Fake News Detection Related to the COVID-19 in Slovak Language Using Deep Learning Methods

Martin Sarnovský, Viera Maslej-Krešňáková, Klaudia Ivancová

Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 04001 Košice, Slovakia. E-mail: martin.sarnovsky@tuke.sk, viera.maslej.kresnakova@tuke.sk, klaudia.ivancova@student.tuke.sk

Abstract: One of the biggest problems nowadays in the online environment is the spreading of misinformation. Especially during a global pandemic, the most popular topics of fake news are related to coronavirus. Therefore, automatic detection of such news in the online media or social networks can help with the prevention of misinformation spreading. During the recent years, deep learning models proved to be very efficient in this task. However, the majority of the research focuses on the training of these models using publicly available data collections, mostly containing news articles written in the English language. As the spreading of fake news is a global phenomenon, it is also necessary to explore these approaches on the various local data sources. The work presented in this paper focuses on using the deep learning models for the automatic detection of fake news written in the Slovak language. We collected the data from multiple local online news sources related to the COVID-19 pandemic and used it to train and evaluate the various deep learning models. Thanks to the combination of bidirectional long-short-term memory network with one-dimensional convolutional layers, we achieved an average macro F1 score on an independent test set of 94%.

Keywords: natural language processing; deep learning; convolutional neural networks; fake news; COVID-19

1 Introduction

In recent years, people are spending more and more of their lives online and on social media. There are many advantages and disadvantages in moving human activity and communication to the online environment. The ability to read, share and publish information for all equally is considered the most significant benefit. Exchanging information via the Internet will take much less time, money, and effort. Despite the fact that the quality of news appearing on social media is much lower than in traditional and verified sources, thanks to the mentioned advantages,

people tend to search for such news more and more. While the vast majority of content on the Internet is user-generated content, the quality of the content varies. The majority of the users try to produce true and decent content; however, there are also users who create misleading content in the online environment.

Authors in [1] define anti-social behavior as behavior that violates the fundamental rights of others. It is characterized by repeated violations of social rules, defiance of authority, and the rights of others. There are several types of anti-social behavior in the online environment. One specific type is related to the production and spreading of hoaxes, fake news, and false reviews. They are purposefully created for the dissemination of misinformation, concealment of true information, misleading the reader, or false beliefs of readers. Their occurrence tends to increase during public events, disasters, or when evaluating new products. In the work presented in this paper, we focused on the detection of fake news. Fake news can be defined [2] as news articles that are intentionally and demonstrably untrue and misleading their readers. There are two basic properties of fake news related to this definition, namely the authenticity of the message, according to which we can retrospectively verify false information and the intention under which fake news is created. As already mentioned, most often, such reports are created for misleading consumers and generally for dishonest intent. The authors in [3] supplemented this definition with the following terms, which they do not consider to be false in this respect:

- Satirical news - contain appropriate content that is not intended to mislead or mislead consumers

- Rumors - which do not come from news

- Conspiracy theories - which are sometimes difficult to describe as real or false

- Misinformation - which is created unintentionally

- Hoaxes - which are created to entertain or deceive an individual

Fake news is not a new problem, as the type of traditional media has changed from print newspapers to radio and television to online news, so has fake news. However, social media and the increasing use and living through the online environment have helped make this problem a major one. From a psychological point of view, spreading misinformation is very easy because people naturally cannot distinguish very well between true and false news. Traditional fake news is aimed mainly at consumers, where their vulnerability factors can be taken into account. Also, many users spread fake news and trust them to gain power or acceptance by others in the community or even to satisfy societal views and needs. Fake news is usually spreading on social media through specific accounts. Due to their low cost and rapid dissemination of messages, they are widely used today. These include social robots, trolls, and cyborgs. Such social robots are created to share content and interact with people in the online environment, primarily using

computer algorithms. They are designed specifically for the purpose of manipulating and spreading false messages. For example, during the 2016 US presidential election, more than 19% of online conversation were found to have been manipulated presidential election results information in terms of spreading false news and disrupting online communities [4].

The solution how to prevent the massive spread of anti-social behavior and false news in the online environment involves the creation of tools for early detection and elimination of such information. Today, manual techniques of detection of such news are being employed (e.g., human moderators, etc.), but they are insufficient as the number of information increases. To eliminate the impact of fake news that would achieve the desired results, it is necessary to create automated tools for their detection. More recently, such approaches can be helpful more than ever, especially during the COVID-19 pandemic. Misinformation related to the pandemic and vaccination are spreading through social media rapidly, and similar tools able to detect them may be helpful in the prevention of their reach.

Moreover, as the pandemic is a global phenomenon, much of the misinformation spreads locally, in different countries, via various local web portals. Therefore, it may be interesting to train the models using the data in particular languages. In this paper, we focused on training the deep neural network models able to detect fake news from the news articles related to COVID-19 written in the Slovak language. We decided to use the different architectures of deep learning models, as they proved very efficient when applied to the related problems (e.g., fake news detection in various domains and languages). In our research, we focused on using only the textual content of the articles to capture the linguistic features that distinguish the regular articles from the misinformation.

The presented paper is organized as follows: Section 2 is dedicated to the description of existing approaches using neural networks for fake news detection. The following section summarizes the data collection and pre-processing steps. Then, in Section 4, we describe models training and evaluation of the experiments.

2   Related Work

Fake news detection is usually considered a binary classification task, where the models predict, based on the content, if a particular news piece contains fake news or not. Neural networks are among the most frequently used methods in the area of automatic detection of fake news from text. Besides the modeling, many works focus on the text pre-processing and appropriate representation. For example, authors in [3] claim that when working with traditional news sources, it is sufficient to work only with the content of the news piece. On the other hand,

when working with the social media posts (or discussion forums or similar sources), information related to the source, attachments (e.g., pictures, videos) may be useful as well. The title of the news article is usually important, as fake news often uses strong or outrageous content, so-called *clickbait*, which forces the user to click on the article and read it. Therefore, it is necessary to explore the linguistic aspects and, besides them, also explore the information related to the authors of the news, including reactions of the readers [5]. Authors in [6] compared neural network models trained using full texts from the articles and just the title text. When comparing the evaluation metrics on the full-text data to title texts, the models still managed to perform on a similar level. One of the reasons may be using of clickbait in the title texts.

Different network architectures have been used to tackle the automatic fake news detection from the text. Convolutional neural networks were used in [7] for the detection of fake news in texts containing political statements. Authors also used the metadata, describing the authors' info (e.g., occupation) or information related to other author's statements. Authors randomly initialized an embedding vector matrix to encode the data and metadata and used the convolutional layer of the neural network to capture the dependence between the metadata vectors. Next, he performed a maximum association operation in the latent space, followed by the LSTM layer of the recurrent neural network. Finally, the author combined the representations of the texts with the metadata representation from LSTM and brought them into a fully connected layer to generate the final prediction. The Word2vec tool was used to create the embedding.

In [8], the authors also aimed to detect fake news using the Capsule neural network. They used this model to improve classical CNN and RNN by adding specific properties to each source and destination node. The model created by the authors is used to identify fake news articles with different lengths. Depending on the size of the pieces, the authors used two different architectures. The model uses pre-trained vectors to initialize learning. For the short texts, the authors developed a structure whose layers are identical to those in the first model, but only two parallel networks are considered. In this model, static word embedding is used, which represents pre-trained vectors that are kept static during training, and only other parameters are trained. The model containing medium and long texts achieved the best accuracy using a non-static word embedding 99.8%. The model containing short texts was still evaluated using metadata because it was more difficult to detect false messages.

In [9], the authors used a dataset containing reports from the period of the US presidential election in 2016. They used a deep neural network as a model and solved the problem of binary classification. The first layer in the neural network consists of pre-trained word embedding using Word2vec. Embedding is used as an input to a convolution layer with 128 filters and a window size of 3. For evaluation, the authors used an accuracy metric where they were able to achieve 93.5% accuracy. The authors in [10] also used content from the US

presidential election in 2016 - toxic comments from Twitter. They provide an overview of various pre-processing options, standard deep learning models, and popular transformers models.

In [11], the authors worked on designing a deep convolutional neural network to detect fake news. They developed the model so that the functions learn to automatically distinguish the elements for classifying fake news through several hidden layers built into the neural network. The authors used the uncontrolled GloVe learning algorithm as an embedding model. The model was followed by three parallel convolution layers, a maximum common layer, and finally, an output layer based on prediction. They also used a single flatten layer, which converts elements taken from a common layer and maps them to a separate column, which is then moved to a fully connected layer. The authors used ReLU as the activation function, the primary function of which is to remove negative values from the activation map by setting it to zero. By evaluating the model, they managed to obtain an accuracy of 98.36%.

When considering the detection of fake news in the Slovak language, the work [12] is the only one that explores the dataset of texts in the Slovak language. The aim was to explore different approaches to detecting fake news based on morphological analysis. The authors have created their own data set, which contains articles in the Slovak language from the local news sources. The authors used articles containing the keywords "NATO" and "Russia". These were classified into two specific classes according to the publisher's source using the web portal *www.konspiratori.sk*. Since the Slovak language has complex rules for declension, the authors have decided that the use of morphologically annotated corpora from the Slovak National Corpus will contribute to automated morphological analysis. The morphological analysis was applied to all articles in the dataset, and each word was assigned a set of morphological tags. Contrary to other works, the authors did not use the neural network but used the decision tree model. They divided the analyzed data set in the ratio 70:30 and set the different depths of the tree with the model using entropy. With the decision tree model, they achieved an accuracy of 75%.

3   Data Collection and Pre-Processing

In the work presented in this paper, we focused on detecting fake news in Slovak online space. To obtain the data from various local online newspapers, we used the MonAnt platform [13]. We used the platform to collect the news articles related to COVID-19 pandemic from mainstream local newspapers, as well as from unreliable sources, often publishing conspiratory content. In general, we focused on covering different types of sources to be able to collect both regular news articles as well as misleading pieces. In the MonAnt platform, we created

connectors to the web news portals *Aktuality, Hospodárske noviny, TA3, Hnonline, Slobodný Vysielač, Zem & Vek, Magazín pán občan, Hlavné správy, Proti prúdu, Rady nad zlato* etc. and filtered the articles containing the selected keywords: *Covid, Covid-19, Coronavirus, SARS-CoV-2.*

To train the models using such data, we needed to obtain the class label. At first, the target feature was derived according to the *www.konspiratori.sk* database. The database is a result of the local media experts aiming to monitor and reporting of fake news in different local media and contains a ranking of many regional online news portals and their respective "score", representing the probability of publishing misleading information and fake news in their sites. We used the score for initial labeling and derived the binary target feature according to the trustworthiness of the sources, separating the reliable sources (with very low scores) from the suspicious ones (the highest scores). However, such labeling only considered the credibility of the whole source (e.g., website or newspaper) but not the credibility of individual reports. Such an approach may lead to incorrect labeling, as many conspiratory websites also publish regular news. We've concluded that the best way to assign relevance to articles correctly is to manually re-label the content from the conspiratory websites and correct the labels for the articles, which contain standard information (e.g., re-published news from the press agencies, etc.). After such correction, the dataset consisted of 12,885 documents containing regular news and 851 articles labeled as fake news. The target attribute was heavily imbalanced. Usually, in such cases, application of some of the balancing techniques (e.g. over/undersampling, or more advanced SMOTE) is very common. However, after conduction of preliminary experiments on a similar data [14] we found out, that such modification is not necessary in this case.

During the pre-processing phase, we removed all punctuation marks, non-alphabetic characters, hypertext links not essential for the detection of fake news and kept only the letters of the Slovak alphabet. We also removed words that referred to the article's source (e.g., writing or photo credits). We wanted to focus solely on the textual content of the news piece; such data in the text may present an information leak about the source; therefore, we decided to remove them. Besides that, also stopwords were removed (e.g., prepositions, conjunctions, pronouns, etc.).

To train the models, it was necessary to convert the text content of the articles into a vector representation. Vector word representation or word embeddings is a technique where individual words are represented as vectors with a real value in a predefined n-dimensional vector space. Word embeddings capture the semantic and syntactic meaning, so almost similar vectors represent similar words placed close together in vector space. The result of word embeddings is a coordinate system in which similar words are placed close together. In our work, we used Word2vec embeddings. Word2vec is a technique for natural language processing published in 2013 [15]. The Word2vec algorithm uses a neural network model to learn word associations from a large corpus of text.

4   Models Training and Evaluation

In the experiments, we used the convolutional neural network (CNN), Long-Short-Term Memory (LSTM) network, and a CNN combined with the bidirectional LSTM. CNN is one of the most popular types of deep neural networks. The main advantage of CNN is that it automatically detects the important features without any human supervision. This is why CNN would be an ideal solution to computer vision and image classification problems. The benefit of using CNNs for sequence classification is that they can learn from the raw time series data directly, and do not require domain expertise to manually engineer input features. CNN's use the convolution operation instead of the general multiplication of matrices in at least one of the layers in their architecture. One dimensional convolutional layer creates a convolution kernel that is convolved with the layer input over a single spatial (or temporal) dimension to produce a tensor of outputs. A typical CNN architecture consists of three layers, a convolution layer, a pooling layer, and a fully connected layer. Layers are used to analyze images, objects, speech, or language features. LSTM is a recurrent neural network introduced by [16]. It is modified to solve the vanishing gradient problem and can model sequences and long-term dependencies more accurately. The LSTM architecture contains special units called memory blocks located in a hidden layer instead of neurons. These blocks have memory cells with separate connections that remember and store the network's temporary state. They also contain multiplicative units that control the flow of information and are called gateways. Authors in [17] introduced a special type of recurrent neural network, namely the bidirectional recurrent neural network (BRNN). The idea is to bring each training sequence in both directions into two separate recurrent networks connected to the same output layer. Two independent RNNs create the BRNNs by dividing the state neurons into the part responsible for the forward and backward direction.
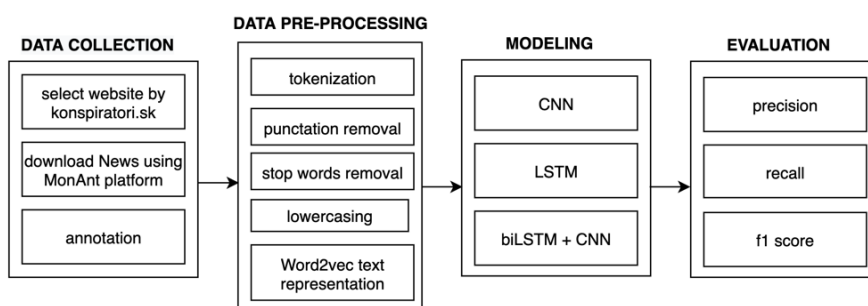

Figure 1
Workflow of the experiments

Figure 1 depicts the workflow of the experiments. The dataset consisted of 13,736 news articles, 12,885 regular, and 851 fake news. We divided the dataset into a training and test set. The training set contained 9,615 documents, of which 9,000 were relevant and 615 fake news. The test set, used for evaluating the model, contained 4,121 documents, of which 3,901 articles were regular news and 220 fake news. We used 10% of the training set to validate the model. The validation set was created by a stratified random split of the training set at each training of the model. Data pre-processing steps were described in Section 3.

4.1   The Architecture of the Models

In the phase of modeling, we used the following settings of models:

- activation function
    - hidden layers: ReLU
    - output layers: sigmoid
- loss function: binary cross-entropy
- optimizer: Adam
- regularization: Dropout, spatial dropouts

Because linear functions are severely limited and cannot recognize and learn complex patterns in the data needed to classify images, text, or sequences, neural network architectures use nonlinear activation functions [18]. Activation function is a function which decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias. We used Rectified Linear Unit (ReLU) - non-linear activation function used in nearly all modern neural network architectures. The output of ReLu is the maximum value between zero and the input value. Output is equal to zero when the input value is negative and the input value when the input is positive [19], [20]. The sigmoidal activation function is used as a nonlinear activation function on the output layer. We use it as the problem is a binary classification task, and usually, it can be used in conjunction with binary cross-entropy. This function transforms values in the range 0-1; thus, it determines the probability with which the input belongs to a given class.

In single-layer neural network architectures, the loss functions can be calculated directly from the weights. For training feedforward neural networks, error backpropagataion is used. Error backpropagation is about determining how changing the weights impact the overall loss in the neural network. The backpropagation works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer [21], [22]. As a loss function, we used Binary Cross-Entropy (BCE). BCE is a type of loss function used in binary classification problems. The function is used in the neural network to predict the

probability of a given example that belongs to one of two classes. The activation function on the output layer is, in this case, a sigmoidal function.

Optimizers are algorithms used to change the attributes of neural networks (e.g., weights, learning rate) to reduce loss. In our work, we used the Adaptive moment estimation (Adam) optimizer. It is a method that calculates individual adaptive learning rates for each parameter. It is designed as a combination of Adagrad and RMSProp methods, where it takes advantage of both. Adagrad works well in sparse transitions, and RMSprop works well in online and non-stationary conditions. Both also maintain the speed of learning. The advantages of this optimizer are that it works with sparse gradients, is directly implementable, and does not require much memory. Overall, this model is considered robust and suitable for wide use in solving optimization problems in machine learning [23], [24].

Goodfellow [25] defined regularization as any adjustment made to a learning algorithm and aims to reduce generalization losses. The dropout regularization technique is one of the most used regularizations in the field of deep learning. It provides a computationally inexpensive but powerful method for a wide range of models. Dropout can prevent overfitting by temporarily removing neurons with all of their incoming and outgoing connections and forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons [26], [27]. Also, we used a spatial version of Dropout. This version performs the same function as Dropout; however, it drops entire 1D feature maps instead of individual elements. In this case, *SpatialDropout1D* will help promote independence between feature maps.

In experiments, we compared the following architectures:

- **CNN model**. The embedding layer was followed by a one-dimensional convolution layer with 100 filters and window size 2 and the activation function ReLu. 1D CNN was followed by the pooling layer - GlobalMaxPooling, whose output is input to a feedforward neural network with one dense layer with 256 neurons and Relu activation function. The output layer contained one neuron and a Sigmoid activation function.

- **LSTM model.** The embedding layer was followed in this model by LSTM layers with 128 units. In this experiment, we added a dropout regularization layer with a parameter of 0.2, representing 20% of the input neurons that will be deactivated with each epoch, thus preventing over-fitting. Then were continued one fully connected layer with 128 neurons, and ReLu activation function and an output layer with one neuron, and a sigmoid activation function.

- **biLSTM + CNN model.** The embedding layer was followed by spatial Dropout with parameter 0.2. The output from the regularization layer was the input to the bidirectional recurrent LSTM layer with 64 units, with a

recurrent dropout of 0.1. The architecture continues with a one-dimensional convolution layer that contained 32 filters and a window size of 3. It was followed by the GlobalMaxPooling layer, which represented the entrance to the feedforward neural network with one dense layer with 64 neurons and a ReLU activation function. Then, the dropout regularization layer with a parameter of 0.2 was used again, and the output layer contained one neuron and a sigmoid activation function. The architecture of the third model is shown in **Table 1**. The structure of this architecture was inspired by previous experiments in the classification of toxic comments [10].

Table 1
biLSTM + CNN model

Layer (type)	Output Shape	Parameters
Input Layer	(None, 1,000)	0
Embedding	(None, 1,000,100)	20,177,100
Spatial Dropout	(None, 1,000,100)	0
biLSTM	(None, 1,000,128)	84,489
Conv1D	(None, 998,32)	12,320
Global Max Pooling	(None, 32)	0
Dense	(None, 64)	2,112
Dropout	(None, 64)	0
Dense	(None, 1)	65
Total params.:	20,276,077	
Trainable params.:	20,276,077	
Non-trainable params.:	0	

4.2   Evaluation

We evaluated all trained models on an independent test set. To evaluation models we used following metrics:

- **Recall** = TP / (TP+FN),

- **Precision** = TP / (TP + FP),

- **F1 score** = 2 * (Precision * Recall) / (Precision + Recall),

where

- **TP – True Positive** examples are predicted to be fake news and are fake news;

- **TN – True Negative** examples are predicted to be relevant news and are relevant news;

- **FP – False Positive** examples are predicted to be fake news but are relevant news;

- **FN – False Negative** examples are predicted to be relevant news but are fake news.

Table 2

Evaluation of models on the test set

	Precision	Recall	F1 score	Support
CNN model				
Regular News	0.98	1.00	0.99	3,901
Fake News	0.98	0.62	0.76	220
Accuracy			0.98	4,121
Macro avg	0.98	0.81	0.88	4,121
Weighted avd	0.98	0.98	0.98	4,121
LSTM model				
Regular News	0.99	1.00	0.99	3,901
Fake News	0.96	0.78	0.86	220
Accuracy			0.99	4,121
Macro avg	0.97	0.89	0.93	4,121
Weighted avd	0.99	0.99	0.99	4,121
biLSTM + CNN model				
Regular News	0.99	1.00	0.99	3,901
Fake News	0.97	0.79	0.87	220
Accuracy			0.99	4,121
Macro avg	0.98	0.89	0.93	4,121
Weighted avd	0.99	0.99	0.99	4,121

Table 2 depicts the results of the experiments. In this task, we focused mainly on increasing the value of recall because we want to detect as much fake news as possible. The best accuracy with a value of 98.76% was achieved in the model with the third architecture. The second and third architectures achieve very similar results, but since the third model is more robust and the resulting recall value is one percent higher than in the LSTM architecture, we decided to continue working with biLSTM+CNN architectures.

In the modeling phase, we also performed the optimization of hyperparameters using the grid search method for the best-performing model. We used the following combination of the hyperparameters and their values:

- **Dropout rate –** 0.1, 0.2, 0.3

- **Batch size –** 16, 32, 64

- **Optimizer –** Adam, Stochastic gradient descent (SGD)

**We obtained the best results using the dropout rate of 0.1, batch size 32, and
Adam optimizer. We trained the best model with these settings and achieved
an accuracy of 98.93 % on the test set.** **Table 3** shows the overall results of the
model after training the model using the best combination that came out in the
grid search optimization with 5-fold cross-validation. **Table 4** depicts the confusion
matrix of this model. We can observe that 34 regular news articles were classified
as fake news while 10 fake news pieces were classified as regular.

Table 3

Evaluation of the CNN + biLSTM architecture after optimization

	Precision	Recall	F1 score	Support
CNN+biLSTM model after grid search with cross-validation				
Regular News	0.99	1.00	0.99	3,901
Fake News	0.95	0.82	0.89	220
Accuracy			0.99	4,121
Macro avg	0.97	0.92	0.94	4,121
Weighted avg	0.99	0.99	0.99	4,121

After optimization, the accuracy of the best model increased even more to
98.93%, which represents an increase compared to the previous best model by
0.17%.

Table 4

Confusion matrix of the best model

Predicted values

	Fake News	Relevant News
Actual values — Fake News	True Positive (186)	False Negative (34)
Actual values — Relevant News	False Positive (10)	True Negative (3991)

Conclusions

In the presented paper, we focused on the detection of fake news in the Slovak
language using deep learning models. As most of the datasets contain news
articles written in English, we had to collect the database of news pieces from
various local online news portals. We decided to focus on the currently very
popular and important topic and collected the news articles related to the COVID-

19 pandemic. Spreading misinformation, especially in this domain, can present a serious issue that may affect people's health and lives. Collected data were annotated using a combination of manual techniques and expert knowledge provided by a curated list of misinformation sources. In the experiments, we used deep learning models, which according to the literature, gains superior results in similar tasks. We used CNN, LSTM, and a combined CNN+biSLTM model, fine-tuned using grid search, which proved to perform with an accuracy of 98.93% and an F1 score of 94%. Although the results sound promising, they are heavily influenced by the data and annotation quality. In the future, we plan to extend the dataset for the training of such models (e.g., create more generic datasets, not just related to the pandemic) and improve the annotation quality. As the data volume grows, we plan to utilize the crowdsourcing approach to obtain the class labels for the data combined with active learning. Also, the initial data collection proved that the final dataset would be heavily imbalanced. In this area, it would be appropriate to deal also with augmentation techniques in future work, which would expand the dataset and increase the robustness of the models.

Acknowledgment

References

[1]   S. D. Calkins and S. P. Keane, "NIH Public Access," Vol. 21, No. 4, pp. 1095-1109, 2010

[2]   H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," *J. Econ. Perspect.*, Vol. 31, No. 2, pp. 211-236, 2017

[3]   K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explor. Newsl.*, Vol. 19, No. 1, pp. 22-36, 2017

[4]   A. Bovet and H. A. Makse, "Influence of fake news in Twitter during the 2016 US presidential election," *Nat. Commun.*, 2019

[5]   K. Machova *et al.*, "Addressing False Information and Abusive Language in Digital Space Using Intelligent Approaches," in *Towards Digital Intelligence Society*, 2021, pp. 3-32

[6]   V. M. Krešňáková, M. Sarnovský, and P. Butka, "Deep learning methods for Fake News detection," in *2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo)*, 2019, pp. 143-148

[7]   W. Y. Wang, "' liar, liar pants on fire': A new benchmark dataset for fake news detection," *arXiv Prepr. arXiv1705.00648*, 2017

[8]   M. H. Goldani, S. Momtazi, and R. Safabakhsh, "Detecting fake news with

capsule neural networks," *Appl. Soft Comput.*, Vol. 101, p. 106991, 2021

[9]   N. O'Brien, S. Latessa, G. Evangelopoulos, and X. Boix, "The language of fake news: Opening the black-box of deep learning based detectors," 2018

[10] V. Maslej-Krešňáková, M. Sarnovský, P. Butka, and K. Machová, "Comparison of Deep Learning Models and Various Text Pre-Processing Techniques for the Toxic Comments Classification," *Appl. Sci.*, Vol. 10, No. 23, 2020

[11] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha, "FNDNet – A deep convolutional neural network for fake news detection," *Cogn. Syst. Res.*, Vol. 61, pp. 32-44, 2020

[12] J. Kapusta, "Improvement of Misleading and Fake News Classification for Flective Languages by Morphological Group Analysis," 2020

[13] I. Srba *et al.*, "Monant : Universal and Extensible Platform for Monitoring , Detection and Mitigation of Antisocial Behaviour," 2019

[14] K. Ivancova, M. Sarnovsky, and V. Maslej-Kresnakova, "Fake news detection in Slovak language using deep learning techniques," in *SAMI 2021 - IEEE 19th World Symposium on Applied Machine Intelligence and Informatics, Proceedings*, 2021

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," Jan. 2013

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, Vol. 9, No. 8, pp. 1735-1780, 1997

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, Vol. 45, No. 11, pp. 2673-2681, 1997

[18] S. Sharma, "Activation functions in neural networks," *Towar. Data Sci.*, Vol. 6, 2017

[19] N. Ketkar and E. Santana, *Deep Learning with Python*, vol. 1. Springer, 2017

[20] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017

[21] R. Hecht-Nielsen, "Theory of the backpropagation neural network," 1989

[22] H. J. KELLEY, "Gradient Theory of Optimal Flight Paths," *ARS J.*, Vol. 30, No. 10, 1960

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014

[24] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, 2018, pp. 1-2

[25]   I. Goodfellow and Y. Bengio, "Deep Learning," 2016

[26]   G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0, 2012

[27]   N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, Vol. 15, No. 1, pp. 1929-1958, 2014