2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) | 978-1-6654-9810-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/INISTA55318.2022.9894256

Social Media Post Impact Prediction using Computer Vision and Natural Language Processing

Mihai-Dimitrie Minuț, Diana-Isabela Crainic, Cătălin Sumănaru, Ciprian Daniș, Ioan Sava,
Cristian Simionescu, Adrian Iftene
Faculty of Computer Science
"Alexandru Ioan Cuza" University
Iasi, Romania
{mihai.minut,diana.crainic,catalin.sumanaru,ciprian.danis,ioan.sava,cristian.simionescu,adiftene}@info.uaic.ro

***Abstract*—Millions of people use Twitter every month, which makes it one of the most popular social networks worldwide. Currently, there is an enormous scope market with the potential to be optimized to increase Twitter posts' popularity and engagement. In this paper, we present a method of predicting the number of likes a given post will receive. We introduce a deep learning model and training procedure that uses both computer vision and natural language processing to reach high accuracy when shown new data. Considered use-cases will show the situations in which our system behaves well and the situations in which we do not yet have a solution to improve the current results.**

***Index Terms*—Twitter, deep learning, computer vision, natural language processing**

I. Introduction

The popularity of the Twitter platform is in continuous ascent as more and more people prefer it to create posts on various topics of interest [1], [2]. On one hand, due to the application's content, it is a valuable social media marketing tool for business and a huge opportunity for brands [1]. On the other hand, influencers, who are creators with a large number of followers that share content on social networks, can build communities around topics and niches [3]–[5]. They use Twitter as a visual platform to make regular posts and generate large followings of engaged people who pay close attention to their views. For that reason, there should exist a special connection between influencers and brands [6], [7]. Influencers are aware of what kind of posts the followers want to consume, therefore, their primary focus is on attracting the audience with useful content and increasing brand prestige [8], [9]. Commonly, Twitter influencers get paid by brands to make the promotion of a product and service. Currently, there is an enormous scope market with the potential to be optimized to increase Twitter posts' popularity and engagement [10], [11].

In this context, this paper proposes a social media post impact predictor. Given the desired image, the neural network must predict the expected number of "likes" or reactions that the post will generate. We extended the application to also use the description to predict the impact of a post with greater accuracy.

The rest of the paper is organized as follows: section 2 describes the state of the art, section 3 presents the solution, section 4 evaluates the proposed solution and, in the end, conclusions were drawn.

II. State of the art

The task of classification through the means of Neural Networks has been widely studied for a long time. In the article [12], the author presented a task of classifying e-commerce products given a dataset consisting of the image of the products, their descriptive text, and their title (label). The author proceeds to conduct several experiments on classifying the products based solely on the image, by using Convolutional Neural Networks [13] or solely based on descriptive text, by using Natural Language Processing Models of the Bag-of-Words [14] type and then compare them to models which take into consideration hybrid approaches which take as input both the image and text. The author shows in the end that better results are given by the hybrid model types.

In the paper [15], the authors presented the problem of detecting if a social media post is offensive. The data originated from a dataset built from social media posts created on Twitter[1]. Within the paper, the authors describe how they used the post's text tokens and put them through a hybrid neural network, consisting of both a Natural Language Processing BERT [16] model and a Convolutional Neural Network to obtain the needed predictions. The results have shown that the hybrid solutions obtained better accuracy when comparing them to the ones consisting only in an NLP model.

Exploiting the data on Twitter is topical because we can find out the feelings of those who post about a certain event [17], we can identify fake news [18], [19], or we can create language tools that are based on this data [20], [21].

III. Proposed Solution

A. Application

A micro-services-based architecture was chosen for the development of the application [22]. The proposed solution consisted of three core micro-services: (1) Account Management Service, (2) Prediction Service, and (3) Predictions Management Service (see Fig. 1).

The responsibilities of each micro-service were as follows:

[1]https://twitter.com/

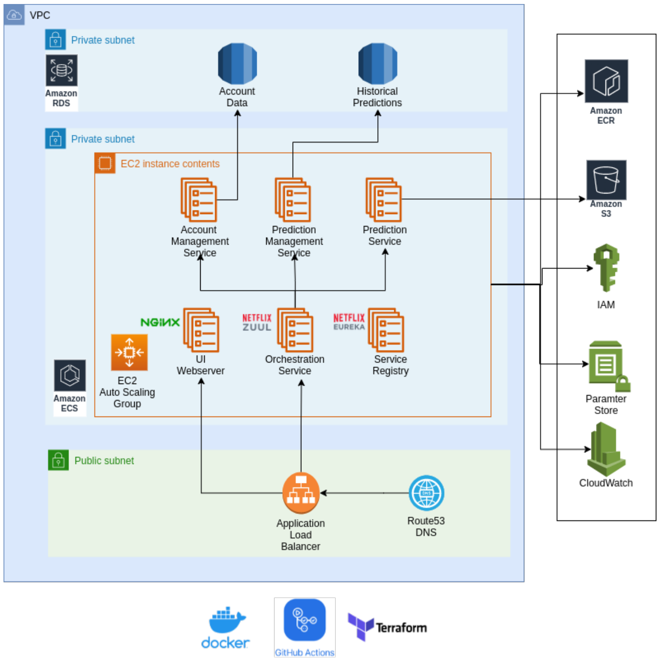

Fig. 1. The architecture of the application

1) Account Management Service: Account Management Service was responsible for handling the registration, authentication, and interaction with the user's Twitter account via Twitter's official API. When a user wanted to log in to the application, he would be redirected to the Twitter OAuth[2] journey. Once the authentication flow was completed, the said user would be redirected back to the main application, where the Twitter-provided credentials would be exchanged for a custom JSON Web Token to enable both authentication and authorization in future requests. The service was developed using Spring Framework[3] and it used an Amazon RDS database[4].

2) Prediction Service: The Prediction Service was tasked with performing social media post impact predictions for a given photo coupled with a description provided by the user via a form. The service was developed using Python[5] and the Flask Framework[6].

3) Prediction Management Service: The Prediction Management Service was responsible for handling historical predictions. It enabled the end-user to view, delete, and post on Twitter the previously processed content. Similar to its first counterpart, this service was developed using the Spring Framework along with an Amazon RDS database.

B. Architecture and Integration

The system was built on top of Amazon Web Services[7] cloud infrastructure. An additional micro-service, named Orchestration Service, was employed to both mediate complex interactions between the previously specified services and act as an API Gateway for proxying the designated endpoints, via Netflix Zuul[8]. Each service featured client-side load-balancing using Ribbon[9] coupled with Eureka[10] for service-discovery. All of these services were hosted on Elastic Container Service, were exposed via a single external load-balancer, and featured distributed log management and configuration via CloudWatch and SSM Parameter Store[11].

C. Dataset & Model

The Prediction Service used a hybrid model ensemble composed of two sub-models: (1) an NLP model for processing the description text of the tweet along with (2) a Computer Vision model for the image. The final prediction result was obtained by averaging both previous results.

The dataset was built by leveraging Twitter API queries for the following categories of tweets: *Travel, Movies, Animals and Habitats, Food, Weather, Sports, Video Games, Musical Instruments, Technology, Medicine, Crypto, Fashion, Cars, Nature&Greening&Anti-Pollution, Anime&Cartoons, Manga &Books&Magazines, Fitness, Careers, Arts, Politics*. The queries were made daily due to the limitations imposed on recent Tweets, and tweets that were published in the past 7 days. Each query result included the following properties: *tweet_id, tweet_text, tweet_image, tweet_impact, tweet_source, user_id, user_followers, tweet_posted_date, user_tweets_count, retweeted*, properties which were considered useful for providing the expected impact of a Twitter post. The final dataset consisted of a total of 120,000 tweets. The NLP model required text-sanitization, which consisted in removing urls and stripping unnecessary spaces and characters. The pre-processing steps for the computer vision model also included image resizing and normalization.

The training process involved fine-tuning two pretrained models: *bert-base-uncased* version for the NLP task and Resnet18 for the computer vision one, see Fig. 2[12] [23] and Fig. 3[13] [24] depict their architectures, respectively. Their outputs, more exactly the last layers of *Softmax*, were combined into a decoder and the final result represented an array of length equal to the number of classes. In the end an *Argmax* was applied on this array to get the final prediction value.

Once the hybrid model was built, it was exported in *h5* format to be used within the prediction service. Once the prediction service started it detached a thread to initialize the model in the background, making calls to it, which directly needed the model, to be unavailable and return an error. Only

[2]https://developer.twitter.com/en/docs/authentication/oauth-1-0a/obtaining-user-access-tokens
[3]https://spring.io/projects/spring-framework
[4]https://aws.amazon.com/rds/
[5]https://www.python.org/
[6]https://flask.palletsprojects.com/en/2.1.x/
[7]https://aws.amazon.com/
[8]https://github.com/Netflix/zuul
[9]https://cloud.spring.io/spring-cloud-netflix/1.4.x/multi/multi_spring-cloud-ribbon.html
[10]https://spring.io/guides/gs/service-registration-and-discovery/
[11]https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html
[12]https://www.researchgate.net/figure/The-architecture-of-the-Fine-tuned-BERT-base-classifier_fig3_351392408
[13]https://www.researchgate.net/figure/Original-ResNet-18-Architecture_fig1_336642248

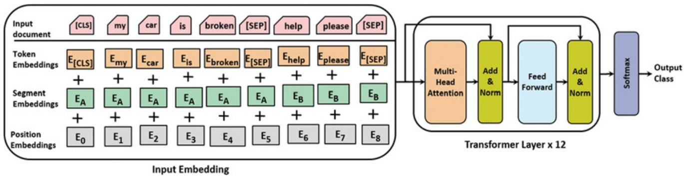

Fig. 2. The architecture of the Fine tuned BERT base classifier [23]

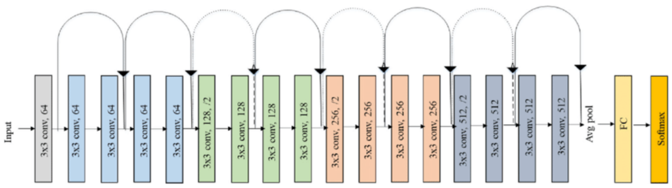

Fig. 3. The architecture of the ResNet 18 model [24]

after the background task of initialization was complete, those methods, such as getting a prediction for an image and text, would be available and exposed as part of the API.

In order to load the model, the prediction service's background task called for storage within the cloud cluster to get the model's binary data loaded locally. Configuration of the hybrid model was achieved by first importing the BERT model structure, BERT tokenizer and the Resnet18 model structure, after which the binary data downloaded before could be used to initialize the architecture. Successfully initializing the hybrid model architecture meant that the prediction service could therefore expose the API endpoints which offered access to the model predictions.

IV. Use Cases

The simple way to utilize the application is as it was exemplified before: (1) login into Twitter, and (2) then, by providing an image and text, for an upcoming tweet, (3) get a prediction on how much impact the posting will generate on the account (see Fig. 4).

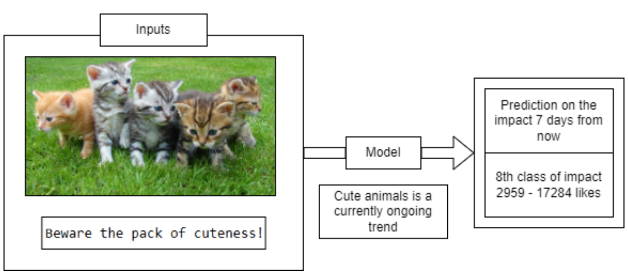

Fig. 4. Standard and correct use case for prediction

By doing this, the user can decide if the tweet is worth posting by taking into account some of his previous impact on the platform.

A. Use Cases for our Application

The process represented by Fig. 5 explains how somebody could obtain a prediction within the application. Before submitting any input the user must log in with his Twitter account; this step is necessary for posting a tweet or obtaining other user metadata, such as the number of user's followers, which might help the prediction. After getting the account setup, the user can get a prediction by firstly uploading an image and a text for the tweet, and then by pressing the "Predict" button. Then the app will retrieve a prediction from the model, which is contained within the prediction service. Since the impact interval implied by the class of prediction would most likely make the user experience much worse, it was decided that the center of the interval should be shown in the front-end design.

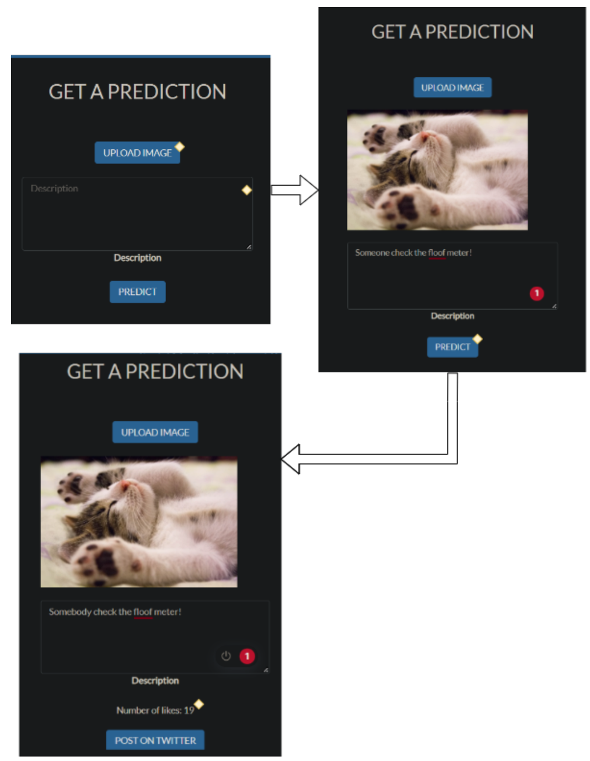

Fig. 5. Generating a prediction within the application

B. Negative Use Cases

Since the model is built and updated with tweets scrapped from current popular category searches, one bad example of usage would be predicting the impact of a tweet that is not contained within the platform's current trends (see Fig. 6). The predictions for very peculiar tweets could be met with strange predictions and should not be trusted since the model behavior on unknown types of features could be undefined.

Predictions made at different times might be incomparable due to the varying trends on the platform which directly impact the model, and in turn, its predictions. If the time gap between the predictions is too large (see Fig. 7), one should completely avoid making any assumptions and/or comparisons between the impact of those tweets. If one would like to still compare the current impact, then both of the postings need to have a recently updated impact prediction.

As it is exemplified within Fig. 7 the prediction on the post about gold is made in 2018 but the one made on the tweet

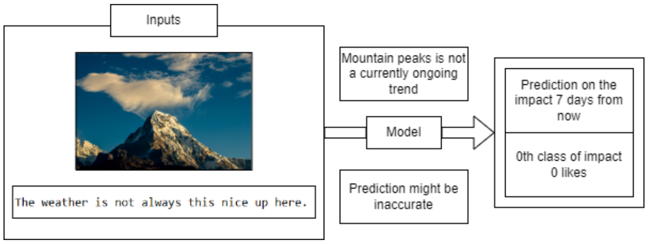

Fig. 6. Standard but incorrect use case for prediction

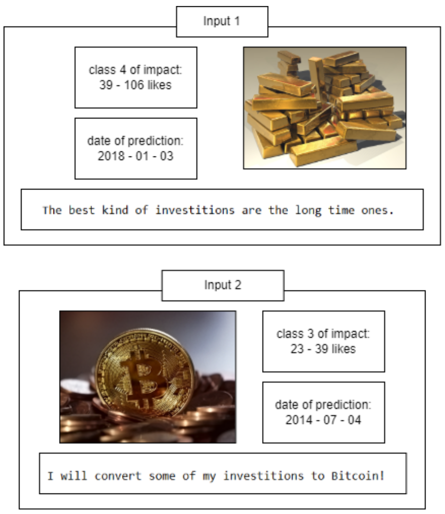

Fig. 7. Bad comparison between predictions

with a cryptocurrency is made in 2014. Since the dates differ by a large number of years, the popularity of the two types of currencies has fluctuated and changed, and, as a result, it is safe to assume that no valid comparison between the predictions can be made, and this kind of use case is incorrect and misusing the application.

C. Positive Use Cases

Good use of the application would be to find the best combinations of images and texts for a tweet. By having multiple photos and text options (see Fig. 8) and utilizing the model, the user can find the best combination for an upcoming posting.

As one can see in Fig. 8, the selection process starts with 3 animal images and 2 related pieces of text, and, by taking into account each combination's results, the application can resolve which pair of images and text results in the most impact.

D. Discussions

As we saw in the examples above, the system is often able to predict a good combination of text and image that is likely to have the greatest impact on the network. Problems arise when trying to make posts with new content that does not have the corresponding text or image on Twitter. In such situations, it is difficult for us to estimate what the effect of these posts will be and we need to find solutions in the future, possibly with information from outside the Twitter network.

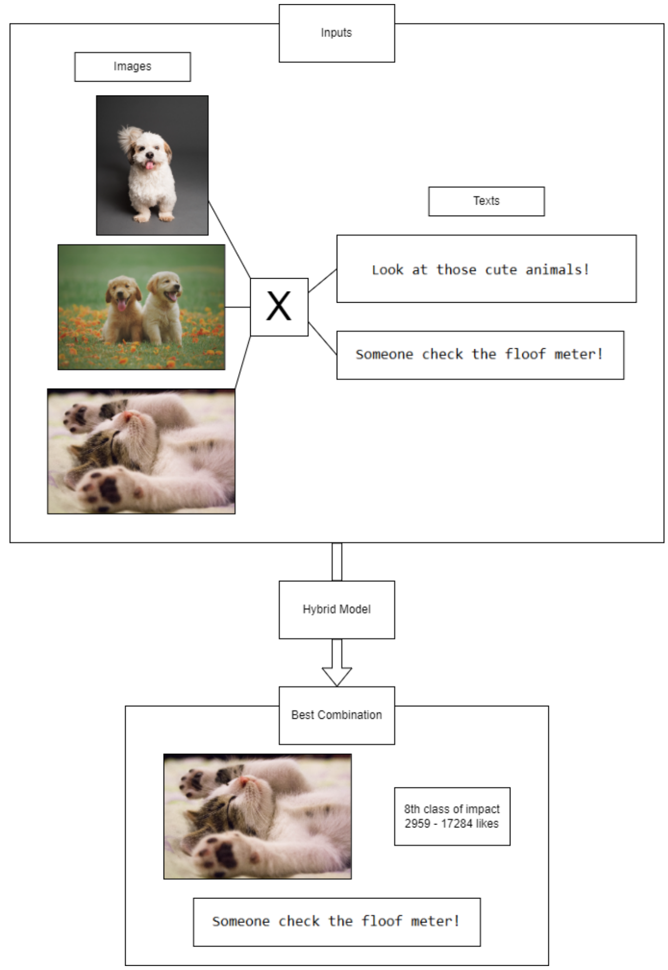

Fig. 8. Finding the best combination of image and text for a tweet

V. Results and Evaluation

A. Tweets Dataset

The final version of the dataset consists in a total amount of 120,190 tweets, each having the tweet's text, the RGB representation for the image and additional metadata such as the number of likes or user followers. The initial set of statistics for the images is:

- Global average width: 947.678,
- Global average height: 675.377,
- Global median width: 1,024,
- Global median height: 527,
- Global std width: 230.186,
- Global std height: 277.905,
- Global width variance: 52,985.424,
- Global height variance: 77,231.202.

After examining the statistics the standard for the image's dimensions was decided to be $366 \times 512 \times 3$, and all of the images were resized, by keeping their original width to height ratio, to fit the new standard with additional white padding where necessary.

After a thorough data mining iteration on the metadata associated with the features, some insights were discovered:

- Under 20% of the tweets have pure duplicate text;
- Under 1% of the tweets are duplicates in image and text;
- Around 33% of the dataset's tweets have 0 impact;
- There is under 1% correlation between impact and the user's number of followers;
- There is a big correlation, around 43%, between the number of tweets and the number of followers that a user has;
- The length of a tweet's text is at most 25 words.

The label for the dataset, the respective target for predicting the social impact of a tweet, was chosen to be the number of likes for the respective posting. The amount of impact generated by a tweet from the dataset ranged from 0 to 1.2 million, and the distribution was very skewed, with many tweets concentrating on the lower part of the spectrum. Due to this imbalance, a binning approach was considered and the bins were formed from the 10 quantile distribution on the impact value for the whole dataset. Thus the classes to represent the labels were chosen as follows:

- Class 0: impact: [0, 7);
- Class 1: impact: [7, 15);
- Class 2: impact: [15, 23);
- Class 3: impact: [23, 39);
- Class 4: impact: [39, 106);
- Class 5: impact: [106, 441);
- Class 6: impact: [441, 830);
- Class 7: impact: [830, 2,959);
- Class 8: impact: [2,959, 17,284);
- Class 9: impact: [17,284, 1,212,284).

When comparing the dataset built for the task to other similar datasets, for the example the one from the Natural Disaster NLP Competition[14] published on Kaggle[15], one can state that some of the positive distinctions amount to having a much larger and broader dataset with many additional features and metadata to work with. Those additional dimensions given to data can make a big difference in the final model quality but they also require much more exploratory analysis and more tests to be able to state the overall dataset quality. The overall time investment for producing even a simple model is increased by having a bigger dataset with hybrid features due to the additional preprocessing steps and training time.

B. Models

The training environment for the models consisted in the python notebooks and because of the big dimensions of the resized images the model could be trained on 10% of the dataset at a time and only for short amounts of time, around 10-20 epochs. The training process switches and preloads other batches of 10% the training data so the whole training dataset is used. The 120,000 tweets were separated (by respecting label distribution) in 100,000 tweets for training and 20,000 for testing.

The models achieved on average 10% accuracy scores on the test data at the end of training. The results show that the testing accuracy is pretty low, making the hybrid model build comparable to taking a random guess. Having some imposed restrictions on the training process most likely negatively affected the end results for the model. Further model generations should be trained and tested on other variations of the dataset featuring lower feature dimensions and as well as more complex techniques in terms of binding the hybrid model layers together.

The hybrid model build upon an NLP BERT model and a CNN image model presented at SemEval [25], which also was used to analyze social media postings, seemed to perform better than its two individual counterparts on the testing experiments, but only after multiple model generations which helped in refining the model training process. One key difference is that the authors faced a binary classification task in order to determine offensiveness, which in the case of measuring the impact does not make much sense by comparison.

As a conclusion, the baseline hybrid model built served as a building block for further experiments and tests and certainly rose the understanding of the dataset and hybrid model training.

C. Comparisons with Similar Applications

In terms of the whole system behind the application, a similar product would be LikelyAI[16] which is another application that aims to predict the popularity of social media posts. The project was developed to mainly focus on social media posts coming from the Instagram[17] platform.

Unlike the approach described so far, which tried to predict how much a social media post will have, LikelyAI makes a prediction by receiving several photos and picking the ones that will generate more likes, comments, and overall Instagram engagement. The application provides the users with insights on photo's emotions, objects, colors and other metrics. The LikelyAI was developed to be used as a web application but also as an Android and iOS application, making it easier to access across devices.

A notable resemblance between the solutions is represented by the integration with the social media platforms for which they are tailored. This provides the user with an easy solution for predictions but also posting on the social media platforms directly through the application.

VI. Conclusions

After performing all the proposed tasks over the dataset, which consists of approximately 120,000 tweets, built from scratch, was obtained exciting information about the number of likes predicted in Twitter posts. There were performed examinations regarding Twitter post data features and perform a significant exploratory data analysis. For instance, there

[14]https://www.kaggle.com/competitions/nlp-getting-started
[15]https://www.kaggle.com/
[16]https://www.likelyai.com/
[17]https://www.instagram.com

exists a big correlation between the number of tweets and the number of followers that a user has.

The chosen model is a hybrid one, working particularly well with images and text data, especially with the number of samples in the dataset. The trained model achieved well on the test dataset, given the difficulty of the task and the amount of data available.

The application was built using a scalable, microservice-based architecture in which each microservice has a well-defined role. Although there is still room for improvement, the current infrastructure is quite solid.

ACKNOWLEDGMENT

Data processing and analysis in this paper were supported by the Competitiveness Operational Program Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

REFERENCES

[1] S. B. M. Bulearca, "Twitter: A viable marketing tool for smes," *Global Business and Management Research: An International Journal*, vol. 2, no. 2, pp. 296–309, 2010.

[2] P. Yaisawas, S. Lerdsri, B. Thanasopon, and P. Netisopakul, "Business popularity analysis from twitter," *International Conference on Computing and Information Technolog*, vol. 566, p. 337–348, 2017.

[3] E. Bakshy, J. Hofman, W. Mason, and D. Watts, "Everyone's an influencer: Quantifying influence on twitter," *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011*, 2011.

[4] V. Nebot, F. M. R. Pardo, R. Berlanga, and P. Rosso, "Identifying and classifying influencers in twitter only with textual information," *Natural Language Processing and Information Systems*, 2018.

[5] Z. Z. Alp and Şule Gündüz Öğüdücü, "Identifying topical influencers on twitter based on user behavior and network topology," *Knowledge-Based Systems*, vol. 141, pp. 211–221, 2017.

[6] S.-A. A. Jin and J. Phua, "Following celebrities' tweets about brands: The impact of twitter-based electronic word-of-mouth on consumers' source credibility perception, buying intention, and social identification with celebrities," *Journal of Advertising*, vol. 43, pp. 181–195, 2014.

[7] J. Bokunewicz and J. Shulman, "Influencer identification in twitter networks of destination marketing organizations," *Journal of Hospitality and Tourism Technology*, vol. 8(1), 2017.

[8] R. G. Bilro, S. M. C. Loureiro, and J. F. dos Santos, "Masstige strategies on social media: The influence on sentiments and attitude toward the brand," *International Journal of Consumer Studies*, 2021.

[9] S. Ibáñez-Sánchez, M. Flavián, L. V. Casaló, and D. Belanche, "Influencers and brands successful collaborations: A mutual reinforcement to promote products and services on social media," *Journal of Marketing Communications*, pp. 1–18, 2021.

[10] V. Wadhwa, E. Latimer, K. Chatterjee, J. McCarty, and R. Fitzgerald, "Maximizing the tweet engagement rate in academia: Analysis of the ajnr twitter feed," *American Journal of Neuroradiology*, vol. 38 (10), pp. 1866–1868, 2017.

[11] W. Read, N. Robertson, L. McQuilken, and A. S. Ferdous, "Consumer engagement on twitter: perceptions of the brand matter," *European Journal of Marketing*, vol. 53(5), 2019.

[12] C. Bonnet, "Classifying e-commerce products based on images and text," *Insight*, 2017.

[13] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, J. Cai, and T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition Journal*, vol. 77, 2018.

[14] Y. Zhang, R. Jin, and Z. Zhou, "Understanding bag-of-words model: A statistical framework," *International Journal of Machine Learning and Cybernetics*, vol. 1(1), pp. 43–52, 2010.

[15] A. Safaya, M. Abdullatif, and D. Yuret, "Bert-cnn for offensive speech identification in social media," *KUISAIL at SemEval-2020 Task 12*, 2020.

[16] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL-HLT*, vol. 1, pp. 4171–4186, 2019.

[17] A. Iftene, M. S. Dudu, and A. R. Miron, "Scalable system for opinion mining on twitter data. dynamic visualization for data related to refugees' crisis and to terrorist attacks," *26th International Conference on Information Systems Development. Larnaca, Cyprus, September 6-8*, 2017.

[18] C. S. Atodiresei, A. Tănăselea, and A. Iftene, "Identifying fake news and fake users on twitter," *Proceedings of International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2018, 3-5 September 2018, Belgrade, Serbia. Procedia Computer Science*, vol. 126, pp. 451–461, 2018.

[19] C. G. Cușmuliuc, L. G. Coca, and A. Iftene, "Identifying fake news on twitter using naive bayes, svm and random forest distributed algorithms," *Proceedings of The 13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018)*, pp. 177–188, 2018.

[20] A. Iftene, D. Gîfu, A. R. Miron, and M. S. Dudu, "A real-time system for credibility on twitter," *Proceedings of The 12th Language Resources and Evaluation Conference (LREC 2020), Marseille, France, May 2020*, pp. 6168–6175, 2020.

[21] C. Florean, O. Bejenaru, E. Apostol, O. Ciobanu, A. Iftene, and D. Trandabăț, "Sentimentalitsts at semeval-2016 task 4: building a twitter sentiment analyzer in your backyard," *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), Association for Computational Linguistics. San Diego, California. June, 2016*, pp. 243–246, 2016.

[22] M. Baboi, A. Iftene, and D. Gîfu, "Dynamic microservices to create scalable and fault tolerance architecture," *Procedia Computer Science*, vol. 159, pp. 1035–1044, 2019.

[23] S. Khatoon, M. A. Alshamari, A. Asif, M. M. Hasan, S. Abdou, K. M. Elsayed, and M. Rashwan, "Development of social media analytics system for emergency event detection and crisis management," *Computers, Materials & Continua*, vol. 68(3), pp. 3079–3100, 2021.

[24] F. Ramzan, M. U. G. Khan, A. Rehmat, S. Iqbal, T. Saba, A. Rehman, and Z. Mehmood, "A deep learning approach for automated diagnosis and multi-class classification of alzheimer's disease stages using resting-state fmri and residual neural networks," *Journal of Medical Systems*, vol. 44(2), 2020.

[25] M. A. Ali Safaya and D. Yuret, "Bert-cnn for offensive speech identification in social media," *UISAIL at SemEval*, vol. 12, 2020.