Conspiracy or Not? A Deep Learning Approach to Spot It on Twitter

BORJA ARROYO GALENDE, GUSTAVO HERNÁNDEZ-PEÑALOZA, (Member, IEEE), SILVIA URIBE, AND FEDERICO ÁLVAREZ GARCÍA, (Member, IEEE)
Visual Telecommunications Application Group, Universidad Politécnica de Madrid, 28040 Madrid, Spain
Corresponding author: Borja Arroyo Galende (bag@gatv.ssr.upm.es)

This work was supported by the H2020 European Project (https://fandango-project.eu/) under Grant 780355.

ABSTRACT
Sentiment analysis is an active topic in Natural Language Processing (NLP). It has attracted a significant interest of research community due to the wide range of applications, including social-media, fake news spotting and interactive applications. In this paper, we present a novel approach for semi-automatic background creation and conspiracy classification. For this purpose, a complete framework including novel recurrent models is proposed. The BORJIS: Best algorithm foR Joint conspiracy and sarcasm detection has been tested on twitter-crawled data and It is composed by: (a) the crawler and labelling module, (b) the features vector extraction and (c) the conspiracy classifier. BORJIS was compared with up-to-date techniques and it showed a significant improvement (≥ 10% accuracy) when applied to diverse datasets.

INDEX TERMS
Fake news detection, natural language processing, sentiment analysis, deep learning.

I. INTRODUCTION
The topic of spotting fake news is critical to multiple entities and has become a worldwide issue, particularly referring to Political and Social Challenges that get reflected in business and industry [1]. As an example, the European Commission considers it as a paramount subject to defend its democratic and ethical values [2].

Nowadays, the post truth is a concept broadly accepted for the analysis of news.\(^1\) Furthermore, A survey conducted by the Pew Research Centre towards the end of last year found that 64% of social networks users said made-up news stories were causing confusion about the basic facts of current issues and events.\(^2\)

The creation and spreading of misinformation is quickly evolving and involves multiple approaches [6]. These involve diverse types such as rumors [7], [8], clickbait [9], [10], conspiracy and sarcasm among others [11], [12]. There are multiple aspects to be considered to face with these issues as depicted in figure 1, involving the source, reputation of the author, and related information.

There is a large effort from the scientific community to detect manipulations in image/video and sound/voice/speech modalities [13]. These are continuously redefined due to the high impact that Artificial Neural Networks (i.e. Generative Adversarial Networks) have created (i.e. via deep fakes). Nonetheless, these techniques are useful to verify individual assets, but are not enough to classify the contents from an overall overview.

Moreover, several approaches for the semantic / pragmatic analysis of data in text analysis have been proposed. Similarly, even a deep analysis of the text might not be enough to classify the contents as misinformation.

Consequently, the manner to spot fake news shall be dynamic and it must tackle the aforementioned modalities. Therefore, the inferences must be extracted from diverse perspectives, based not only on contents and its manipulation, but also the context. This is an essential step due to the disparity between writers and readers caused by different mindsets, cultures, etc, and the ambiguity of the language.

The relevance of the topic is given by the strong impact on the society, and it must be addressed by all involved actors such as:
• Media Companies (Press agencies, news, TV Broadcaster etc)
• Governmental institutions and organisations
• The overall Industrial ecosystem
• The entire society

Moreover, there is a substantial interest for the continuous analysis of social media as a fast and up-to-date source of
information. In fact, there are studies showing polarization and opinion shaping on social media [3]. Especially, in social media, there are multiple factors that can support the decision making about the data posted. In particular, sentiment analysis can be applied to discriminate the information in such a complex task.

In this paper, a novel approach to automatically label data and classify it into conspiracy is presented. This algorithm aims to reach it by: (a) using metrics concerning context, such as popularity and polarity among others; (b) proposing a Deep Learning architecture to classify the information and (c) weighting several metrics to take into consideration context information while improving the quality of the results.

A. WHY CONSPIRACY?
Nowadays, conspiracy is critical as proven during the COVID 19 pandemic, where even reluctance to vaccination has been boosted. There are a few novel approaches [14] that, however, are not related to the field of social media.

The rest of this paper is organised as follows: Firstly, the section II studies the State-of-the-art and provides the main contributions. In section III, the data model is presented with a special emphasis on text data and metrics extraction. Section IV describes the overall architecture and technical modules. Lastly, section V shows the results and establishes a discussion around them.

II. STATE-OF-THE-ART AND CONTRIBUTIONS
As mentioned above, the topic of spotting fake news has become an important research area during the last few years due to its impact on the society. In this context, and with the aim of fighting against the spreading of misinformation, different approaches for helping in the information verification process have appeared.

Fact-checking, as the task of monitoring the accuracy of news, can be considered one of the most important exponents in this fight. At the moment, this is mainly done manually [15] since the automation of the entire process is not yet feasible and human supervision is still necessary [16], but current efforts are especially focused on providing different automated solutions for helping in specific parts of the process. In this regard, stance detection, which is in charge of determining the position of one piece of text to another one [17], is one of these parts where different AI models have been extensively applied. The use of recurrent neural networks in this area is quite intensive ([18], [19]), but most of these solutions are based on the use of truncated text inputs due to efficiency issues, with the consequent loss of information. This problem has been faced by using different approaches as in [20], where authors apply a two-stage stance detection process based on a simple information retrieval process that is able to use full articles, providing a a high accuracy following a real-life process of fact-checking.

The detection of conspiracy, as part of the fact-checking process, is also vital for the misinformation analysis and it can easily benefit from applying AI models. Different solutions have been suggested for specific scenarios such as the detection within headlines [21], [22], but this environment is not as complex as the social media one, where the communication has not a uniform structure and it is noisy in terms of labels and language.

In the AI context, the use of machine learning techniques has confirmed a high accuracy when deciding whether a text is sarcastic or not, even with simple methods like SVM ([26], [27]) or binary logistic regression [28]. Following the evolution of the AI scenario, current approaches are more focused on the application of deep learning techniques. In [29] authors use NN for sentiment and sarcasm classification, obtaining a very high performance (F-score close to 90) and in [30] and [31] the use of CNN provides a better accuracy than previous solutions based on machine learning approaches, confirming the improvement of the deep learning architecture when facing this issue.

Contrary to this case, the provision of tools for automated detection of conspiracy is not as widespread since it represents a more complex area. In fact, current research works are more focused on identifying different aspects such as its spread over the network even from a social point of view [32] or on defining specific pipelines for its discovery as in [33]. Nevertheless, some latest researches have emerged recently due to the COVID-19 crisis with the aim of allowing conspiracy automated identification, such as the one proposed in [34] which uses a machine learning approach.
A. MAIN CONTRIBUTIONS

In this paper, a multistage framework for conspiracy in Twitter social media is presented. Up to the best of our knowledge, this framework is the first approach to spot conspiracy. The main contributions of this work are:

- To provide a reduced scheme to extract the most relevant features for conspiracy detection from the Twitter V2 API.
- To create and release an open dataset for applied research on the mentioned topics.
- To propose a semi-automatic crawling and labelling tool using diverse metrics.
- To propose a Deep Learning approach (BORJIS from now on) able to learn from the features to exploit text patterns in the creation/posting of sarcasm/conspiracy, being able to process variable length entries such as Twitter conversations.
- To develop a set of metrics to enrich the context of fake news detection and sentiment analysis in Twitter.

III. TWITTER DATA MODEL AND DATASET CREATION

Nowadays, Twitter has become a reference for information, due to its high credibility and wide use at a worldwide level [4]. There exist some public datasets intended to train algorithms for sarcasm detection.\(^3\) However, up to the best of our knowledge, there is not a specific dataset created to spot conspiracy. Twitter has reached a safe manner to boost research and data access via its public API [5]. The current version (V2) releases a set of novel features that are of interest for the NLP area in general, and conspiracy detection specifically.\(^4\) Twitter API V2 is used within the scope of this article.

In order to have statistically significant information to train the algorithms, the dataset is collected using not only individual tweets but the whole conversation about the topics of interest. Indeed, a tweet count threshold is established at one thousand, ensuring that there is enough information (i.e., words) accessible for the algorithm. This approach allows for a better understanding of the different uses of vocabulary and speech that several users may have.

The features collected to create our data set are described in Table 1. These parameters involve general information and each tweet’s text separated by “.”. Furthermore, there is a set of features provided by Twitter that are also gathered for the research, and these are outlined in Table 2. The features are related to popularity as the number of followers, activity referred to the number of retweets, replies in time, account creation, etc.

Furthermore, as part of the preprocessing stage, several language processing rules are applied to the text of each tweet. The most important ones are the following:

- **Hashtag extraction.** This step allows for real-time queries to the Twitter API with the objective of retrieving information about any hashtag included in each tweet. For the sake of example, a further engine could retrieve records based on a frequency hashtag ordering.
- **Emoji extraction.** Emojis are an important source of information that permit the study of sentiment analysis of textual information. Its adoption by Twitter users is large and demonstrates non-textual information.
- **Mention deletion.** In order to clean up the conversation for a textual processing engine, meaningless words need to be omitted as it could introduce noisy inputs. Into this category falls the mention case that targets another user for responding, referencing, etc purposes. The data should only retain the useful information that can be fed to the algorithm. Of course, by having this type of proceeding, we are disconnecting all relations that occur within the social network, but graph analysis is out of the scope of this work.

Apart from the preprocessing stages, the current version implements a deterministic conspiracy tagger. This tagger consists of a set of deterministic rules that can be applied to the Twitter streaming API, such as the mandatory existence of a word (e.g., conspiracy) within a tweet. Nevertheless, a more robust tagger can be easily constructed by taking into account any of the information which is provided together with the text.

The aforementioned steps result in the Twitter conspiracy dataset accompanying the text. In total, the dataset is composed by more than 4500 conversations, with an average of 1000 tweets per conversation and more than 33000 words. The dataset is annotated as conspiracy. This dataset can be downloaded from here: https://www.kaggle.com/borjaarroyogalende/twitter-conversations-conspiracy

Considerations on the Use of The Data: this research is part of an EU Funded project (see acknowledgements). The data was collected using an academic account and according to the authorisation of Twitter developer policies. The data was used only for the project objectives. Non-personal data

\(^3\)Kaggle Dataset for sarcasm detection. Available at https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection

\(^4\)Twitter API V2. Available at https://developer.twitter.com/en/docs/twitter-api/early-access

| Table 1. List of twitter parameters for the dataset creation. |
|-------------------------------------------------------------|
| Parameter name     | Example value           |
| AuthorID           | 57268907                |
| Created_at_twitter | 1620809009.0            |
| Possible sensitive | False                   |
| Conversation_id    | 139225830567396864      |
| Text               | focus on not supporting terrorism, eh? |

| Table 2. List of twitter features involved in the design of the BORJIS detector. |
|-----------------------------------------------|
| Feature name      | Example value           |
| Retweet count     | 57268907                |
| Created_at_twitter| 1620809009.0            |
| Possible sensitive| False                   |
| Conversation_id   | 139225830567396864      |
| Text              | focus on not supporting terrorism, eh? |
was processed (only conversations). The ID’s provided as the output of this work are available for download following the policies defined by Twitter.

IV. BORJIS ALGORITHM DESCRIPTION

A. FRAMEWORK ARCHITECTURE

The complete framework pipeline is depicted in figure 2 and contains two main modules. The first deals with Twitter crawling, data preprocessing, initial semi-automatic labelling, and ingestion into the database. The latter one is in charge of assessing the contents in a multistage approach. Both are below described.

B. DATA CRAWLING AND LABELLING

The starting point of data is located in the connection between our module and the Twitter API v2. This connection uses the filtered stream endpoint provided by Twitter to get a tweet with the requirements specified by the user. After that, the crawler sends another request with the conversation ID and extracts the full conversation related to that tweet. This way we can ensure that the data is recent and that it contains the filters that we may require.

Once the desired data has been gathered, the preprocessor module, whose purpose is to clean the text and extract specific fields that we consider are useful, is called. The preprocessor does not extract just some metrics for our research purpose, but for future studies within this topic. More specifically, the data preprocessor is capable of extracting emojis, hashtags, mentions and links. Moreover, the preprocessor extracts some basic features at the conversation level. Those features are composed by the number of tweets per second, the number of sensitive labelled tweets, the number of verified accounts participating in the conversation, the average number of mentions per tweet, the average ratio of followers per the following accounts and the average user tweets per second.

C. SEMI-AUTOMATIC LABELS

Once the information has been successfully extracted, the labeller module decides whether a conversation is classified as conspiracy or not. The tag “semi” appears due to the fact that every interaction with the Twitter API needs to be triggered and specified by the user, so the topic involved in each request needs to be defined manually.

The labeller takes into account several fields of the metadata accompanying the text (see Table 3) and creates a metric based on those indicators. In order to create the metric, each of the metadata fields is normalized to the (0, 1) interval, which leads to a type of probability value that can be aggregated in terms of a weighted sum. The following equation describes the process mathematically. 

\[
p_{\text{conspiracy}} \propto \frac{\text{conv\_tweets/sec}}{\text{tweet\_count}} + e^{-n_{\text{retweets}}} \ldots \\
\ldots + e^{-n_{\text{verified}}} + 1 - e^{-n_{\text{sensitive}}} 
\]

where the probability of conspiracy labelling (\(p_{\text{conspiracy}}\)) is proportional to the sum of the normalised relation of tweets posted in the conversation \(\frac{\text{conv\_tweets/sec}}{\text{tweet\_count}}\) plus several normalised indicators.

D. NEURAL NETWORK

The algorithmic solution involves a Neural Network (NN) consisting of several layers as shown in figure 3, described as follows (all these properties are synthesized in Table 6):

1) **Text-to-vec.** The starting layer is a text to vector map that translates words into numeric vectors. It is implemented as a hash table covering the 50000 most used words in terms of term frequency (TF). This layer is trained beforehand with the full list of examples, being immutable to test samples and acts as the input of the NN.

2) **Embedding.** An embedding layer is added in order to create a latent space consisting of fewer dimensions than the input space (specifically 8) acting as a summarizer that only keeps track of the most important features. Moreover, a L1 regularizer is applied in order not only to maintain the output as 0 and facilitate posterior computations, but also to prune the least informative attributes.

3) **Max pooling.** The pooling layer keeps track of the most extreme events that happen in the whole conversation. As it is the case of the embedding, it is used to accelerate posterior computations, also keeping the memory from possible overflow that may occur in other case.

4) **Bidirectional Gate Recurrent Unit GRU.** The recurrent layer traverses the ragged tensor and produces a constant length output, so it acts as a sequence to vector layer. Furthermore, several regularizers are applied to the GRU preventing it from overfitting.

5) **Dense structure.** The last stage is covered with the traditional dense structure with two layers that first reduces the input (1st layer) and then produces an output with a sigmoid as an activation function (2nd layer). Between these two layers a dropout regularizer is added to prevent from overfitting.

---

**TABLE 3. List of metrics for the initial labelling.**

| Feature name       | Explanation                          |
|--------------------|--------------------------------------|
| conv_tweets/sec    | Ratio of tweets/time                 |
| followers_count    | Average number of user followers     |
| following_count    | Average number of user followings    |
| following/followers| Average following/followers ratio of the users |
| like_count         | Average number of likes              |
| quote_count        | Average number of quotes             |
| reply_count        | Average number of tweet replies      |
| retweet_count      | Average number of retweets           |
| listed_count       | Average number of users in list      |
| list/followers     | Average ratio of users in list/followers |
| mentions/tweet     | Average ratio of mentions/tweet      |
| n_verified         | Number of verified accounts          |
| n_sensitive        | Number of sensitive tweets           |
| tweet_count        | Number of tweets                     |
| user_tweets/s      | Average ratio tweets/time of the users |
| hashtags           | All the hashtags included in the text |
| text               | Processed text                       |

---
It is important to note that the input length is not fixed (see figure 3, hence the algorithm works fine with any conversation extracted from Twitter. This feature adds value to the solution as it prevents from using padding, which would introduce computational overhead and might introduce noise in the recurrent layer. Moreover, it also treats the conversation as a whole, avoiding the use of sliding windows that potentially could produce noisy examples, as not all windows created would have the same label as the whole conversation. In other words, not every subset of contiguous words extracted from a conversation are obliged to have the same value as the whole set.

Prior to selecting this architecture, several other options are studied. The Table 5 sums up the hyperparameters for the best models obtained through tuning.

V. EXPERIMENTS AND RESULTS
In this section we present the outcomes and results obtained in the tests performed. The system setup is synthesized in Table 7. The aforementioned dataset is used for the training (around 4.5 Million tweets). The implementation of the proposed algorithm can be downloaded for comparison from here: https://gitlab.com/BorjaArroyo/conspiracy-article/edit#js-general-project-settings.

The model is trained on the dataset for 100 epochs with a strong regularisation in all layers. In the simulations, a 98.5% of accuracy was attained in the training stages.

Due to the special conditions of the dataset, some of the outputs are randomly selected to show the performance of the algorithms. These are drawn in the Table 4.

Moreover, the training performance in terms of accuracy achieved during epochs in the interval [1, 30] is shown in figure 4. Due to the regularization, training occurs slowly and it is not stable.

After training, the algorithm is validated against a validation set created entirely for this purpose. The validation set comprises 800 conversations that gather the various Twitter trending topics that occurred from the beginning of November to January. The following metrics are obtained: loss: 1.0352, auc: 0.9261, binary crossentropy: 0.5149, precision: 0.9552, recall: 0.7846, false negatives: 123, false positives: 21. Jointly, the true labels are compared to the predicted in figures 5 and 6. Note that the subfigure on the left has the
TABLE 4. Examples of tweets conversations, the corresponding label and outcomes of the BORJIS algorithm.

| Conversation ID | BORJIS | Label |
|-----------------|--------|-------|
| 13963809464493104 | 98.822403 | 1 |
| 138932130091145396 | 99.8085 | 1 |
| 13961251311377729 | 99.894047 | 1 |
| 13961656480339281 | 99.75371 | 1 |
| 139542347105462275 | 99.7005 | 1 |
| 13968165480339281 | 99.75371 | 1 |
| 1396109926120451 | 99.50552 | 1 |
| 1396462915159425024 | 99.40598 | 1 |
| 139549551216381827 | 99.77005 | 1 |
| 138067805653671978 | 99.79424 | 1 |
| 139549101529240066 | 99.876493 | 1 |
| 139711264261083484 | 99.88085 | 1 |
| 1396713099126120451 | 99.50552 | 1 |
| 139576797595427333 | 99.90028 | 1 |
| 13968165480339281 | 99.75371 | 1 |
| 13965559786582214 | 99.50552 | 1 |
| 85637084708584448 | 99.90028 | 1 |
|x axis rotated compared to each other. The results claim an accuracy of 0.82 in this validation set.

The assumption of independent and identically distributed data cannot be adopted as the topics searched happen to be trending topics at that moment. Not even all tweets related to a specific topic can be assumed iid as there is a huge random effect introduced by the behaviour of their authors. Thus, in order to adapt to the unlimited and time-dependent way of communicating, the algorithm should be likewise trained on an unlimited source of information.

However, considering that the trending topics are selected so that the conversations are long enough in conjunction with a wide time window of data exploration results in a broad applicability in one of the most used social medias.
TABLE 5. Summary of the best results obtained through hyperparameter tuning, where EOD refers to the embedding output dimensions, D is the dropout, AF means activation function, LR refers to the learning rate, NL is the number of dense layers, DU is the number of units per layer and Sc shows the score in terms of binary accuracy.

| Base     | Embedding | Kernel | Recurrent (RNN) | Activity (RNN) | EOD | D         | AF     | LR      | NL       | DU       | Sc       |
|----------|-----------|--------|----------------|---------------|-----|----------|--------|---------|----------|----------|----------|
| RNN      | 11_12     | 11_12  | 11             | 11_12         | 20  | No       | Tanh   | 1e-4    | 1        | [256]    | 0.988    |
| RNN      | 11_12     | 11_12  | 11             | 11_12         | 20  | No       | Tanh   | 4e-3    | 1        | [192]    | 0.987    |
| RNN      | 11_12     | 11_12  | 11             | 11_12         | 20  | No       | Tanh   | 2e-3    | 1        | [32]     | 0.987    |
| RNN      | 11_12     | 11_12  | 11             | 11_12         | 20  | No       | Tanh   | 5e-3    | 1        | [160]    | 0.987    |

TABLE 6. Summary of the hyperparameter values of the RNN architecture.

| Hyperparameter   | Value                  |
|------------------|------------------------|
| Batch size       | 32                     |
| Optimizer        | Adam                   |
| Loss             | Binary cross entropy   |
| Input size       | Variable length        |
| Regularizers     | L1, L2 in GRU kernel   |
| Dropout          | 0.4                    |
| Activation       | selu, sigmoid in output|

TABLE 7. System settings used in the experiments.

| Setting        | Value      |
|----------------|------------|
| Processor      | Intel i9-10900 |
| RAM            | 32 GB      |
| Python         | 3.7.12     |
| Tensorflow     | 2.5.0      |
| DB             | Mongo 4    |
| Twitter API    | V2         |

FIGURE 5. Results of PCA (right) and t-SNE (left) for the validation set with real labels of conspiracy.

FIGURE 6. Results of PCA (right) and t-SNE (left) for the validation set with predicted labels of conspiracy.

FIGURE 7. Accuracy obtained per epoch during the training phase.

Due to the fact that the proposed architecture can process variable length inputs, the most common applicable deep learning layers are the sequence-to-vector recurrent layer and the one-dimensional convolutional layer. The convolution shares parameters along the whole pass, so it suffers in accuracy. On the other hand, the recurrent layer provides a sufficiently complex design that captures the linguistic relations between words. The CNN approach cannot attain a considerable accuracy, as shown in figure 7.

VI. CONCLUSION AND FUTURE WORK

In this paper, a complete framework for spotting conspiracy is presented. The framework is intended to define an initial set of metrics that will permit semi-automatic labelling and ingestion of data. Metrics based on popularity, activity and basic text analytics. The goal of this initial data ingestion and population is to define a set of features to be ingested. We have created a dataset with multiple features which has been released in order to boost research on the topic. Furthermore, we present a Deep Learning framework to analyse and exploit the text similarities in Twitter conversations. This framework uses conversation sequences in order to exploit more complex semantic features of the text.

In terms of future work, the evolution of NLP and machine learning in general is moving towards data-driven quality solutions. The importance of data is undoubtedly emerging in contrast to complicated algorithmic solutions, hence the possible improvements could come from the redefinition of the labeller with better heuristics or even unsupervised learning. Furthermore, expert labelling of a small subset of data could be another approach in order to provide a seed for semisupervised classification.
REFERENCES

[1] D. M. J. Lazer, M. A. Baum, Y. Benkler, A. J. Berinsky, K. M. Greenhill, F. Menczer, M. J. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. A. Sloman, C. R. Sunstein, E. A. Thorson, D. J. Watts, and J. L. Zittrain, “The science of fake news,” Science, vol. 359, no. 6380, pp. 1094–1096, 2018, doi: 10.1126/science.aao2998.

[2] European Commission. Action Plan Against Disinformation. Accessed: Jul. 10, 2021. [Online]. Available: https://ec.europa.eu/sites/default/files/action_plan_against_disinformation.pdf

[3] R. Levy, “Social media, news consumption, and polarization: Evidence from a field experiment,” American Economic Review, vol. 111, no. 3, pp. 70–831, 2021.

[4] American Press. How people use Twitter. Accessed: Jul. 10, 2021. [Online]. Available: https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-news/single-page

[5] Twitter API Version 2. Accessed: Jul. 25, 2021. [Online]. Available: https://developer.twitter.com/en/docs/twitter-api/early-access

[6] A. Morales, A. Acien, J. Fierrez, J. V. Monaco, R. Tolosaena, R. Vera, and J. Ortega-Garcia, “Keystroke biometrics in response to fake news propagation in a global pandemic,” in Proc. IEEE 46th Annu. Comput. Softw., Appl. Conf. (COMPSAC), Jul. 2020, pp. 1604–1609, doi: 10.1109/COMPSAC46808.2020.00260.

[7] A. Ebrahimian Fard, M. Mohammadi, S. Cunningham, and B. Van de Walle, “Rumour as an anomaly: Rumour detection with one-class classification,” in Proc. IEEE Int. Conf. Eng., Technol. Innov. (ICE/ITMC), Jun. 2019, pp. 1–9, doi: 10.1109/ICEITMC.2019.8792644.

[8] L. Li, G. Cai, and N. Chen, “A rumor events detection method based on deep bidirectional GRU neural network,” in Proc. IEEE 3rd Int. Conf. Image Vis. Comput. (ICIVC), Jun. 2018, pp. 755–759, doi: 10.1109/ICIVC.2018.8492819.

[9] S. Geng and E. Surer, “Detecting ‘Clickbait’ news on social media using machine learning algorithms,” in Proc. 27th Signal Process. Commun. Appl. Conf. (SIU), 2019, pp. 1–4, doi: 10.1109/siuj.2019.8866257.

[10] S. Chawda, A. Patil, A. Singh, and A. Save, “A novel approach for clickbait detection,” in Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI), Nov. 2020, pp. 483–487, doi: 10.1109/ICCCSP49186.2020.9342742.

[11] B. A. Galende received the degree in environmental engineering from the Universidad Politécnica de Madrid (UPM), in 2019, and the degree in computer engineering from the Universidad Nacional de Educación a Distancia (UNED), in 2020, where he is currently pursuing the master’s degree in computer science and engineering. At the same time, he is working as a part of the GATV Research Group, UPM. His main research interests include data driven applications, ML research in general, and specifically, probabilistic learning.
GUSTAVO HERNÁNDEZ-PEÑALOZA (Member, IEEE) received the Telecom Engineering degree from Universidad Santo Tomás, in 2007, the Master of Science degree in telecommunications technologies, system and networks from the Universidad Politécnica de Valencia (UPV), in 2009, and the Master of Business Administration (M.B.A.) and Ph.D. degrees (cum laude) from the Universidad Politécnica de Madrid (UPM), in 2019. From 2010 to 2013, he was an Associate Research Fellow with the Universidad de Valencia (UV). He is currently working as a Postdoctoral Fellow with UPM. He has been involved in multiple national and European funded projects in technical and management activities. His main research interest includes artificial intelligence (AI) applied to healthcare.

SILVIA URIBE received the Telecom Engineering degree (Hons.), the master’s degree in communications technologies and systems, the master’s degree in telecommunication management, and the Ph.D. degree (cum laude) from the Universidad Politécnica de Madrid, in February 2008, September 2010, 2013, and 2016, respectively. She has been a member of the Visual Telecommunication Application Group (GATV), since 2006. She is the author and the coauthor of several papers and scientific contributions in international conferences and journals. Her professional research interests include interactivity technologies, content personalization technologies, and big data. Related to this, she has been participating with technical responsibilities in some national and European projects.

FEDERICO ÁLVAREZ GARCÍA (Member, IEEE) received the Telecom Engineering (Hons.) and Ph.D. (cum laude) degrees from the Universidad Politécnica de Madrid (UPM), in 2003 and 2009, respectively. Since 2003, he has been working for the research group of the Visual Telecommunications Applications Group (GATV), UPM. He is currently working as an Assistant Professor at UPM. He has been participating with different managerial and technical responsibilities in several national and EU projects, being a coordinator of five EU projects in last six years. He has participated in national and international standardization for a (DVB and CENELEC TC206). He is the author and the coauthor of more than 60 papers and several books, book chapters, and patents in the field of ICT networks and audiovisual technologies. He is a member of the program committee of several scientific conferences.

* * *