furthermore, the feature extraction step is crucial to obtain a potential feature vector, denoted as $x^T \in R^d$. In this case, the dimensionality of the feature vector is high and the range among the different features is different, so that, a normalization is also mandatory in order to avoid noisy results in the clustering performance.

After computing and normalizing the feature vector, the clustering analysis is applied. As it was mentioned before, the spectral clustering algorithm which is widely used in clustering scenarios[47][48][49]. More specifically, the scope of cluster data that is connected but not necessarily compact or clustered within convex boundaries. One of the main advantages of this procedure is its capability of clustering together data that does not follow Gaussian shapes. The main steps that are required to perform the spectral clustering is the following:

1. Project the whole data into R^n
2. Define an Affinity matrix denoted as *A*, which consists of a similarity matrix among the different points. More specifically, it uses an Adjacency matrix (i.e. $A_{ij} = i, j$).
3. Then, we need to build the Graph Laplacian from matrix *A*.
4. The next step consists of solving the *Eigenvalue* problem defined as: $Lv = \lambda v$ and selecting the set of eigenvectors $\{v_i, i = 1, \ldots, k\}$ in order to obtain a k-dimensional subspace.
5. Finally, define clusters in such subspace using standard clustering methods such as K-means.

In the following figure, we present the results that were obtained after performing the clustering analysis. In order to be feasible to visualize in 2 dimensions, we had to perform and additional step which is the TSNE algorithm in order to project the result of the clustering in a 2-D subspace where the data points are well-separated. As it can be observed in the figure below, using this clustering procedure the system is capable of separated properly the use accounts according to the aforementioned features.

46 Python implementation available: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76

47 Li, C., Bai, J., Wenjun, Z., & Xihao, Y. (2019). Community detection using hierarchical clustering based on edge-weighted similarity in cloud environment. Information Processing & Management, 56(1), 91-109

48 Wang, Y., Wu, L., Lin, X., & Gao, J. (2018). Multiview spectral clustering via structured low-rank matrix factorization. IEEE transactions on neural networks and learning systems, 29(10), 4833-4843

49 Trillos, N. G., & Slepčev, D. (2018). A variational approach to the consistency of spectral clustering. Applied and Computational Harmonic Analysis, 45(2), 239-281

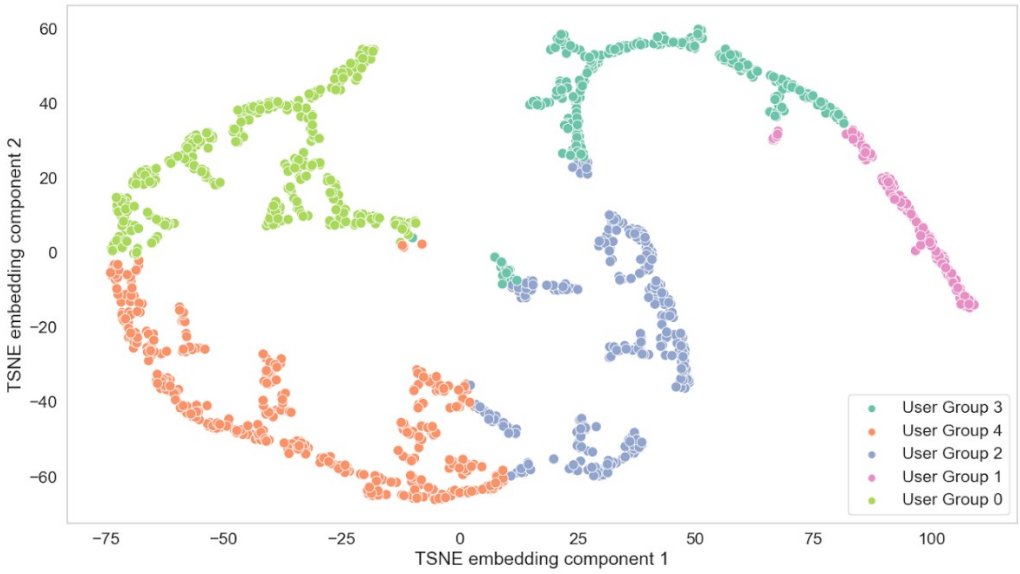


Figure 11. Spectral clustering representation using the so-called TSNE algorithm

The main drawback of using clustering procedures lies in the lack of labels to better evaluate the performance of the system. However, in the case of Twitter, the large amount of user accounts makes a supervised procedure very inefficient since it would require a large collection of manually tagging accounts labels which would imply a high amount of time consuming. Hence, using unsupervised techniques such as Spectral clustering is the best approach to be followed in this scenario.

Moreover, the number of clusters is not a trivial task and requires many experiments to obtain the optimal value. After the experimental wave, five clusters or user account categories were obtained. In the following table, we present different statistics associated to each of the aforementioned user account groups.

Feature Name	*Min*	*Median*	*Mean*	*Max*
Retweet Count	0	26	33	430
Followers	0	141	326	1861
Following	2	689	1344	5002
Favorite Count	0	0	0.12	8

Table 6. User account group 0 statistics

Feature Name	*Min*	*Median*	*Mean*	*Max*
Retweet Count	0	5	16	60
Followers	38	2905	77954	5392060
Following	0	497	1079	10646
Favorite Count	0	0	5	87

Table 7. User account group 1 statistics

Feature Name	*Min*	*Median*	*Mean*	*Max*
Retweet Count	0	8	23	430
Followers	46	1574	4793	101150
Following	40	1608	4707	102791
Favorite Count	0	0	0.32	14

Table 8. User account group 2 statistics

Feature Name	*Min*	*Median*	*Mean*	*Max*
Retweet Count	0	8	22	430
Followers	5	1577	4330	122925
Following	0	823	2192	55517
Favourite Count	0	0	1.32	63

Table 9. User account group 3 statistics

Feature Name	*Min*	*Median*	*Mean*	*Max*
Retweet count	0	11	24	431
Followers	0	641	1007	4039
Following	0	1123	1700	6992
Favorite count	0	0	0.50	109

Table 2. User account group 4 statistics

Moreover, in the next figure we present the relationship between the number of followers and the number of followings according to each group. As one may observe, it is clear the correlation among the user accounts based on this pair of features. The size of the data points refers to the amount of retweets and the color indicates the cluster or the group of each user account.

Finally, it is clear that using clustering analysis, we can potentiate the investigation of twitter user accounts that may spread fake news all over the network by analyzing the characteristics of them in comparison with standard user profiles due to, in many cases, the bots are created automatically in a similar way, so that, similar description, similar amount of followers etc.

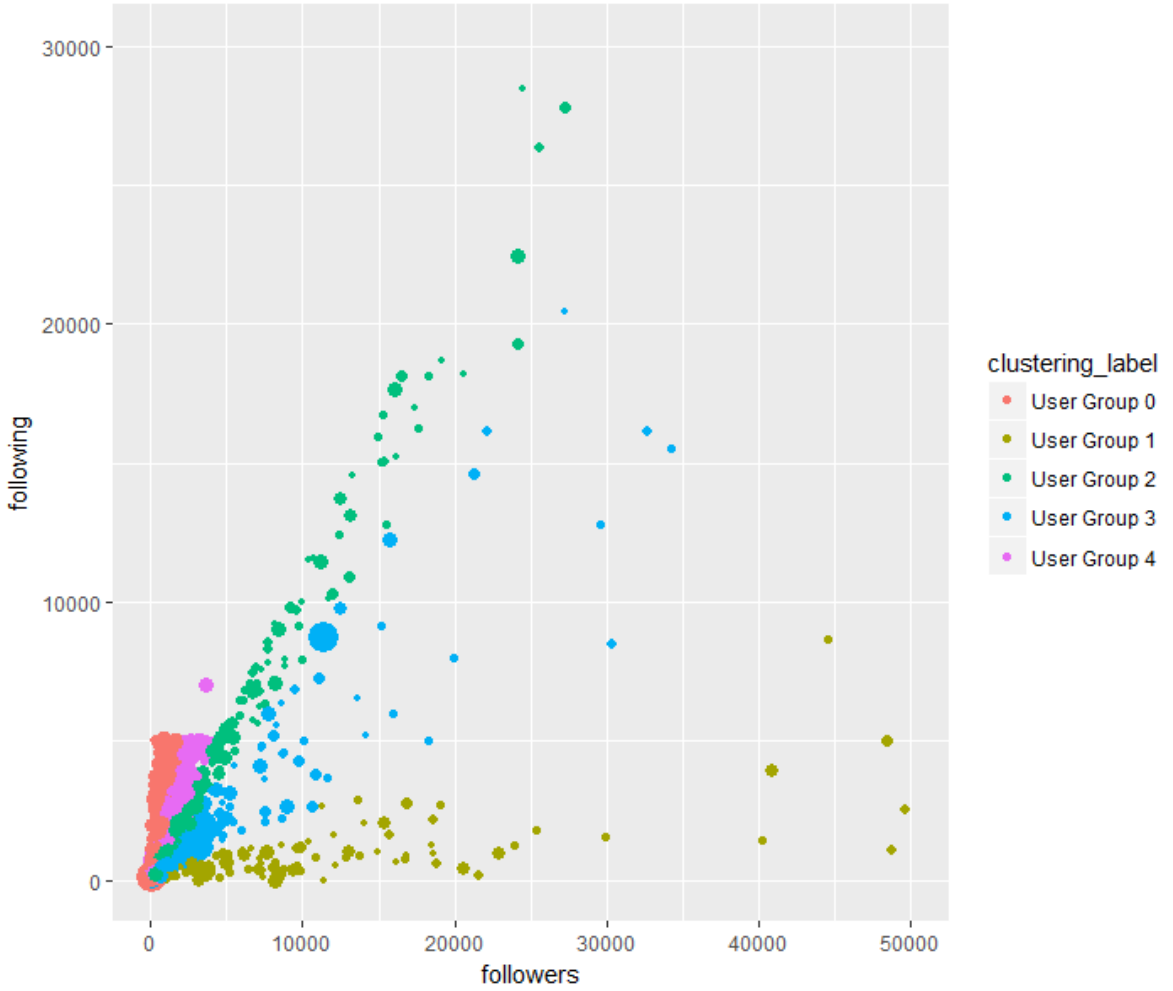


Figure 12. Relation between the number of accounts following with respect to the followers for the user account groups identified.

2.3.2. Tweet Location Visualization

Another different approach can be the use of the location of the users all over the world in order to see in which countries and cities, people are more involved in a certain topic such as the European elections climate change or global warming.

The main drawback in this scenario lies in the lack of information that many tweets have regarding location where the user account is sending tweets due to it is not mandatory in Twitter to provide the location. However, we have collected many tweets where the location is available in order to do the analysis.

Furthermore, we have used a large scale dataset which was sued in the FiveThirtyEight Story[50] which was collected by Clemson University researchers. The dataset contains around 3million tweets which were sent from a Russian troll factory to distribute propaganda during the U.S.A. presidential elections. The first paper written by the researches from Clemson University is based on this dataset[51]. In this case, most of the tweets were spread in the U.S.A. as it is expected since the target was to influence in the decision of the people regarding the presidential elections. However, this analysis can be very meaningful to many other scenarios and situations.

50 https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/

51 Boatwright, B. C., Linvill, D. L., & Warren, P. L. (2018). Troll factories: The internet research agency and state-sponsored agenda building. Resource Centre on Media Freedom in Europe

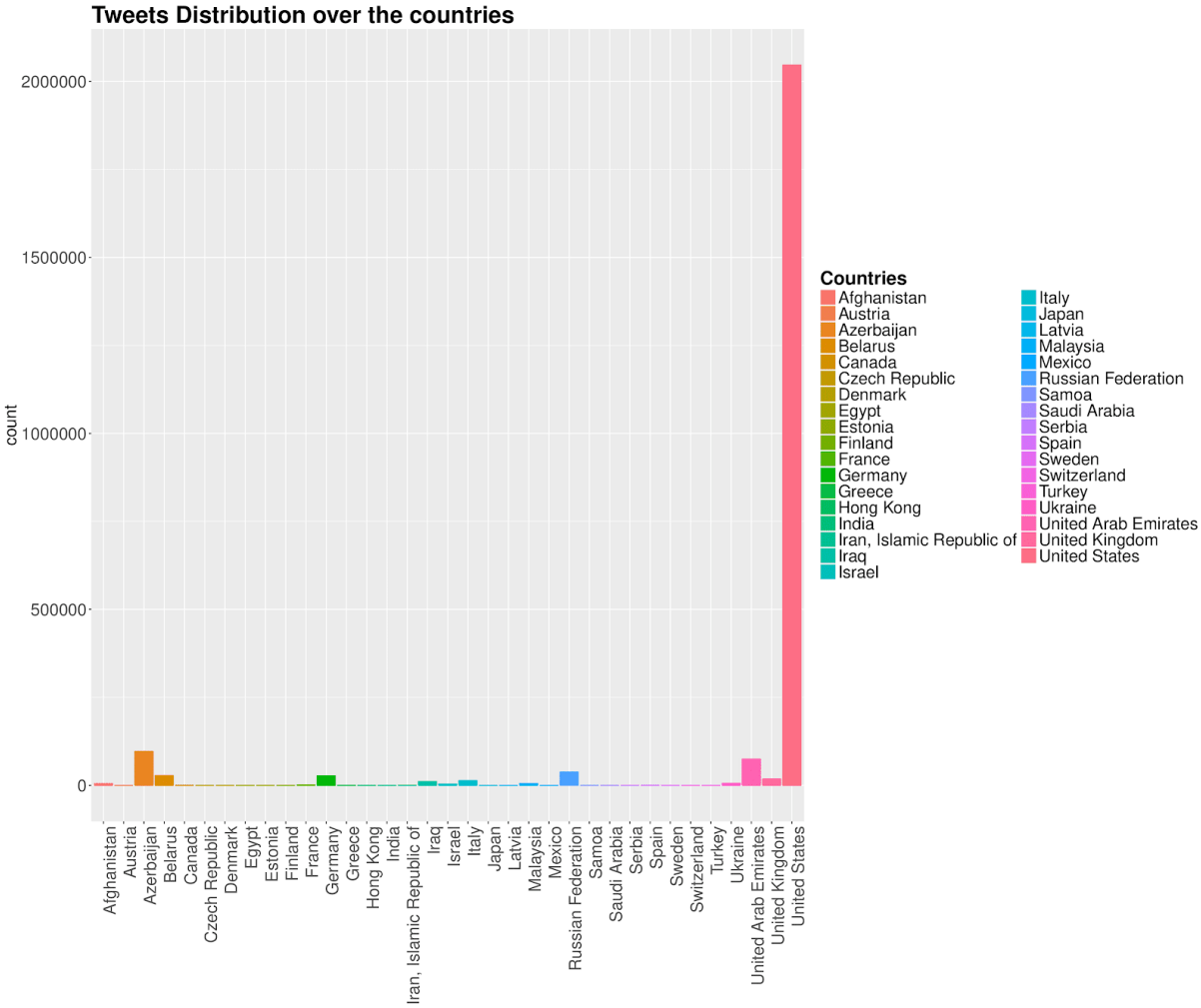


Figure 13. Representation of the amount of tweets generated in each country in the Russian Trolls 3M dataset.

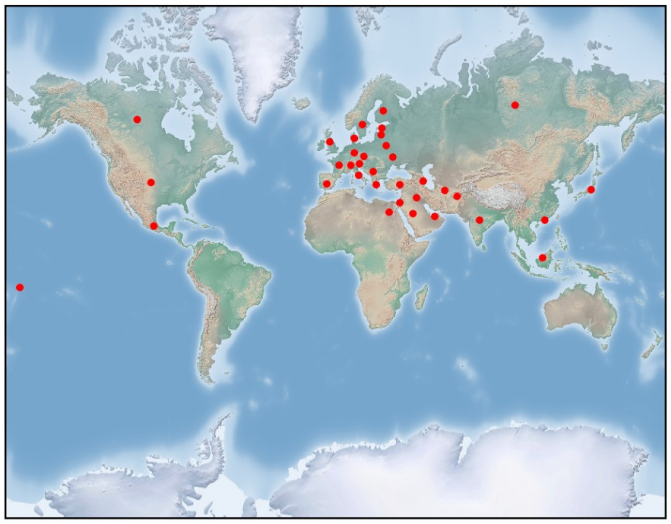


Figure 14. World Map with the red points where the tweets of the Russian Troll Factory were detected

Therefore, it is clear that once a user account is detected as a candidate to be a fake news broadcaster, using a global visualization of the countries where these news are reached arises as a potential application. Hence, the visualization of the events in an intuitive way such as a map will help journalists and other media-content analyzer to understand in which countries the influence has reached faster and with more power.

2.3.3. SENTIMENT ANALYSIS

Sentiment analysis is a very widely application where the main goal is to predict or detect the opinion of people according to a certain topic. Therefore, we will present a potential solution to analyze the perspective of people in a certain country or area based on the information that is stored on each tweet. In the figure below, a block diagram of the sentiment analysis is represented. The main difference is the implementation of a model to detect the sentiment of tweets after the text feature extraction step.

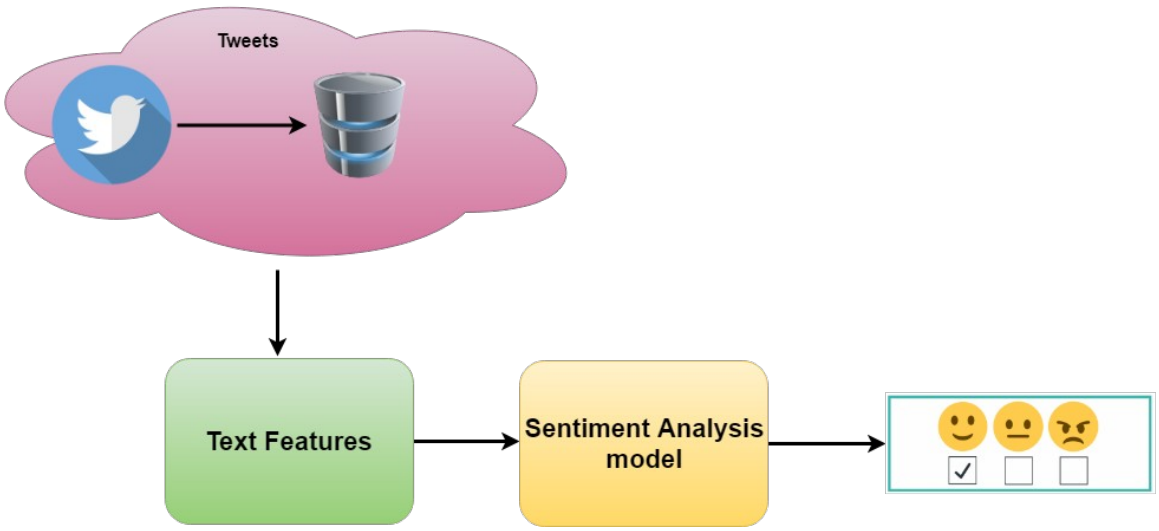


Figure 15. General framework of the procedure for Sentiment analysis of tweets.

The procedure is a little bit different than in the clustering analysis scenario. In this case, once the data is collected from a certain topic, which in this case is climate change, a deep learning model is employed to predict the sentiment or the opinion of tweets. As it is expected, this problem is naturally a supervised problem where the model is trained using the labels of the tweets in order to learn the patterns of the sentiment. More specifically, the output of the model consists of a score in the range (-1,1) indicating the negativity or positivity of the tweet along the scale. Thus, a tweet whose sentiment output is close to one will indicate that the tweet supports or it is positive according to the topic, where a value close to -1 will indicate the opposite.

This application is very well used in different scenarios and in the case of social networks analysis, a sentiment analysis can be correlated with the amount of fake news spread in order to see whether the influence of the fake news is affecting a certain topic in a specific period of time. Hence, we have implemented a sentiment analysis module that can be used for different topics in order to inform in a more adequate way about the current status of the perspective of people in different scenarios.

In the following figures, we present some of the results that were obtained after the sentiment analysis. Firstly, a figure representing the set of n-grams more common in this topic is presented. The basic idea of this representation is to provide the co-occurrence of words in a 2-D space. On the other hand, the next figure presents a histogram with the distribution of the tweets regarding the score of the model. As it can be observed, most of the tweets were neutral since their scores are around 0. Moreover, many other tweets contained a positive support of climate change in this particular period. The data collected corresponds to tweets from last November 2018 until now. As expected, this distribution will change over time according to external factors including the spread of fake news or biased propaganda.

Furthermore, as it is described in the paper[52] , using sentiment analysis can be used to detect bots on Twitter since most of the tweets spread by bots do not contain too many sentiments in comparison with human tweets.

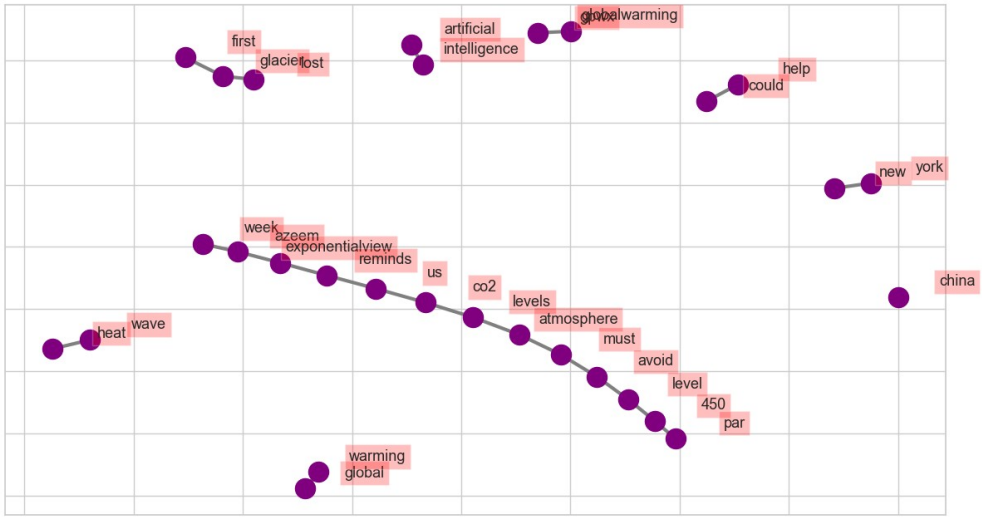


Figure 16. Representations of the n-grams according to the textual analysis of a large amount of tweets that contained some climate change hashtags

52 Dickerson, J. P., Kagan, V., & Subrahmanian, V. S. (2014, August). Using sentiment to detect bots on twitter: Are humans more opinionated than bots?. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (pp. 620-627). IEEE Press.

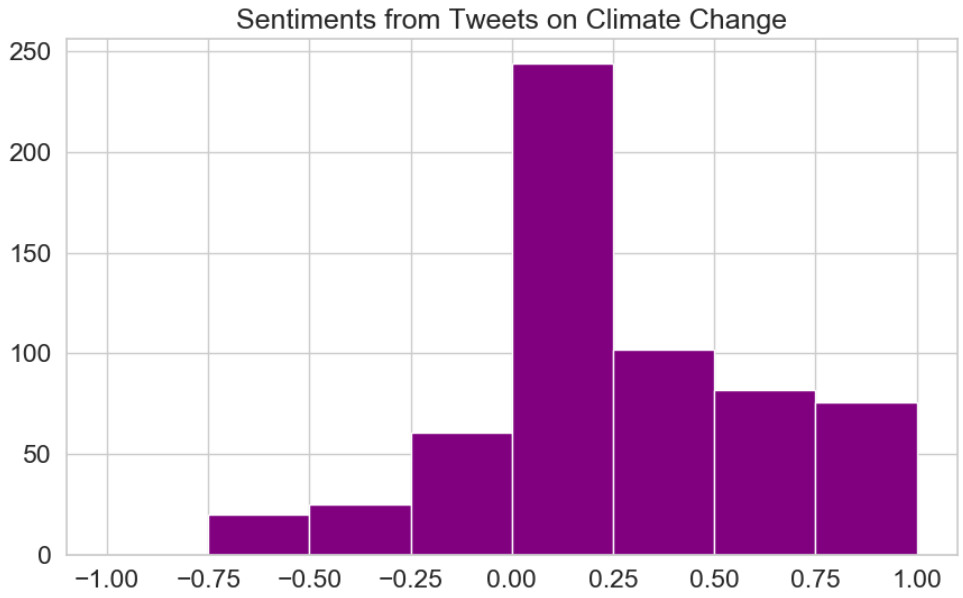


Figure 17. Representation of the distribution of the sentiment score regarding the climate change

4. Service Implementation

Section Summary: This section provides the technical details on the implementation of the source credibility score service deployment. It provides the logic of the service functionalities, as well as the technical components and how these interact to retrieve results to the end users. Finally, the results obtained are illustrated.

In the previous section, we have described all the services that have been integrated or will be in a future release. In this section, the main goal is to present a description of the service is explained.

3.1. Source Credibility and Profiling Service

g

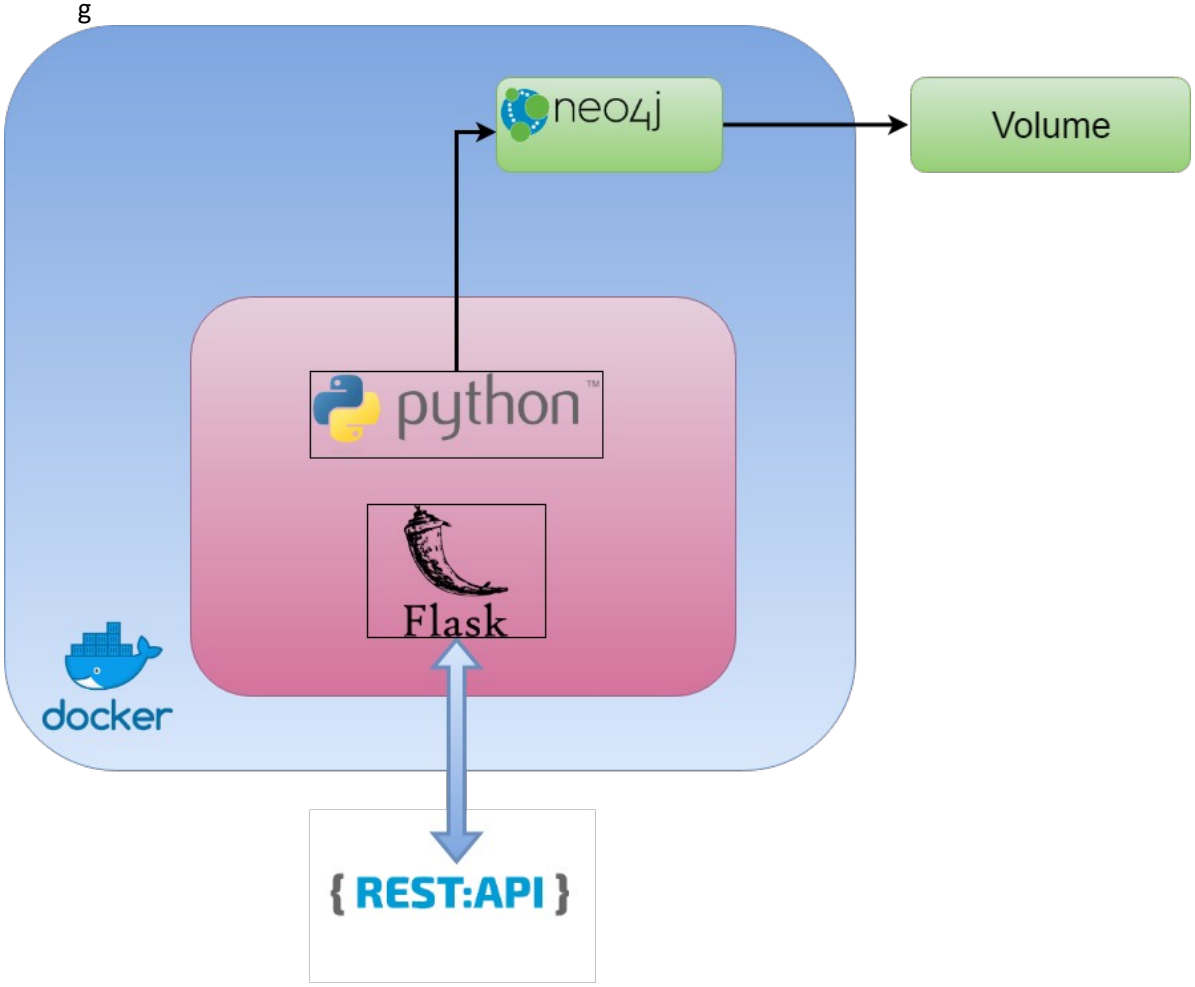


Figure 18. General interaction of the technologies involved in the FANDANGO Source Credibility Service.

3.1.1. Technologies Involved

In order to perform the source credibility score, NEO4J was used as the graph framework to perform all the graph algorithms and operations. More specifically, a Python driver named as Py2neo[53] was used to

53 Python library. Available: https://py2neo.org/v4/

develop the service in pure Python. Then, the powerful Python library Flask[54] was used as the back-end of the service to provide HTTP requests between the different components that are implemented in FANDANGO. To see more information about the services, please refer to D2.4.

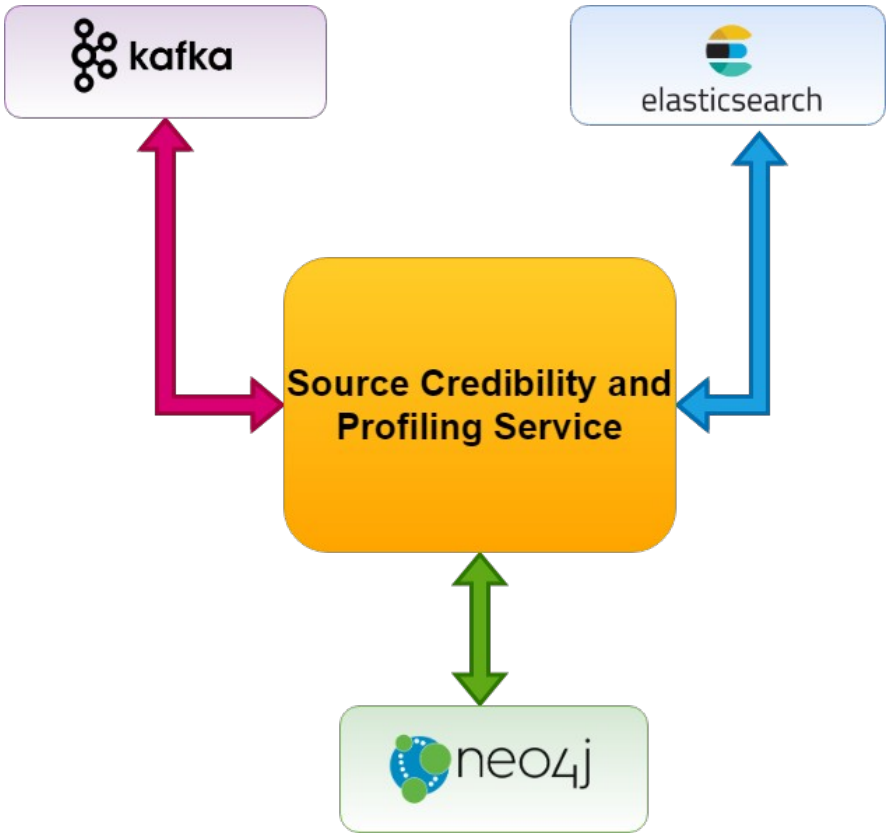
Furthermore, in order to attain a scalable and replicable integration of the service, a docker-compose implementation is employed which makes the service more robust to be implemented in any machine with a minimum amount of effort and resources.

The process was performed as follows:

- Initially, partners create individual docker container with the technical requirements. As a result, a Dockerfile document containing the docker commands to construct the image were provided.
- Once this Dockerfile was functional, then a docker-compose (YAML) document was created to automate the services.
- This document will be orchestrated by using Kubernetes virtualization manager[55].

3.1.2. Service Specification

In this section, a description of the process and the pipeline of this service will be presented. The purpose of this section is to clarify how the service works in terms of communication with other components.



54 Python library. Available: https://palletsprojects.com/p/flask/
55 Kubernetes orchestrator: Avaialble at: https://kubernetes.io/

Figure 19. Source credibility and profiling service components. Bi-directional arrows indicate that the service reads and writes from/to the rest of components

In the figure above, we present the different components that are involved in the source credibility service including: Kafka, Elasticsearch and Neo4j. For more information regarding these components, please go to D2.4. Moreover, the service is implemented using two main threads: the first thread will read the data from kafka and will put it into Elasticsearch, whereas the second thread will map the data into NEO4J and it will update the scores in Elasticsearch after finishing the graph analysis process. To summarize the process, the following steps are required:

1. The service starts reading from Kafka queue *input_preprocessed*. This process continues until the kafka queue is empty.
2. It takes all the authors and the publisher involved in the article document and search them in Elasticsearch. From now on, each author and/or publisher is denoted as entity to make easier the explanation. Moreover, we will denote the *score* as the trustworthiness indicator that is computed using the graph algorithms.
    1. For each entity, the service search it on Elasticsearch in the corresponding indexes: fdg-person, fdg-organization for both authors and publishers respectively. If the entity exists such index then, the service retrieves it; otherwise it generates a new entry in the index using the name and the url to compute a hash identifier in the case of the publisher and just the name in the case of the author. The length of the identifiers is 128.

b. Then, it stacks the information from Elasticsearch of all the entities in a Neo4j queue. In addition, identifiers and non-updated scores of the entities are ingested into the Kafka queue *analyzed_auth_org*. The format of this output can be found in D2.4.

c. The thread associated with the Neo4j queue reads the data and map it into the graph database. Then, it computes the algorithm to compute the scores of each involved entity. Finally the scores are updated into Elasticsearch in the aforementioned indexes. The process continues until the Neo4j queue is empty.

By following the above pipeline, the service is capable of keeping the real-time requirement to reach the aggregation stage that is going to join the different scores from the corresponding services. More information about the aggregation stage can be found in D2.4.

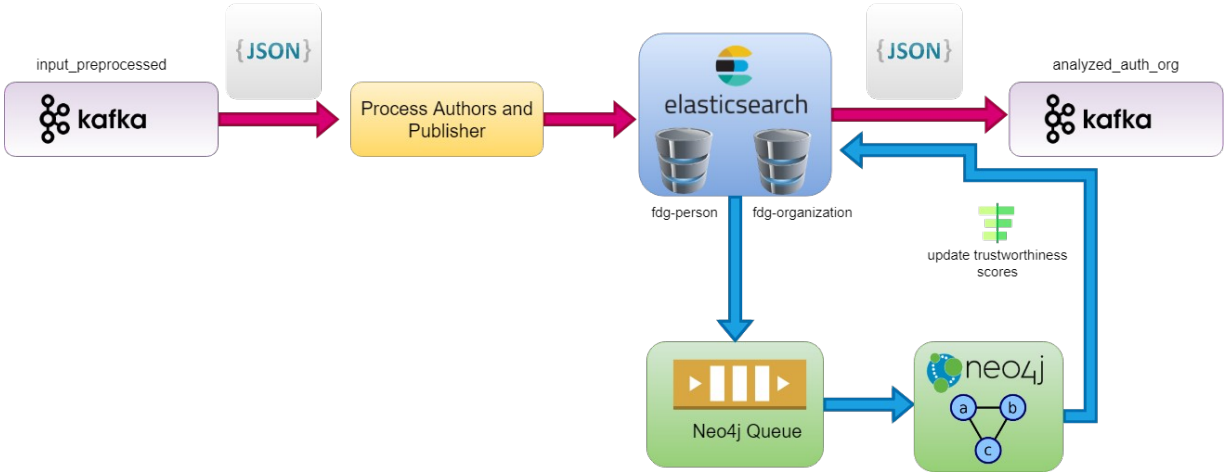


Figure 20. Source Credibility and Profiling service pipeline. In purple arrows, the main thread workflow is presented. On the other hand, in blue arrows we indicate the workflow of the neo4j queue including the update of the different scores.

In the above figure, we present the main two workflows that are followed in this service. More specifically, in purple we denote the main thread workflow where the data flows from a Kafka queue to Elasticsearch and after retrieving/creating the entities and getting the non-updated scores, it ingests the data into a new Kafka queue. On the other hand we denote blue arrows to remark the parallel process that taking as input the Elasticsearch entities, it maps into the neo4j graph database and finally compute the new scores that are updated into Elasticsearch.

3.2. Source Graph Analytics Service

In previous section a full-description of the social analysis that was performed using Twitter was presented. In particular three different analysis were carried out in order to investigate potential solutions to the problem of fake news distribution in this particular social network.

As a result, these services can be integrated together in order to be presented in a more adequate way to journalists and media agencies that can use this set of services to improve the quality of their research.

3.2.1. Technologies Involved

Firstly, the social graph analytics that was performed in Twitter, a Python library was also required: Tweepy. Using this library we are capable of accessing all the end-points of the API provided by Twitter to interested developers. In a future implementation, the set of services will be integrated as an additional service for being explored in the principal FANDANGO interface.

Finally, Neo4j will be used as well in order to visualize the network of a certain tweet or account by mapping the Twitter data model into the neo4j database. Recall that this new graph database will be different than the one related to articles that is used in the source credibility implementation since they are different services.

3.2.2. Service Specification

This service will be available in a future release using the same architecture as the one described in the source credibility and profiling service.

Regarding the specification of the service, both inputs and outputs will be analyzed together with end-users in order to offer them the best solution in terms of visualization in order to promote the investigation of the propagation of fake news throughout the Twitter network.

3.2.3. Experimental Results

- Fandango supports its modules/services (described above) in 4 different languages (English, Italian, Spanish and Dutch).

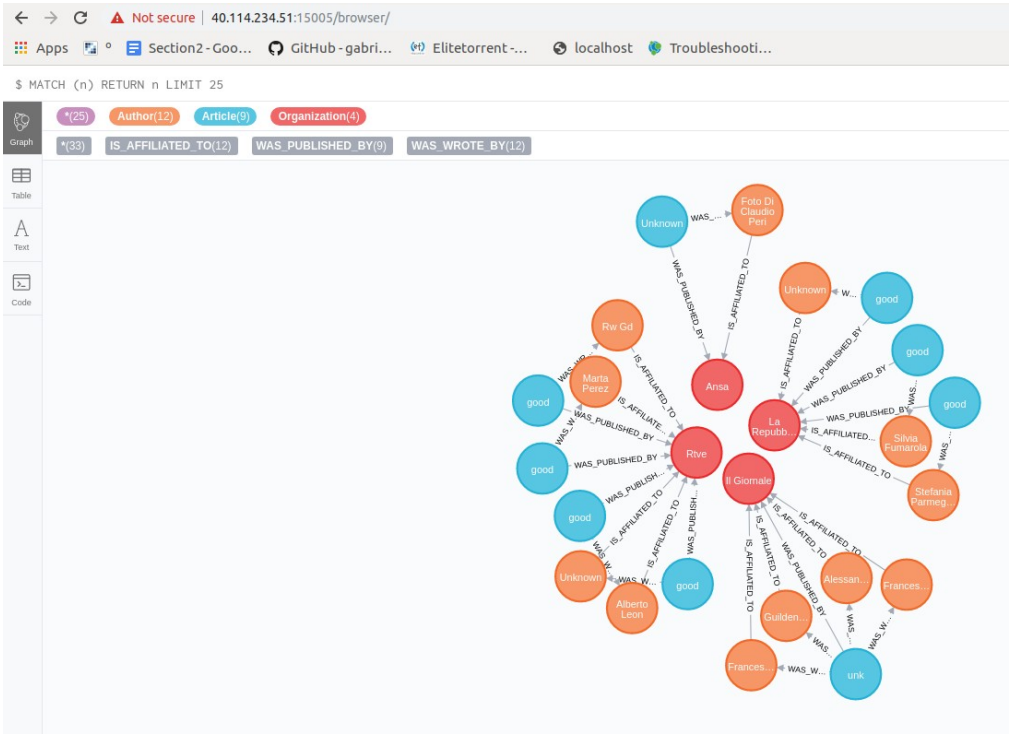


Figure 21 General overview of the project NEo4j Dashboard for some articles ingested for some particular news publisher of interest.

In the figure above, it might be observed the general overview dashboard of the relationship among the nodes. As an example, entities *publisher* and *author* are linked by the relationship *IS_AFFILIATED_TO.* Furthermore, the entity *Article* and *Publisher* are related by the link *WAS_PUBLISEHD_BY.* Finally, the nodes Author and article are related by the link *WAS_WRITTEN_BY.* For fast visualization, Neo4J is generally limited to 25 nodes.

Also, results can be exported in JSON format according to the data model requirements:


```
{
        "trustworthiness": "87.41",
        "gender": "",
        "nationality": "",
        "affiliation": [
        "b5b65d82accd1f6772cd0d7ea4d829f6c8066bba3e317227489dcc98cd61ccd14f25822ecce1c638a7eb17f6064947083409dab1b5e01d3e02346b76c9421f54"
 ],
        "bias": "",
        "jobTitle": "",
        "name": "Guildenstern Vo",
        "uuid":
"db34ae0efb199a5c902842d4f9756eb4cdb11396b19b4431ef31487ecdc8488322851ad918a38f45e0a82a02bc82adbf3e5c2a967277273164ef351e1bc88de9"
```

}

By the date of this report (Mid July), ENG as platform owner provided an IP address to check the status of this service. The status of the graph can be consulted here:

10.114.234.51:15005/browser

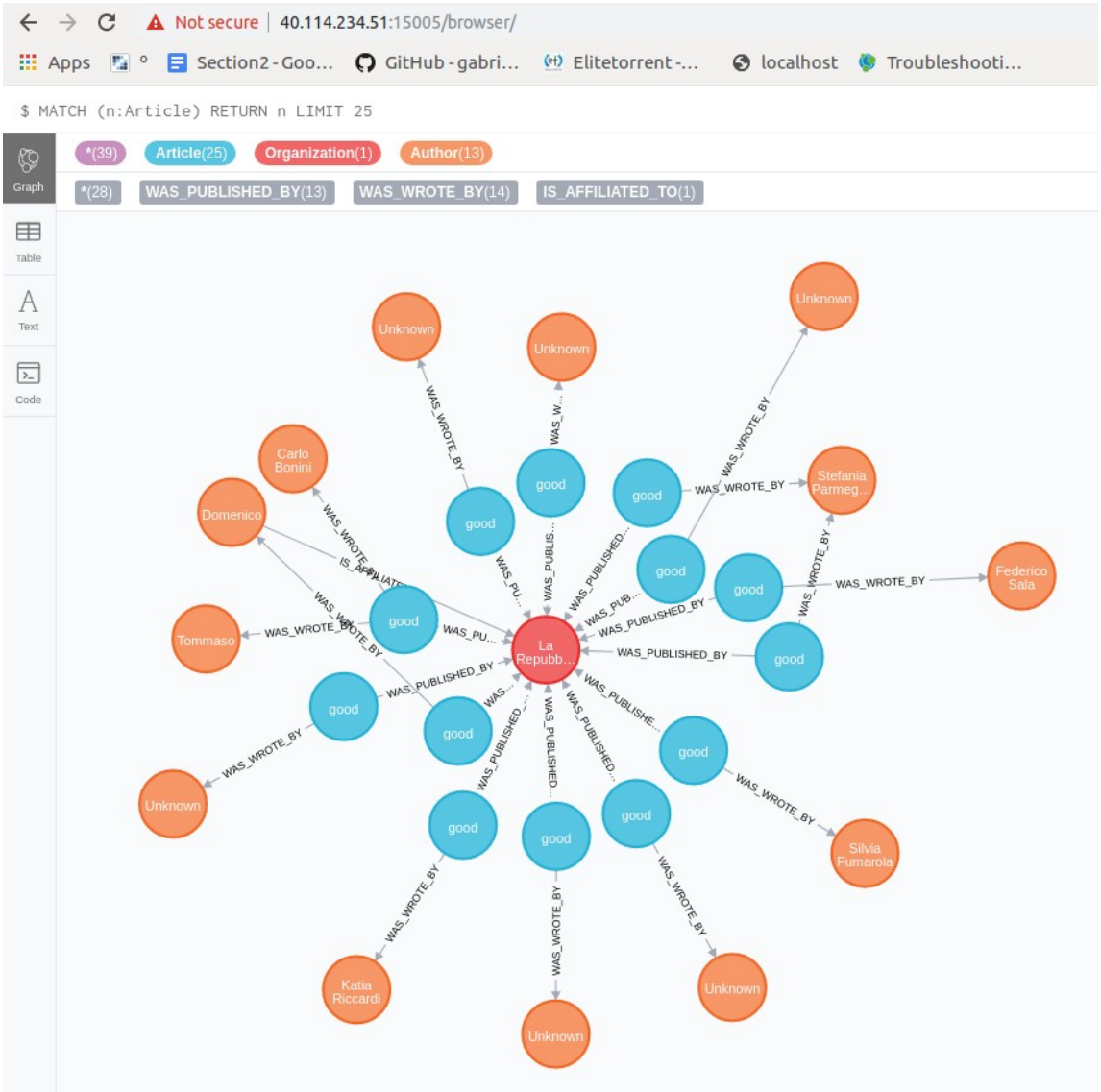


Figure 22. Particular evaluation for a "trust" news agency (www.larepubblica.it). The results of the algorithm (blue circles) contain the label of the evaluation. The orange circles contain the name of the author that publishes the related articles.

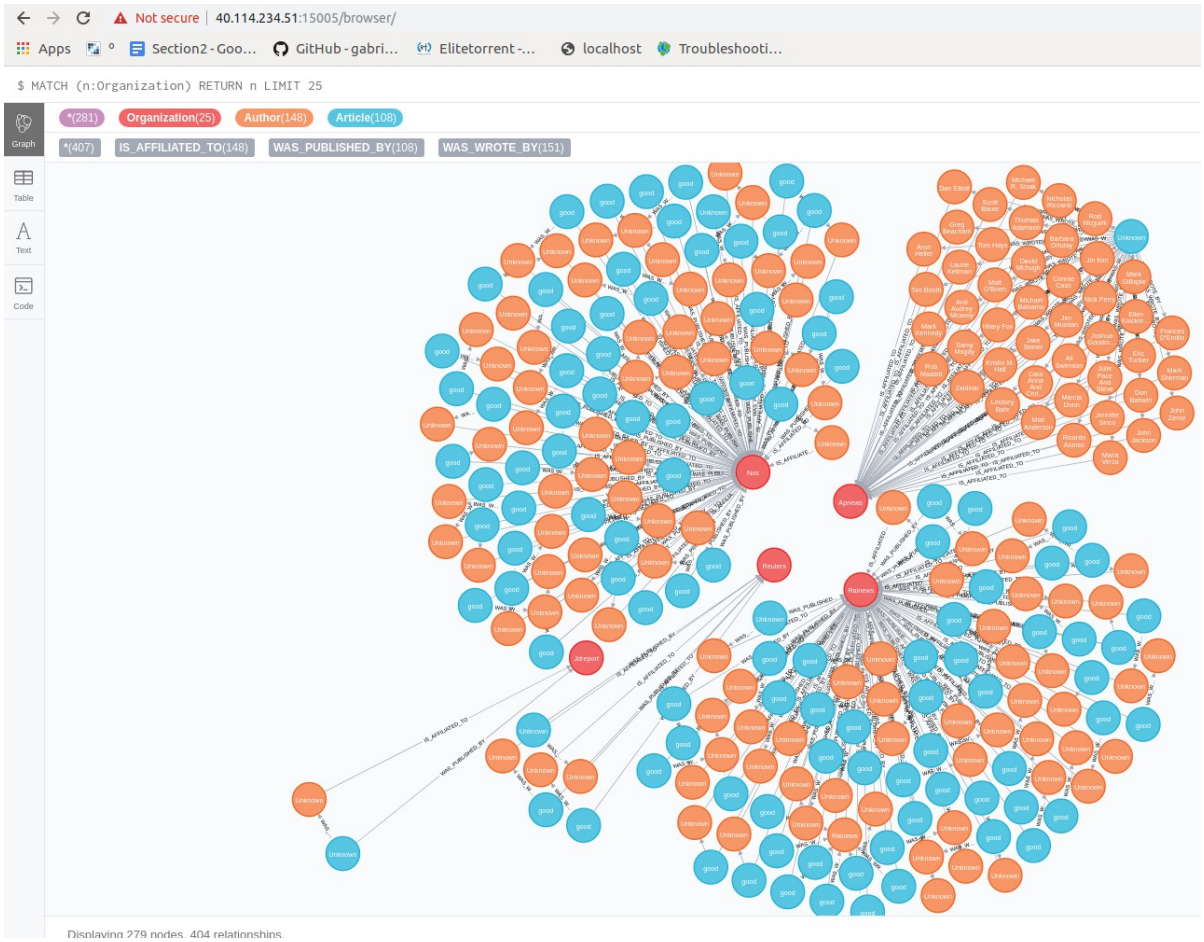


Figure 23. General Tree expansion for the articles collected for some days for some particular sources. Similarly, Blue and orange circles correspond to ratings and authors respectively.

By the time of this deliverable, the neo4j interface contains information from 39117 unique nodes (including authors, organizations and articles). Additionally, there is a large number of relationships among these nodes.

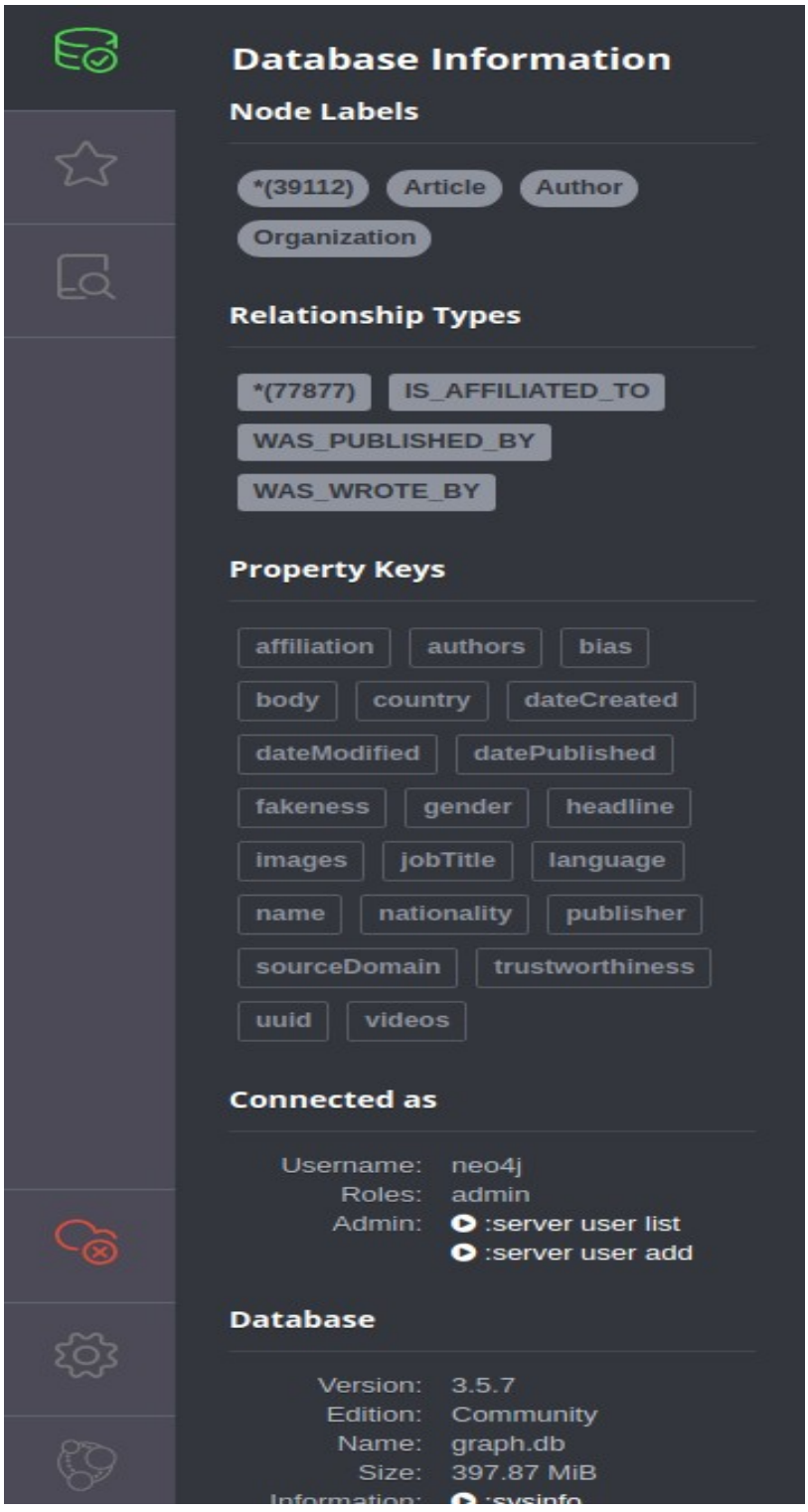


Figure 24. Database contents by July 2019.

All code and general guidelines to allow replicability of the solutions proposed by FANDANGO were released under open access (MIT License). Therefore, scientific community can access the tools developed by FANDANGO for graph analysis. We presented a dockerized solution to implement Neo4J which, so far is not available in its most up-to-date version to community. For the seek of replicability,

scalability and fast deployment of FANDANGO tools, the code has been released by UPM in a open GITLab[56] account and can be accessed and downloaded from here:

https://gitlab.com/davidGATV/fandango-source-credibility.git

Furthermore, the images created, as well as the docker-compose files to run the virtual (self-contained) application can be downloaded from here:

docker push tavitto16/fandango_source:tagname

By the date of this report, the latest tag is the 1.0.5. However, the most updated versions can be accessed from the dockerhub organization account *fandangorg*

56 Available online: https://about.gitlab.com/

GENERAL CONCLUSIONS AND FUTURE LINES

This deliverable reported the work done so far with respect to source credibility and media analytics in the FANDANGO context. In this report, initially a complete review of the state-of-the-art was performed in order to highlight the current trends in the research community to face with the *fact checking, source credibility scoring and news (media) back trace propagation.* As a matter of fact, It is important to stress that section 2 reviewed the literature of these topics from multiple perspectives. Firstly, novel approaches based on Deep Learning methods were reviewed, and specially, the *Graph Convolutional Networks* were stated and described. There are two open issues regarding these techniques: *(a)* The amount of data required to have unbiased estimations and *(b)* The computational cost of both *training* (background offline process) and *test* (on-line process that provides estimations by using the training models). The former problem is inherent to all Deep Learning based approaches, and the manner to face it is to continuously re-training the models, and updating the databases. The latter is faced by increasing the resources available in terms of memory access and specially GPU capabilities.

As this report is directly linked to Task T4.4, this is a continuous evolving task, and consequently, the strategy adopted allowed FANDANGO partners to implement a functional solution for source credibility scoring based on networking approaches as described in section 4 of this document. These algorithms work with data labelled by end-user partners in order to classify the reliability (trustworthiness) of an ingested news (or website) based on classical estimation metrics. These approaches are currently extended and showed some significant results, however, trends (State-of-the-Art) show us that improvement in the performance of this complex task must be coped from the perspective of advanced Neural Networks.

Furthermore, in section 3 of this document, a description of the implementation process and the algorithms applied is provided. The technology for the graph database is Neo4J and the implementation details are presented. Results of the data collected so far are shown. The main goal of this task is not to "make a decision" to journalists, but provide some insights to allow them to have more knowledge and therefore more consistent decisions on the reliability of a news. The system is described with the technologies employed and the integration into the FANDANGO platform.

Additionally, in this document, the media analysis using Twitter social Network was presented. It was tackled from two main perspectives: (a) Content based analysis and (b) Propagation (networking) analysis. The former was subsplitted in several Research topics. The ambition of FANDANGO consortium is to develop fusion techniques to combine the results obtained from each of the modules outlined to enrich the results presented to the end users. The content-based analysis relies on Natural Language Processing techniques to provide results on the multi- media content of the tweets. The latter is focused on providing tools to professionals suppporting them in their decision-making, by creating smart traceability of the contents in this social network.

However, there is a strong commitment of FANDANGO dissemination and coordination partners to foster the pressure to the main social network to open its API for Research purposes that represent add-value to community out of personal / commercial interests.

Lessons Learned

The literature review showed that there are multiple manners of facing with the problems tackled by FANDANGO. However, most of these approaches rely on supervised techniques. Supervised techniques have several problems, but the main problem is the complicated task of *"news annotation".* As seen, within WP4 activities, there is considered a tool for annotation. This tool will be used to label data related to the use cases identified by FANDANGO. However, in the viability of the platform after the project ends, the support of community to update the contents, by annotating trained facts, will be crucial.

There is a clear interest on the fact checking task, however, literature shows that it must be analyzed from multiple perspectives. In a single modality (regardless of image, text), it is very difficult to reach a large accuracy in misinformation detection. The approach presented here provides insights from the metadata graph, and social media analysis. However, significant improvements can be attained in the proper fusion of the outcomes provided for each of the FANDANGO modules.

The performance of the Deep learning approaches heavily depends on the amount and quality of training data. As FANDANGO is evolving, now the amount of data is becoming significant to create prediction models that can overcome biasedness of few data. The efforts of FANDANGO consortia have been focused on solving the data integrity and data quality ingestion.

As part of the research activities in this WP, an article has been published:

> *G. Palaiopanos, P. Stalidis, N. Vretos, T. Semertzidis, P. Daras, 'Embedding Big Data in Graph Convolutional Networks', 25th International Conference on Engineering, Technology and Innovation (ICE/IEEE ITMC 2019), Sophia Antipolis Innovation Park, France, 17-19 June 2019.*

Future Lines

Finally, in the future lines, a roadmap for the integration of the Research modules is provided. In the upcoming months, social media will be incorporated to FANDANGO platform. The system will be prepared to integrate solutions involving (hopefully) the Facebook API access.

The work performed so far on media analysis is still not fully integrated into the FANDANGO platform. By the end of the project, the aggregation task will also consider the media analysis in the weighting