Article

Similarity approximation of Twitter Profiles

Niloufar Shoeibi [1]*, Nastaran Shoeibi[2],Pablo Chamoso[1,3], Zakieh AlizadehSani [1], and Juan M. Corchado [1,3]

[1] BISITE Research Group, Universidad de Salamanca, Salamanca, Spain; Niloufar.shoeibi@usal.es
[2] Babol Noshirvani University of Technology, Babol, Mazandaran, Iran.
[3] Air Institute, IoT Digital Innovation Hub, Salamanca, Spain
* Correspondence: Niloufar.shoeibi@usal.es; Tel.: +34-617-939-365

Abstract: Social media platforms have been entirely an undeniable part of the lifestyle for the past decade. Analyzing the information being shared is a crucial step to understanding human behavior. Social media analysis aims to guarantee a better experience for the user and risen user satisfaction. However, first, it is necessary to know how and from which aspects to compare users. In this paper, an intelligent system has been proposed to measure the similarity of Twitter profiles. For this, firstly, the timeline of each profile has been extracted using the official TwitterAPI. Then, all information is given to the proposed system. Next, in parallel, three aspects of a profile are derived. Behavioral ratios are time-series-related information showing the consistency and habits of the user. Dynamic time warping has been utilized for the comparison of the behavioral ratios of two profiles. Next, the audience network is extracted for each user, and for estimating the similarity of two sets, Jaccard similarity is used. Finally, for the Content similarity measurement, the tweets are preprocessed respecting the feature extraction method; TF-IDF and DistilBERT for feature extraction are employed and then compared using the cosine similarity method. Results have shown that TF-IDF has slightly better performance; therefore, the more straightforward solution is selected for the model. Similarity level of different profiles. As in the case study, a Random Forest classification model was trained on almost 20000 users revealed a 97.24% accuracy. This comparison enables us to find duplicate profiles with nearly the same behavior and content.

Keywords: Twitter; Social Media; Social Networking; Social Network Analytic; DistilBERT; Text Similarity; Natural Language Processing; User Engagement.

1. Introduction

Social media platforms are now a part of the lifestyle of human beings of any age. This popularity has its advantages and disadvantages. The great benefit is the faster and easier communication and overcoming physical limitations [1].

Social networks are the many overlapping networks that link and move friendships, information, money, power, Etc. We can gain new insights into culture, politics, history, and many other things by analyzing social networks. In other words, users' connections are a significant factor in what they know and how they think [2]. Social network analysis allows us to quantify the connections between individual points. It helps to find patterns in the connections that sustain the society [3]. How the individual is connected or disconnected from people, groups, or populations; in other words, how the individual distributes their energy across different social groups over time or explores how an idea, belief, or disease passes through the individual's network [4].

Social media analysis aims to understand people's behavior, provide more safety and achieve higher user satisfaction. There are different malicious types of activities; rumor control [5], detecting fake news and stopping their propagation [6], fake profiles, and bot detection [7], and detecting duplicate profiles.

As much it is essential to have a safe society, it is crucial to guarantee the safety of the virtual community users. One step to make this society more secure is to detect and remove these duplicate profiles. These profiles can induce unethical thinking or activities, such as sexist ideologies [8]. Sometimes, the aim is to detect specific topics in real-time[9]. Other times, assuring cybersecurity is a challenge because the safety of social networks is a tremendous concern due to mass user engagement, so the aim is to detect criminal activities and eliminate them [10,11]. Considering the possibilities that social media platforms give for illegal activity, it is crucial to analyze the behavior of users.

For detecting duplicate profiles, it is mandatory to know how to compare two profiles [12]. The idea of seeing identical profiles led to implementing a methodology for comparing Twitter users, considering three aspects: behavioral similarity, audience similarity, and context similarity. Having the information derived from the users' comparison enables the detection of duplicate profiles, eliminates these frauds, and protects users' policies and privacy, leading to a secure virtual society to make life more comfortable, safe, and updated [13].

The research's questions:

- How to define the similarity of the profiles?
- From which aspects are two profiles similar?
- Which similarity measurements can calculate the distance of the two profiles?
- Which features define the selected aspects of a profile to calculate the similarity?

As the behavior of users on social media is all related to human beings, countless features are affecting their behavior, and there may be a reason for certain behaviors, while for others, no. Also, the chain of transformation of each act flows through all the users worldwide, as in the butterfly effect, creating a stochastic-dynamic environment, which makes it more challenging to analyze and find behavioral patterns. However, the biggest challenge in this kind of research is to define the aspects that profiles can be similar to each other [14].

This article focuses on Twitter, a social media platform, enabling two-way communication and interacting with other users quickly and easily. Twitter allows users to generate content by posting "tweets" and sharing other users' content by "retweeting" [15]. The proposed architecture considers three aspects of similarity measurement, calculates the similarity, and decides whether they are replicated or not. These aspects are *the audience similarity*, who are interacting with the profile and its contents, calculated using the network of audience. *the behavioral similarity* that are ratios of activities of account, for example; for two accounts are tending to constantly post the same amount of tweets in the morning between 9:00 to 12:00 am. *the content similarity* is measured through two strategies—the number of the same tweets/retweets and the context similarity. The similarity of the context is calculated using a TF-IDF text vectorizer and the Cosine Similarity.

For doing it, the user's timeline is extracted as a list of Tweet objects, which are the entities containing all the tweet information. Then in parallel, the audience, the users interacting with the primary user are obtained; besides, behavioral ratios, the time-series-related features are calculated; moreover, the content, the tweets, and retweets the user has posted are collected. For comparing the audience, the inter-communications network of the primary profile is created and later compared to the other profiles' audience and measure the overlap of a user's audience with another.

For the next aspect, the frequency of the user's activities is calculated during the time and being compared to another user's features using dynamic time warping (DTW) [16].

For checking the context similarity, two ways have been taken into account;

How many tweets are precisely the same?

How much are the concepts of the posts on different users' timelines similar? They can be in the same language or not.

One of the essential techniques in natural language processing is text feature extraction, which turns the texts into the numerical vector representing the words and the sentences. In this research, two strategies are examined; a straightforward technique like TF-IDF calculates the vectors based on the frequency of appearance of the words. And DistilBERT, which is the encoding part of the transformer architecture of the language model. DistilBERT is a pre-trained language model. Then, the cosine similarity between these vectors is calculated. It is necessary to mention that the preprocessing for these two methods is different.

A language model is a distribution over sequences of tokens or symbols of words in a language. A good language model of English can look at a sequence of words or characters and tell how convincing it is to happen in English, how it is possible to be an English phrase or a sentence, then use it for many different tasks. For generating text, users can sample from that distribution and put conditions on the probability distribution look like for the other words and keep giving it the output. The language model is in many tasks such as Translation, Summarization, Chatbox, and enhancing many language-related tasks [17].

One of the developments in language models is handling dependencies of any kind but especially long-term dependencies. Recurrent Neural Networks (RNN) suffer from short-term memory. So for longer paragraphs, RNN may miss important information from the beginning [18].

Long Short Term Memory networks (LSTM's) are a particular kind of RNN capable of learning long-term dependencies. They were designed to fix the long-term dependency problem. LSTM has internal mechanisms called gates that have the stream of information and decide which parts of the input to pay attention to, which features to use in the calculation, and which parts to ignore [19].

DistilBERT is lighter and faster, with 40% of the size of standard BERT, saving 97% of its language understanding capacity but 60% faster. The output vector size is 768, meaning each sentence will have a fixed-size vector with 768 values [20].

This paper has been organized as follows: In Section 2, the related work is presented; then, in Section 3, the architecture of the proposed method is described. In section 4, a successful case study is outlined, and its results are overviewed. Finally, in Section 5, conclusions are drawn, and future lines of research are discussed.

2. Review of state of the art

Many works have been done on data analysis [21], but the focus is on data extracted from social media platforms [22–24] in this paper. In Social Media Analysis, there are many rooms left for investigation and improvement of the existing tools and algorithms. In the literature, several pieces of research have been done on data extraction. However, it is necessary to consider that each platform has its policies for data extraction and publishing. For instance, Twitter allows researchers to extract public information via official Twitter APIs and conduct academic research to make improvements. However, in general, most of the research in this area is related to taking advantage of social network data and applying Artificial Intelligence algorithms, such as machine learning methods (supervised and unsupervised), deep learning, graph theory, Etc. In this paper, the focus is on the calculation of the similarity of the profiles on Twitter.

A social network can be interpreted as a complex network graph consisting of nodes connected by edges. The nodes represent the users in the network, and the edges define the connections between these users. Social network analysis requires specific analysis tools; Akhtar et al. in [25] conducted a comparative study of these tools in general graph analysis and social network analysis. They conducted a comparative study of four social network analysis tools (NetworkX, Gephi, Pajek, and IGraph) based on platform, runtime, graph type, algorithm complexity, input file format, and graph features.

Semantic analysis is a powerful technology in Natural Language Processing (NLP) applications. Such as text similarity estimation, text classification, speech recognition, Etc. Chen et al. introduced a framework for semantic similarity detection for deep reinforcement learning for the Siamese attention structure model (DRSASM). It automatically detects the word segmentation and word distillation features and proposes a new recognition mechanism model to improve semantics [26].

There are many strategies for Similarity detection depending on the final goal, such as Euclidean distance, Pearson correlation coefficient, Spearman's rank correlation coefficient, and others. Chicco et al. review applying The Siamese neural network architecture for complicated data samples that have different dimensions and types of features [27].

Semantic similarity detection in text data is one of the challenging obstacles of Natural Language Processing (NLP). Due to the versatility of Natural language, it is challenging to represent rule-based methods for detecting semantic similarity patterns. Chandrasekaran et al. determine the evolution of several available semantic similarity methods and review their pros and cons. Classify by the underlying policies as corpus-based, hybrid approaches, knowledge-based, and deep neural network-based methods [28].

In [29] they introduced a Siamese model of the Long Short-Term Memory (LSTM) network for assessing the semantic similarity between texts. They add word-embedding vectors enhanced by synonymic data to the LSTMs, based on a fixed size vector to encode the underlying meaning implied in a sentence. They constrain the sentence representations detected by the proposed model to create a structured space whose geometry reveals complex semantic similarities. It reduces subsequent procedures for relying on a simple Manhattan metric.

Siamese Neural network is a method for computing similarity with demanding less training data. An architecture with language-independent features for finding short text similarity detection in multiple languages and domains was proposed in [30]. They used these corpora from shared tasks: ASSIN 1 and ASSIN 2 with Portuguese journalistic texts and N2C2 (English clinical texts). Then implemented the proposed SNN by Mueller et al. in two forms. The evaluation calculates the Pearson correlation (PC) and the Mean Squared Error (MSE) among the models' predicted values and corpora's gold standard. This method held better results in both languages and domains.

Many studies are done aiming to improve the performance of text classification models, such as centroid-based classifier, multinomial naïve bayesian (MNB), support vector machines (SVM), and convolutional neural network (CNN). However, Park et al. presented a cosine similarity-based methodology to enhance the performance. For increasing the precision of classifiers, This methodology merges cosine similarity and conventional classifiers, And then the Conventional classifiers with cosine similarity are named enhanced classifiers. Enhanced classifiers are applied to famous datasets such as 20NG, R8, R52, Cade12, and WebKB, And they show notable accuracy improvements. Also, word count and term frequency-inverse document frequency (TF-IDF) is more suitable in terms of the performance of the classifier [31].

A variety of users and content in online social networking sites (OSN) will cause a fear of identity theft attacks (profile cloning), malware attacks, or structural attacks of cybercriminals. Profile cloning is stealing existing users' identities and creating duplicate accounts with the existing users' credentials. Chatterjee et al. proposed a way to supervise the threat of profile cloning in social networks. Users can use it to prevent cloned, and fake profiles and identity theft [32].

In [33] a detection technique is proposed for discovering fake and cloned profiles on Twitter. For detecting profile cloning, they used two methods: similarity measures and the C4.5 decision tree algorithm. In Similarity measures, Similarity of characteristics and Similarity of network relations are analyzed. C4.5 applies a decision tree by considering information gain. These two methods help in detecting clone profiles and preventing them.

A framework for finding cloned profiles in social networks is stated in [34]. It will analyze user profiles, friends and follower networks, and posting habits. This framework has three parts: Twitter Crawler, Attribute Extractor, and Cloning Detector. The best classification performance is with the decision tree, and the average accuracy of classifying the real or fake posts was 80%.

BERT is a method to merge topics by pre-trained contextual representations. For pairwise semantic similarity detection, Peinelt et al. proposed a unique topic-informed BERT-based structure. This advanced architecture performance over strong neural baselines beyond different classes of English language datasets. Adding topics to BERT helps in determining domain-specific problems [35].

Knowing which text feature extraction strategies will perform better depends on the text being analyzed, mostly on its length. For example, in [36,37], D. Carun et al. performed sentiment classification on the banking financial news, and they discovered that Distilbert as text feature extractions performs better than TF-IDF with a 7% improvement of accuracy. On the other hand, In [38], I. Vogel et al. investigated monitoring and defining the factors for detecting the Twitter users who are spreading the hate of speech. They have realized that the simple n-grams feature extraction and traditional machine learning models like SVM performs better than BERT feature extraction and Bi-LSTM.

The variety of the results of the investigations in selecting the better text feature extraction methodology for further natural language processing tasks is one of the motivations for examining both methods in this work.

The following section presents the proposed method, combining the information extracted from the three aspects of profile behaviors. A couple of examples of knowledge inquiring of each aspect are reviewed in state-of-art, and the ideas helped design and implement the proposed method.

3. Proposed Model

Twitter which is the most news-friendly social media platform, is the primary focus for applying this tool. However, the proposed methodology can be used on different social media platforms with a few modifications. Twitter has a unique feature among all social media: On Twitter, users can respond to each others' tweets and "like" each other's tweets or leave a comment and leave comments to share their opinions and viewpoints. Tweets can include text, photos, videos, links, Etc. Users can also share the status of other users' tweets by retweeting them. The data relating to the tweets and some information about the profiles are provided as Tweet Object in JSON format [39]. In this section, the platform's architecture for measuring the similarity of profiles is presented. It extracts data from a user's Twitter timeline, analyzes them, and transforms them into meaningful information. Below, the functionality of the platform is described in a very general way.

The first step is extracting the recent tweets of a profile by extracting the user's timeline and storing it as a list of JSON files. Then, in the Advanced Feature Extraction component, the data is restructured. More advanced features are created from the selected primary features, including time-series features and ratios indicating the users' behavior and habits, like the number of tweets posted each day. At the same time, from the JSON file extracted from the user's timeline, the audience is the people who are interacting with the chosen profile, defined by the replies and the retweets. In parallel, all tweets of a profile are selected by its language. Not English ones are translated to English; depending on the feature extraction method, they are preprocessed and turned into vectors. The results showed that the two selected feature extraction strategies of TF-IDF and DistilBERT have very similar results. Even TF-IDF has slightly better performance; therefore, TFIDF has been chosen to be implemented in the platform.

The output of all these components will be handed to the Similarity Measurements component. A respective similarity measurement has been applied for each aspect of these new features, explained in detail in each relative sub-section. Algorithm 1 represents the process as a pseudo code.

Algorithm 1 Proposed Method's Algorithm.

```
Inputs: Screen_names of the Twitter profiles.
Output: Similarity of the profiles
Step 2, 3, 6, 10 are executing in parallel Begin
 1: for each Twitter User
 2:         Extract the timeline
 3:         Feature Engineering
 4:                 Extracting Primary Features
 5:                 Extracting Advanced Features
 6:         The Audience Network
 7:                 Extracting the Audience of the user
 8:                 Building a set of users who are interacting with the user
 9:                 Finding the strong friendships of the user
10:         Content Preprocessing
11:                 Text Preprocessing
12:                 Text Feature Extraction (Text Vectorization)
13:         end for
14: Similarity checking
15:         for each Twitter User
16:                 Advanced-Features Similarity Approximation
17:                         Dynamic Time Warping (DTW)
18:                 The Audience Network Similarity Detection
19:                         The overlap between sets of Audiences using Jaccard Similarity
20:                 Content Similarity Measuring
21:                         The number of same tweets/retweets
22:                         Cosine Similarity
23:         end fore
End
```

This architecture is optimized because it has been designed in the most parallel way possible and consists of four components described in the following subsections. The outputs of this model aim to provide a better understanding of how similar two profiles are by calculating the distance between the users from the mentioned points of view.

The proposed architecture is presented in Fig 1. The designed system aims to calculate the similarity of the profiles based on their network of the audience, behavioral traits, and context similarity. This architecture consists of five main components: Timeline Extraction, Advanced Feature Extraction, The Audience Network, and Content Processing. These aspects are discussed in their respective subsections.

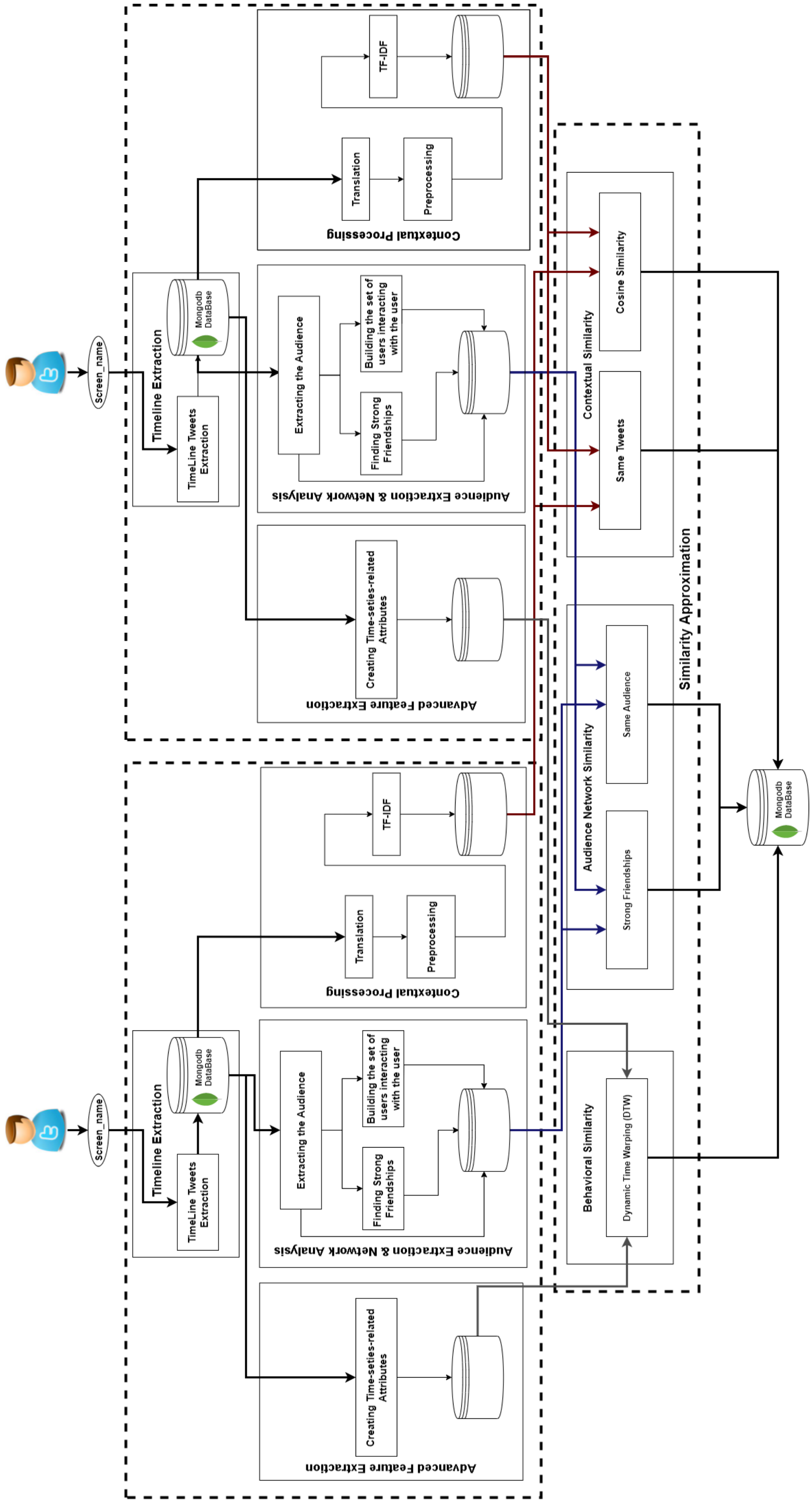

Figure 1. The proposed model for similarity measurement of the profiles.

This architecture is optimized because it has been designed in the most parallel way possible and consists of four components described in the following subsections.

3.1. Timeline Extraction

Social media platforms like Twitter enable people to distribute and utilize news by interacting with each other and following some policies. The way they share and spread news contains remarkable meaningful hidden information that is interpreted into complex conclusions, from law obligation [40], to the marketing point of view [41].

In this research, the aim is to calculate the similarity between different Twitter profiles. The first step is to extract the timeline of the user using a component called Timeline Extraction. In this component, the official Twitter API, which is provided by the Twitter development team, has been utilized [42]. It extracts the timeline of the given screen_name of the profile. The 3400 recent tweets on the timeline are extracted using the official Twitter API. The only thing that is mandatory to consider is the API limitations [43]. There are many ways to deal with it. The output of this component is a list of tweet objects containing all the information of the tweets in JSON format.

3.2. Advanced Feature Extraction

In this component, data is restructured, and more complex concepts are derived from the primary features existing in the tweet objects. Especially the behavioral inclination of the user is defined by considering the ratios of activities during the time. Extracting these time-series-related features make us enable to extract behavioral patterns. In other words, calculating these features gives extra information about the user, which is a reliable measure for comparing the similarity of the profiles; for example, User 1 tends to post in the mornings. However, User 2 has a higher ratio of activities during the night. The output of this component is the set of advanced new features during the user's recent activities. These features have been presented in Table 1.

Table 1: The advanced features extracted from the user's most recent 3240 posts.

Feature	**Description**
Tweets per day	The number of statuses the user has posted in the recent posts (Tweets + Retweets)
Posted retweets per day	The number of retweets the user has posted in the recent posts
Retweets received per day	The number of likes the user has received recently
Replies per day	The number of replies the user has done in the recent posts
Mentions per day	The number of times the user has mentioned others, in the recent posts

After measuring the time-series-related features, the distance between the same time series features in two profiles will be calculated. Dynamic Time Warping (DTW) [44] is a suitable distance similarity measure that allows the comparison of two time-series sequences with different lengths and speeds. This algorithm is a perfect choice because the time series's length in various features varies and depends on how much the user was active recently. In other words, a user with a high ratio of activities can make 3240 tweets in one month; however, another user does this amount of posts in three months. Algorithm 2 shows the details of how DTW works. This algorithm computes and returns a dynamic time warping (DTW) similarity measure between (potentially multivariate) time series [45]. In the proposed model, the distance between two time-series features, for example, the distance between the mean number of posted statuses per day, exposes how similar two profiles behave during the extracted timeline.

Algorithm 2 Dynamic Time Warping Algorithm [16].

```
Inputs: TSF1, TSF2: Time series Features.
Output: D: The distance betweenn TSF1, TSF2.
Begin
  M(length(TSF1), length(TSF2)) : Cost matrix;
  D(1,1) ← (TSF1(1)-TSF2(1))2;
  for i ← 2 to length(TSF1)
      cost ← (TSF1(i)-TSF2(1))2;
      D(i,1) ← cost + D(i-1,1);
  end for
  for j ← 2 to length(TSF2)
      cost ← (TSF1(1)-TSF2(j))2;
      D(1,j) ← cost + D(j-1,1);
  end for
  for i ← 2 to length(TSF1)
      for j ← 2 to length(TSF2)
          cost ← (TSF1(i)-TSF2(j))2;
          D(i,j) ← cost + minD(i-1,j), D(i,j-1), D(i-1,j-1);
      end for
  end for
  D ← sqrt(D(length(TSF1),length(TSF2)));
  Return D;
End
```

Table 2 is an example of the similarity of two different features. Understandably, the @User_4 and @User_5 are more similar in the behavioral ratios (likes received, posted statuses, etc.). The results of applying the Dynamic Time Warping algorithm have been presented in Table 2.

Table 2: The advanced features extracted from the user's most recent 3240 posts.

Users	Posted Statuses	Posted Retweets	Posted Tweets	Likes Received	Retweets Received	Polarity	Subjectivity
@User_1 & @User_2	104.50	10.23	10.23	634612.10	73611.20	7.16	7.36
@User_2 & @User_3	114.26	6.78	12.13	48.90	611.73	27.65	16.35
@User_4 & @User_5	102.44	0.40	0.05	0.10	0.00	0.20	0.02

3.3. *Audience Network*

The attitude of any human being towards their environment and others can tell a lot about them. One of the most significant sources of information is by achieving information about the users connected to a specific user. An audience is a group of users interacting via retweeting, quoting, replying, and mentioning. Mapping this information into a directed graph makes analysts derive further information by simply considering the nodes as users and the edges as the connection. That is one of the possible ways of retweeting, quoting, replying, and mentioning. Categorizing the audience based on the frequency of links, called weights of digraph, is a way to measure the acquaintanceship of the profile audience. In this sample case, the scenario is as below:

- @User_1 has mentioned @User_2, 4 times
- @User_3 has retweeted from @Use_2, twice
- @User_3 has quoted a tweet from @User_1, 2 times
- @User_4 has replied on the statuses of @User_1 and @User_3, once each

The relationship matrix of this scenario is shown in Table 3.

doi:10.20944/preprints202106.0196.v2

Table 3: The Network Relationship Sample.

Source	Target	Weight
@User 1	@User 2	4
@User 3	@User 2	2
@User 3	@User 1	2
@User 4	@User 1	1
@User 4	@User 3	1

The strong friendships between users are defined based on the frequency of repetition of the links between two users; in this case, based on the network relationship sample, the *Weight* determines the strength of a friendship. In the "Audience Network" component, first, relationships between people and the list of the screen names of the audience in contact with the primary user are extracted. The main user whose timeline has been extracted in this representation is in the middle, and all other nodes are connected to this node. To compare the similarity between two users, nodes, the overlap of the audience of these two nodes needs to be calculated.

Jaccard's similarity has been applied to the set of the audience of different profiles to calculate the similarity between the two sets. It's a classical measure of the similarity between two sets, introduced by Paul Jaccard in 1901 [46]. Given two sets of audiences of @User_A and @User_B, Jaccard's similarity is measured by dividing the number of nodes from the audience of @User_A that exists in the audience @User_B of a total number of nodes minus the same nodes in both sets.

$$jaccard(Set\_1, Set\_2) = \frac{|Set\_1 \cap Set\_2|}{|Set\_1 \cup Set\_2|} \tag{1}$$

The intersection of two sets points out the common nodes between two of them, and the union of the sets means to sum the number of the audience of each profile but remove the ones that are repeated (the intersection).

3.4. Content Processing

In this step, the similarity of tweets is calculated in two aspects; the number of exact same tweets and the likeness of the content. The number of the same tweets is easy to calculate by checking the content of two timelines. However, two different text feature extraction strategies have been applied to turn the words into the vector of numbers to calculate the similarity in context; the TF-IDF and DistilBERT. Before using text vectorization (text feature extraction), the text needs to be preprocessed. Each of these methods requires a particular preprocessing approach.

It is worthy of mentioning that this component aims to calculate how similar the tweets of the two profiles are. Users can post these tweets in different languages. For unifying these tweets, they need to be turned into the same language, English, in this case. Google translate API [47], covers a very vast range of different languages. Therefore, before doing any further analysis, the languages of the tweets are unified to English. Also, It is necessary to check the dictation of the words because due to the limitation of the number of characters possible to post as a tweet, which is 280 characters, users usually abbreviate the words to add more information to the tweet. Hence, returning these abbreviations to the original terms is necessary. After preparing the text of the tweets, two mentioned feature extraction methods are applied. The following subsections give more information about each process.

3.4.1. Text Feature Extraction using DistilBERT

Language models are deep neural network models that are context-sensitive and comprehend the language and probability of appearance of the words in sequence. The quality of performance of NLP tasks highly depends on how extensive the network is and the data that is trained on [48].

As language models are deep neural network models that deal with sequential data (i.e., words in the sentences), deep neural models are commonly used to implement language models. Recurrent Neural Networks (RNNs) and Long Short-Term Memories (LSTMs) are the most famous examples. The problem with RNNs and LSTMs, apart from the complexity of the network, long training time, and computation expenses, is that the memory of these networks is limited, meaning that the longer the text, the more information is lost from the beginning of the text. LSTMs are an improved version of RNNs that can selectively remember the past by employing a gating mechanism. Bi-LSTM is a version of LSTM that can move through the sequence in both ways [49]. However, the data must still be passed through the network sequentially, which is a considerable disadvantage. To solve all these challenges, Transformers, which are attention-based models [50], are appeared.

The transformer consists of two key elements, encoder and decoder. The encoder learns what grammar is, what context is, what language is [51]. It contains a self-attention mechanism and a feed-forward neural network. Self-attention is an attention mechanism correlating different forms of a single sequence to estimate the design of the sequence. The decoder is a word embedding concatenated with a context vector that is both made by the encoder. BERT is the stacked encoders. It is used in many tasks like Neural Machine Translation, Question answering, Sentiment analysis, Text summarization, and many more. Pre-training BERT can explain these tasks to learn the language and fine-tune it to learn particular tasks. Training of BERT has two phases. The first phase is pre-training, the model understands the language and context, and the second phase is fine-tuning, and the model learns the language but does not know how to solve this problem.

The goal of pre-training is to make BERT learn what a language is and what context is. BERT learns language by training on two unsupervised tasks simultaneously. They are masked language modeling(MLM) and next sentence prediction(NSP). For MLM, BERT takes in a sentence with random words filled with masks. The goal is to output these masks tokens. It helps BERT understand a bi-directional context in a sentence for predicting the subsequent sentences. BERT takes in two sentences and decides if the second sentence supports the first. This kind of binary classification problem helps BERT understand context over different sentences themselves and use both of these together [52].

In fine-tuning phase, BERT is trained on particular NLP tasks by training both MLM and NSP to reduce the merged loss function of the two strategies. Rather than LSTM that has to hang on to an enormous amount of memory, BERT can selectively look at the relevant things, and the system learns where to look, where to pay attention [53].

DistilBERT is a smaller, quicker, and more affordable version of BERT [20] which includes 40% of the size of original BERT, by maintaining 97% of its language comprehension capability yet 60% quicker. DistilBERT transforms the input sentence into a fixed-size vector with 768 values. The significant advantage of using DistilBert for feature extraction is that all the words in the sentence are vectorized simultaneously in parallel, but will it have a notable impact on the results compared to the less complex methods? It is the primary motivation to try the process explained below and compare the results together.

3.4.2. **Text Feature Extraction using TF-IDF**

TF-IDF stands for Term Frequency Inverse Document Frequency. It is a well-liked algorithm for converting text into a meaningful representation of numbers to adapt machine learning algorithms for prediction. The count vectorizer provides the frequency count for the word index, and TF-IDF considers the overall word weight document [54].

The documents with similar content will have similar vectors. Formulas below represent the process of calculating the vector for each word using TF-IDF [55].

$$TF\_IDF = TF * IDF \tag{2}$$

$$TF(t,d) = \frac{f(t,d)}{\sum_k f(w_k,d)} \tag{3}$$

$$IDF(t,d) = log(\frac{N}{1+df_t}) \tag{4}$$

Where;

f(t,d) represents the number of accurance of term t in document d.

N is the total number of documents in the corpus.

df_t is the number of documents containing the term t.

The TF-IDF performs slightly better than DistilBERT because of the limit character in posting tweets, 268 characters. So the length of the vector in TF-IDF is much smaller than DistilBERT, a fixed size of 768 values. Moreover, both possible models transform the text into the related vector, and the similarity metrics have been compared as an experiment. TF-IDF is a straightforward model with the benefits of a simple model, meaning that it is fast, and DistilBERT, explained in detail, has a better understanding of the context of the language. Fig 2 shows that the distribution of the density of the two models is very similar. Therefore, the simpler model, TF-IDF, has been chosen for the further process.

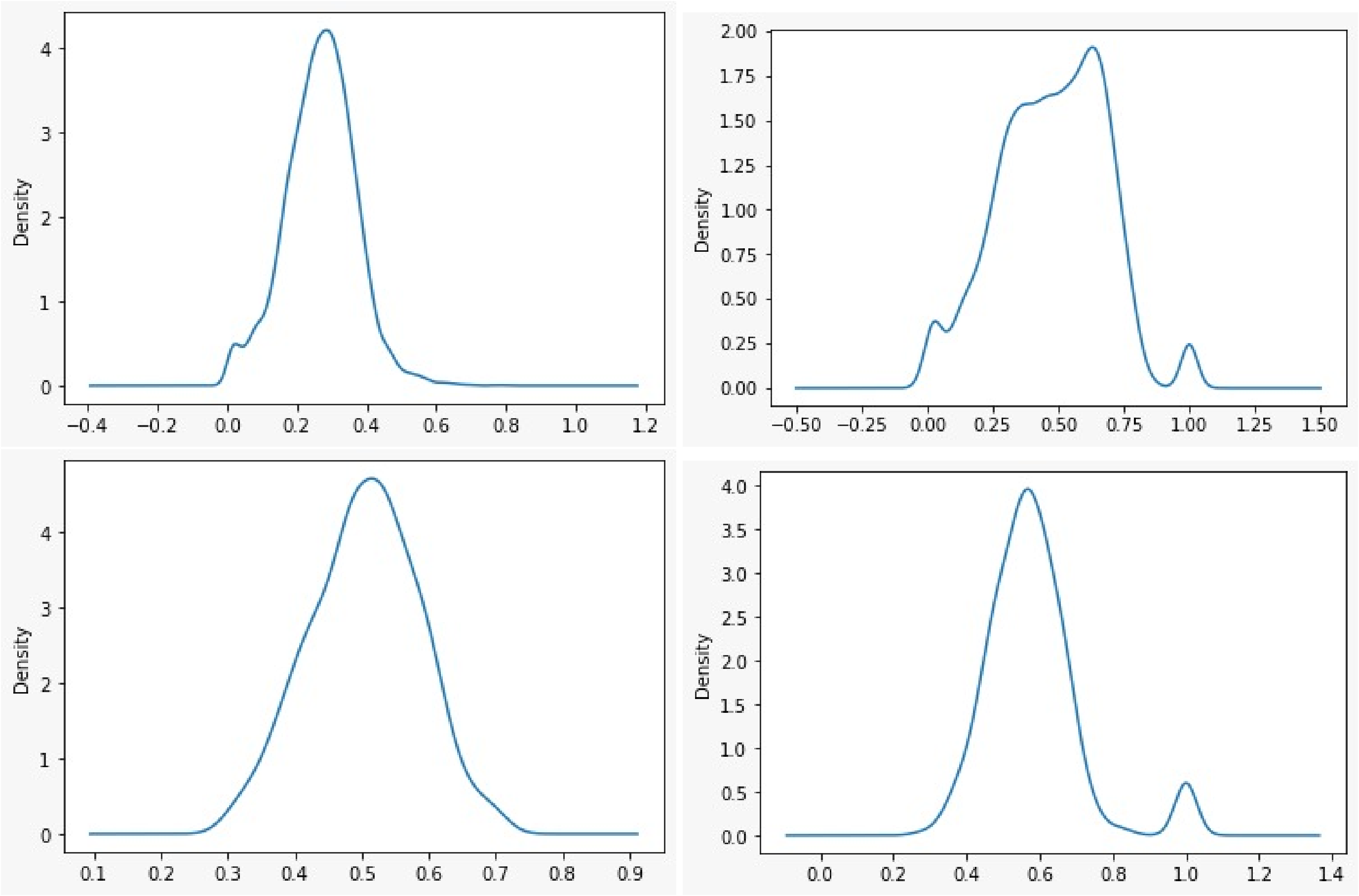

Figure 2. The density distribution of the similarity values calculated by TF-IDF and DistilBERT for the similar profiles and the not similar ones.

After applying this vectorization method, the distance between these vectors is calculated using the cosine similarity algorithm.

Cosine Similarity

After applying this vectorization method, the distance between these vectors is calculated using the cosine similarity algorithm. Cosine similarity is a measure used to assess the similarity of documents, regardless of their size. Mathematically, it measures the cosine of the angle between two vectors projected in multidimensional space. Cosine similarity is helpful because two similar documents can be separated by a Euclidean distance (due to the size of the document) but oriented closer to each other. The smaller the angle, the greater the cosine similarity [56].

4. Case Study and Results

Evaluating the proposed solutions and methods for solving social media-related problems, especially Twitter, is so sophisticated. Due to the policy changes of twitter during the previous years, there are so many research articles like [57] that are addressing the same problem but proposing the solutions under different policies and restrictions; also, the way they grouped the data make them lose the sense of living in different time zones because when a query is done, @user_X can be a user from Japan and @User_Y is from the United States of America. Twitter doesn't reveal the Geo-location data. So, both of these queries are matched with the timezone of the person who is making the query; it doesn't make sense to compare a user's behavior at night with another one in the early morning, etc. But suppose the activity ratios of the user are considered as time series. In that case, Dynamic Time Warping algorithm is more flexible to the repeated patterns and will feel the difference in time zones.

Offering solutions under other constraints lead to creating different solutions for the same issue by improving the technologies and algorithms and dealing with various rules. For instance, in [58], which was published in 2013, for validation of their model, they used and published a dataset of users and the respective identity of each user. It was possible in 2013, but since 2016, the policies have been changed, and they are much more restricted, forcing us to design solutions considering other points of view.

As a case study, a dataset has been created using almost 100 US politicians and Senators and 100 top singers. The dataset is created by comparing all the possible two selections of Twitter users. Therefore, the size of the final dataset is almost 20,000 containing the comparison values calculated by this model and labeled manually, considering the *singers* are similar with each other and *Politicians* are alike with each other. However, they are similar but not identical. And also, the politicians and singers are different.

Fig 3 presents the distribution of the labels among these almost 20,000 couples of users for comparison. As has been demonstrated, it is a balanced dataset.

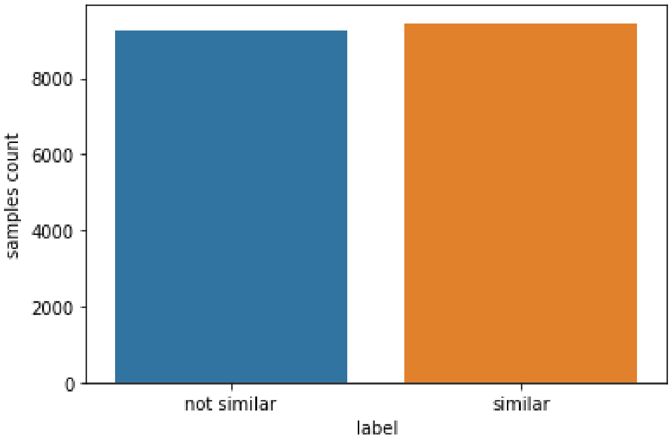

Figure 3. The number of samples in each category.

For creating this dataset, the timelines of each user in the list of the selected singers and politicians have been extracted using official Twitter APIs. Then for each user's timeline, consisting of the most recent, 3240 tweets, in parallel, the set of the audience of the user, primary and advanced features, and the tweets' text are extracted. In the next step, two by two, the similarity measurements of the timelines of these profiles from different explained aspects are calculated, and then this dataset is labeled manually.

Fig 4 represents a comparison of the activity level of three selected profiles. As shown, @User_1, who is Joe Biden tends to post original tweets more than retweeting others; his posts have a higher user engagement by having higher retweet and favorite ratios and tend to post neutral facts. On the other hand, @User_3, Hillary Clinton, a female politician, makes more retweets. On the contrary, @User_2, Jennifer Lopez, a singer, tends to post positive content, mostly her ideas rather than facts. She has a steady behavior in posting tweets and retweets, and her fans also have a constant engagement compared to the other two profiles. Also, both politicians have a higher user engagement on the weekends.

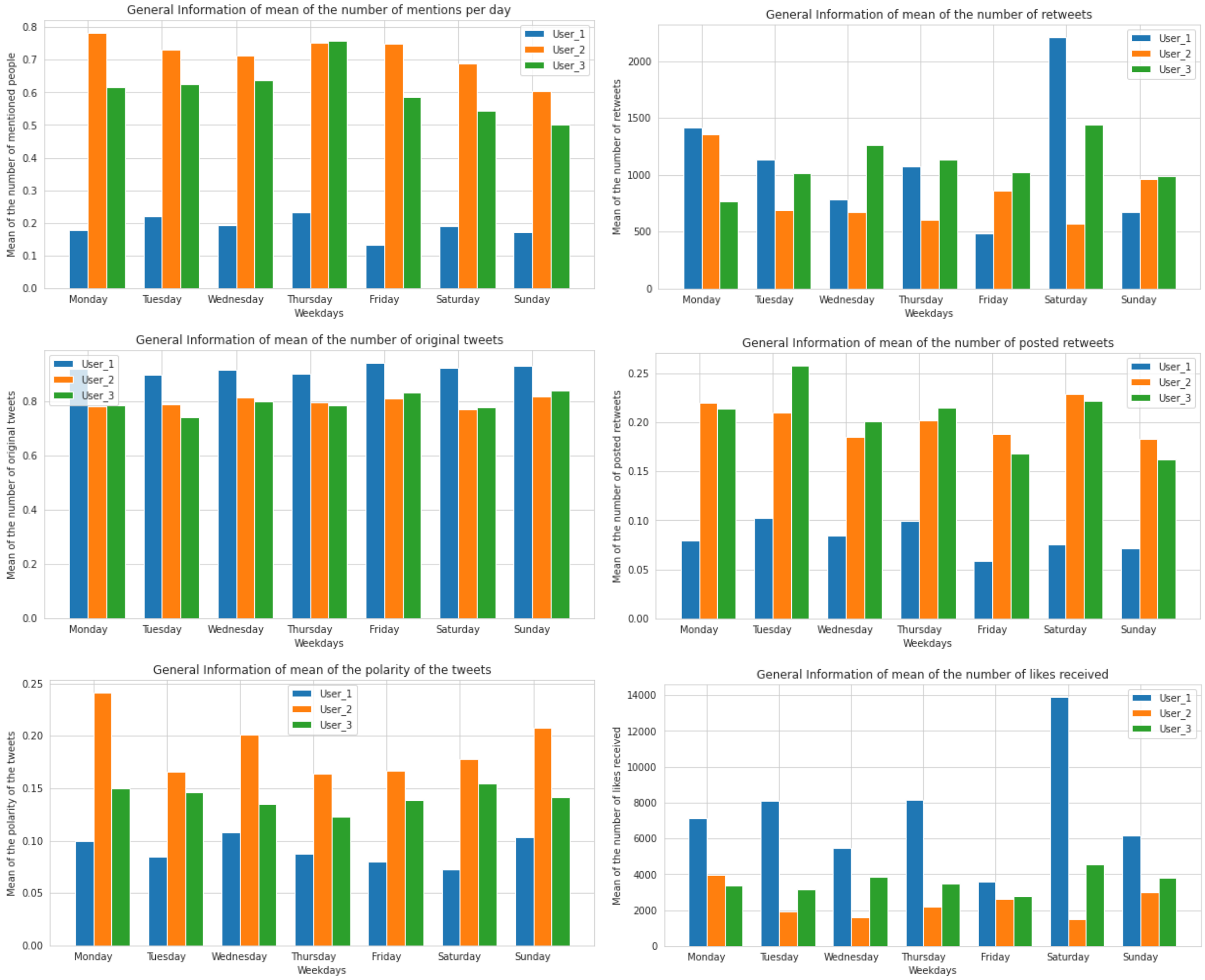

Figure 4. The number of samples in each category.

The dataset has been divided into the training and test datasets using stratified train-test split to select an evenhanded number of samples of each category(i.e., similar, not similar) to keep the train test sets balanced; and then used for training the models, for measuring the performance of the proposed model. Different classification models have been trained using the training dataset and tested on the test dataset. Table 4 represents some of the models' hyperparameters [59,60] used for Randomized search. The best model is achieved by fine-tuning these hyperparameters.

Table 4: The hyperparameters tuned for each models optimization.

Model	Hyper-Parameters	Description
SVM	C	Regularization parameter.
	Kernel	The type of Kernel has been used in the algorithm.
	gamma	Kernel coefficient
	verbose	If True, it verboses the result.
	max_iter	Maximum iteration.
KNN	n_neighbors	The number of neighbors for queries.
	weights	Weight function.
	algorithm	Algorithm of computing the closest neighbours.
	n_jobs	The number of parallel searches in the neighbours.
Random Forest Classifier	n_estimators	The number of trees in the forest
	max_depth	The maximum depth of the trees
	min_samples_split	The least amount of samples for opening a node.
	bootstrap	If it's True, the bootstrap is used to create the trees.
	n_jobs	The number of parallel searches in the neighbours.
	verbose	Control verbosing when it is getting trained and predicting.

After the best models are found, they are tested on the test dataset. Table 5 demonstrates the results and performance of each best model.

Table 5: The evaluation metrics of the classification models; trained on the proposed dataset by employing TF-IDF for text feature extraction.

Model	Classes	Precision	Recall	F1-Score
SVM	**Not Similar**	0.94	0.94	0.95
	Similar	0.96	0.94	0.95
KNN	**Not Similar**	0.90	0.95	0.92
	Similar	0.95	0.82	0.92
Random Forest Classifier	**Not Similar**	0.96	0.98	0.97
	Similar	0.98	0.96	0.97

The results show that the Random Forest Classification has a slightly better performance than the other models and detects whether the users are similar or not, with 97.24 % of accuracy.

5. Conclusion and Future work

Many studies have been done to quantify the behavior of human beings interacting through social media platforms. Still, the question is that how can we compare these profiles with each other? From which aspects can a profile expresses itself? What kind of analysis can be used for comparing these aspects? In this paper, a method has been proposed to calculate the distance of two profiles. The similarity extent of the profiles is measured in three different ways; the behavioral ratios, the graph of the audience, and the contents they post. After extracting the recent 3240 tweets on the timeline of each profile using official Twitter APIs, the data is preprocessed in three ways regarding each aspect of similarity measurement.

First, the behavioral ratios are calculated using Dynamic time warping (DTW), which calculates the distance between time series features. This measurement enables us to understand better the difference between the behavioral ratios of activity, habits, like the number of posts in the day, retweets posted by the user, when the user is more active during the day, etc. Moreover, there is extra information about the ratios of user engagement of the profile because it considers the number of likes and retweets of the profile's audience per day. Next, the user's audience is extracted by defining the relationship between the profiles, which are replies, retweets, quotes, and mentions. In this step, the network of the user's audience interacting with each other is built. Using the Jaccard similarity, the similarity between the two sets of the two selected users is calculated. The results show that the users in the same sector category have more similarities in the audience graph. Finally, in the content similarity measurement, the number of the same tweets is calculated. Then, based on the content, all the tweets are unified into the same language, English. The text is preprocessed by employing natural language processing techniques; tokenization, lemmatization, etc. Then two different vectorization methods are applied, TF-IDF and DistilBERT, for turning the words into their respective vectors. Then, by using cosine similarity, the similarity between two vectors is calculated. The distribution of the calculated similarities presented a similar pattern; therefore, the simplicity and ability to perform the vectorization in a semi-real-time manner is due to the character limitation on posting a Tweet; hence, TF-IDF was chosen and implemented in the model.

In the future, investigation of the role of gender and similarity of users on Twitter is on the plan, also, combining the information from different social media platforms to create a general-purpose social media analytics platform based on the deepint.net platform, supporting all types of data and contains a full suite of artificial intelligence techniques for data analysis, including data classification, clustering, prediction, optimization, and visualization techniques [61] is on the plan. The abilities provided by deepint make it a perfect choice for implementing the proposed model.

Funding: This project has not received any external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Due to the restricted policies of Twitter data publication, this data is confidential, the steps of data extraction from this platform using the official Twitter API have been clearly explained.

Acknowledgments: The project partially supported this research "Computación cuántica, virtualización de red, edge computing y Registro distribuido para la Inteligencia artificial del Futuro," Reference: CCTT3/20/SA/0001, financed by Institute for Business Competitiveness of Castilla y León, and the European Regional Development Fund (FEDER).

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: Not available.

1. Allman-Farinelli, M.; Nour, M. Exploring the role of social support and social media for lifestyle interventions to prevent weight gain with young adults: Focus group findings. *Journal of Human Nutrition and Dietetics* **2021**, *34*, 178–187.
2. Thelwall, M. Word association thematic analysis: A social media text exploration strategy. *Synthesis Lectures on Information Concepts, Retrieval, and Services* **2021**, *13*, i–111.
3. Osorio-Arjona, J.; Horak, J.; Svoboda, R.; García-Ruíz, Y. Social media semantic perceptions on Madrid Metro system: Using Twitter data to link complaints to space. *Sustainable Cities and Society* **2021**, *64*, 102530.

4. Alamsyah, A.; Rahardjo, B.; others. Social network analysis taxonomy based on graph representation. *arXiv preprint arXiv:2102.08888* **2021**.
5. Li, Z.; Zhang, Q.; Du, X.; Ma, Y.; Wang, S. Social media rumor refutation effectiveness: Evaluation, modelling and enhancement. *Information Processing & Management* **2021**, *58*, 102420.
6. Choudhary, A.; Arora, A. Linguistic feature based learning model for fake news detection and classification. *Expert Systems with Applications* **2021**, *169*, 114171.
7. Derhab, A.; Alawwad, R.; Dehwah, K.; Tariq, N.; Khan, F.A.; Al-Muhtadi, J. Tweet-based Bot Detection using Big Data Analytics. *IEEE Access* **2021**.
8. Ayo, F.E.; Folorunso, O.; Ibharalu, F.T.; Osinuga, I.A.; Abayomi-Alli, A. A probabilistic clustering model for hate speech classification in twitter. *Expert Systems with Applications* **2021**, *173*, 114762.
9. Albalawi, R.; Yeap, T.H.; Benyoucef, M. Using topic modeling methods for short-text data: A comparative analysis. *Frontiers in Artificial Intelligence* **2020**, *3*, 42.
10. Dhiman, A.; Toshniwal, D. An Approximate Model for Event Detection From Twitter Data. *IEEE Access* **2020**, *8*, 122168–122184.
11. Wu, W.; Chow, K.P.; Mai, Y.; Zhang, J. Public Opinion Monitoring for Proactive Crime Detection Using Named Entity Recognition. IFIP International Conference on Digital Forensics. Springer, 2020, pp. 203–214.
12. Martyniuk, H.; Kozlovskiy, V.; Lazarenko, S.; Balanyuk, Y. Data Mining Technics and Cyber Hygiene Behaviors in Social Media. *South Florida Journal of Development* **2021**, *2*, 2503–2515.
13. Sushama, C.; Kumar, M.S.; Neelima, P. Privacy and security issues in the future: A social media. *Materials Today: Proceedings* **2021**.
14. Marmo, R. Social media mining. In *Encyclopedia of Organizational Knowledge, Administration, and Technology*; IGI Global, 2021; pp. 2153–2165.
15. Luo, Y. Using tweets to understand how COVID-19–Related health beliefs are affected in the age of social media: Twitter data analysis study. *J Med Internet Res* **2021**, *23*, e26302.
16. Ge, L.; Chen, S. Exact Dynamic Time Warping calculation for weak sparse time series. *Applied Soft Computing* **2020**, *96*, 106631.
17. Roberts, A.; Raffel, C.; Shazeer, N. How Much Knowledge Can You Pack Into the Parameters of a Language Model? *arXiv preprint arXiv:2002.08910* **2020**.
18. Xiao, J.; Zhou, Z. Research Progress of RNN Language Model. 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). IEEE, 2020, pp. 1285–1288.
19. Zhao, J.; Huang, F.; Lv, J.; Duan, Y.; Qin, Z.; Li, G.; Tian, G. Do rnn and lstm have long memory? International Conference on Machine Learning. PMLR, 2020, pp. 11365–11375.
20. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* **2019**.
21. Hernández-Nieves, E.; Parra-Domínguez, J.; Chamoso, P.; Rodríguez-González, S.; Corchado, J.M. A Data Mining and Analysis Platform for Investment Recommendations. *Electronics* **2021**, *10*, 859.
22. Romero, C.; Ventura, S. Educational data mining and learning analytics: An updated survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **2020**, *10*, e1355.
23. Di Minin, E.; Fink, C.; Hausmann, A.; Kremer, J.; Kulkarni, R. How to address data privacy concerns when using social media data in conservation science. *Conservation Biology* **2021**, *35*, 437–446.
24. Rivas, A.; Chamoso, P.; González-Briones, A.; Casado-Vara, R.; Corchado, J.M. Hybrid job offer recommender system in a social network. *Expert Systems* **2019**, *36*, e12416.
25. Akhtar, N.; Ahamad, M.V. Graph tools for social network analysis. In *Research Anthology on Digital Transformation, Organizational Change, and the Impact of Remote Work*; IGI Global, 2021; pp. 485–500.
26. Chen, G.; Shi, X.; Chen, M.; Zhou, L. Text similarity semantic calculation based on deep reinforcement learning. *International Journal of Security and Networks* **2020**, *15*, 59–66.
27. Chicco, D. Siamese neural networks: An overview. *Artificial Neural Networks* **2021**, pp. 73–94.
28. Chandrasekaran, D.; Mago, V. Evolution of Semantic Similarity–A Survey. *arXiv preprint arXiv:2004.13820* **2020**.
29. Mueller, J.; Thyagarajan, A. Siamese recurrent architectures for learning sentence similarity. Proceedings of the AAAI Conference on Artificial Intelligence, 2016, Vol. 30.
30. de Souza, J.V.A.; Oliveira, L.E.S.E.; Gumiel, Y.B.; Carvalho, D.R.; Moro, C.M.C. Exploiting siamese neural networks on short text similarity tasks for multiple domains and languages. International Conference on Computational Processing of the Portuguese Language. Springer, 2020, pp. 357–367.
31. Park, K.; Hong, J.S.; Kim, W. A methodology combining cosine similarity with classifier for text classification. *Applied Artificial Intelligence* **2020**, *34*, 396–411.
32. Chatterjee, M.; others. Detection of Fake and Cloned Profiles in Online Social Networks **2019**.
33. Sowmya, P.; Chatterjee, M. Detection of Fake and Clone accounts in Twitter using Classification and Distance Measure Algorithms. 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2020, pp. 0067–0070.
34. Punkamol, D.; Marukatat, R. Detection of Account Cloning in Online Social Networks. 2020 8th International Electrical Engineering Congress (iEECON). IEEE, 2020, pp. 1–4.
35. Peinelt, N.; Nguyen, D.; Liakata, M. tBERT: Topic models and BERT joining forces for semantic similarity detection. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7047–7055.

36. Dogra, V.; Singh, A.; Verma, S.; Jhanjhi, N.; Talib, M.; others. Analyzing DistilBERT for Sentiment Classification of Banking Financial News. In *Intelligent Computing and Innovation on Data Science*; Springer, 2021; pp. 501–510.
37. Dogra, V.; others. Banking news-events representation and classification with a novel hybrid model using DistilBERT and rule-based features. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* **2021**, *12*, 3039–3054.
38. Vogel, I.; Meghana, M. Profiling Hate Speech Spreaders on Twitter: SVM vs. Bi-LSTM. CLEF, 2021.
39. Haustein, S. Scholarly twitter metrics. In *Springer handbook of science and technology indicators*; Springer, 2019; pp. 729–760.
40. Zahra, K.; Imran, M.; Ostermann, F.O. Automatic identification of eyewitness messages on twitter during disasters. *Information processing & management* **2020**, *57*, 102107.
41. Sheth, J.; Kellstadt, C.H. Next frontiers of research in data driven marketing: Will techniques keep up with data tsunami? *Journal of Business Research* **2021**, *125*, 780–784.
42. Twitter API Documentation | Docs | Twitter Developer.
43. rate limits | docs | twitter developer, url=https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits journal=Twitter, publisher=Twitter.
44. Lahreche, A.; Boucheham, B. A fast and accurate similarity measure for long time series classification based on local extrema and dynamic time warping. *Expert Systems with Applications* **2021**, *168*, 114374.
45. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. KDD workshop. Seattle, WA, USA:, 1994, Vol. 10, pp. 359–370.
46. Gosliga, J.; Gardner, P.; Bull, L.; Dervilis, N.; Worden, K. Foundations of Population-based SHM, Part II: Heterogeneous populations–Graphs, networks, and communities. *Mechanical Systems and Signal Processing* **2021**, *148*, 107144.
47. Vollmer, S. Google Translate. In *Figures of Interpretation*; Multilingual Matters, 2021; pp. 72–75.
48. Wang, C.; Li, M.; Smola, A.J. Language models with transformers. *arXiv preprint arXiv:1904.09408* **2019**.
49. Shaikh, S.; Daudpota, S.M.; Imran, A.S.; Kastrati, Z. Towards Improved Classification Accuracy on Highly Imbalanced Text Dataset Using Deep Neural Language Models. *Applied Sciences* **2021**, *11*, 869.
50. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in neural information processing systems, 2017, pp. 5998–6008.
51. Xiong, R.; Yang, Y.; He, D.; Zheng, K.; Zheng, S.; Xing, C.; Zhang, H.; Lan, Y.; Wang, L.; Liu, T. On Layer Normalization in the Transformer Architecture. Proceedings of the 37th International Conference on Machine Learning; III, H.D.; Singh, A., Eds. PMLR, 2020, Vol. 119, *Proceedings of Machine Learning Research*, pp. 10524–10533.
52. Nozza, D.; Bianchi, F.; Hovy, D. What the [mask]? making sense of language-specific BERT models. *arXiv preprint arXiv:2003.02912* **2020**.
53. Le, N.Q.K.; Ho, Q.T.; Nguyen, T.T.D.; Ou, Y.Y. A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information. *Briefings in Bioinformatics* **2021**.
54. Subba, B.; Gupta, P. A tfidfvectorizer and singular value decomposition based host intrusion detection system framework for detecting anomalous system processes. *Computers & Security* **2021**, *100*, 102084.
55. Qiu, Y.; Yang, B. Research on Micro-blog Text Presentation Model Based on Word2vec and TF-IDF. 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). IEEE, 2021, pp. 47–51.
56. Aljuaid, H.; Iftikhar, R.; Ahmad, S.; Asif, M.; Afzal, M.T. Important citation identification using sentiment analysis of In-text citations. *Telematics and Informatics* **2021**, *56*, 101492.
57. Johansson, F.; Kaati, L.; Shrestha, A. Detecting multiple aliases in social media. 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013). IEEE, 2013, pp. 1004–1011.
58. Goel, A.; Sharma, A.; Wang, D.; Yin, Z. Discovering similar users on twitter. 11th Workshop on Mining and Learning with Graphs. Citeseer, 2013.
59. Agrawal, T. Hyperparameter Optimization Using Scikit-Learn. In *Hyperparameter Optimization in Machine Learning*; Springer, 2021; pp. 31–51.
60. Yang, L.; Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing* **2020**, *415*, 295–316.
61. Corchado, J.M.; Chamoso, P.; Hernández, G.; Gutierrez, A.S.R.; Camacho, A.R.; González-Briones, A.; Pinto-Santos, F.; Goyenechea, E.; Garcia-Retuerta, D.; Alonso-Miguel, M.; others. Deepint. net: A Rapid Deployment Platform for Smart Territories. *Sensors* **2021**, *21*, 236.