Performance Comparison of Turkish Web Pages Classification

Saed ALQARALEH
Computer Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
saed.alqaraleh@hku.edu.tr

Hatice Meltem NERGIZ SIRIN
Software Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
hatice.sirin@hku.edu.tr

Furkan OZKAN
Computer Engineering Department
Hasan Kalyoncu University
Gaziantep, Turkey
furkan.ozkan@hku.edu.tr

***Abstract*—Nowadays, web page classification is essential for efficient and fast search engines. There is an ever-increasing need for automatic classification techniques with higher classification accuracy. In this article, a performance comparison of existing Turkish language CNN models for web pages classification systems is performed. In more detail, the content of web pages is extracted first, then preprocessing steps that aim to detect the important parts and eliminate useless contents are used. Next, Bert word embedding is integrated to represent the texts by efficient numerical vectors. Finally, three state-of-the-art CNN models that fully support the Turkish language are investigated to find the best classifier.**

Overall, the three studied models obtained an acceptable performance while classifying the Turkish webpages, however, the third model was able to achieve slightly better than the other two models.

Keywords— Convolutional Neural Networks, Web Page Classification, Textual Content, Multi-label technique

I. INTRODUCTION

The amount of digital information people share has increased very rapidly with the information age. People create and share new data rapidly [1-5]. It is necessary to organize and classify information according to certain rules to help users to find and access the required information among this extremely huge amount of data. Due to this rapid growth, web page classification is needed to help internet users. It is impossible to classify web pages manually, as the number of web pages increases with each passing minute, and an automated way of doing this is needed. Hence, such a system is essential for managing the enormous amount of information that exists on the Internet.

Mainly, web pages classification can be defined as the process of assigning each page to one or more predefined categories. Comparing the classification of the content of the web page to the standard text classification, web pages are semi-structured HTML documents. Also, web pages are usually written in HTML and have some HTML tags, hyperlinks, and some metadata. Some applications of web page classification are focused browsing, topic-based web link analysis, contextual advertising, and topical structure analysis of the web. In general, based on classification purposes, the webpage can be classified into a single label or multiple labels. Single-label classification is the application in which the only one of the existing classes is selected to be relevant for each webpage. Multi-label, on the other hand, multiple labels that are determined to be relevant to the input from the classes are selected. Overall, while determining a single class is sufficient in many applications, it may be desirable to use more than one class in some applications. For example, a news text containing the statements made by the prime minister of a country at a concert can be included in both the "politics" class and the "culture and arts" class.

Each web page consists of different types of elements such as images, videos, tables, text, demonstrations, and much more, arranged in a way that engages and conveys information in the best and easiest way. In this study, classification of web pages is done over text, ignoring the other multimedia components of web pages.

Web page classification is also necessary for topic specific crawling (searching web pages related to predefined topics), helping to develop web directories, assist in analyzing the current structure of the web, improving the quality of the website.

The main purpose of this study is to investigate the possibility of building an efficient classification model that fully supports the Turkish language.

This paper is organized as follows: Section 2 presents relevant studies and developments in the field of web page classification. Section 3 explains how to extract the features of web pages and the proposed model for the classification process. While how the data set is created, and the obtained results are described in Section 4. In the conclusion section, the performance evaluation of this study is highlighted and ideas for future studies are given.

II. RELATED WORK

The subject of text classification of web pages is well studied for some languages, especially English. However, when it comes to the Turkish language, more intensive investigation needs to be carried out to build an efficient classification system. In the following, we have summarized some recent studies related to this study.

In [6], the Support Vector Machines (SVM) using linear, polynomial, radial basis function, and sigmoid kernels were examined for web pages classification. In this article, a support vector machine has been used to train SVM, choosing the page property in terms of which category it belongs to. As a case study, the approach of [6] was used to classify the freelance and remote work opportunities on IT business web pages. The experimental results showed that the linear kernel function has obtained the best performance.

In [7], a deep learning-based model that combines long and short feature extractors was proposed. On this basis, the model examines composite features extracted from the title, text content, and description of the web page using both CNN and RNN networks. In order to further increase the accuracy of web page classification, an attention mechanism has been developed against unhelpful content. In the last

2021 Innovations in Intelligent Systems and Applications Conference (ASYU) | 978-1-6654-3405-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ASYU52992.2021.9599069

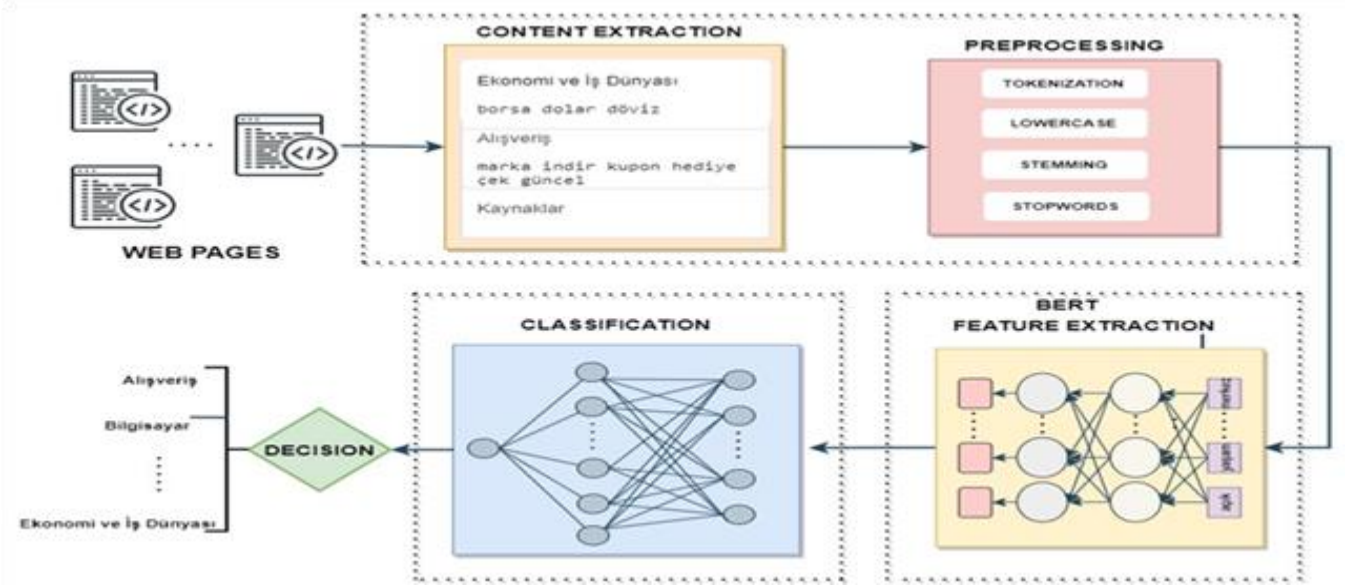

Fig. 1. Main Components of the proposed Turkish Web pages Classification system

layer of classification, XGboost, which is an implementation of gradient boosted decision tree, was used to further improve the accuracy of classification.

The classification of the text follows a hierarchical pattern and uses a combination of multiple neural networks was developed in [8], and its performance was compared against some machine learning algorithms. Overall, the weights of the word combined with Naïve Bayes, SVM, and Logistic regression using the GloVe embedding model yielded the best results.

A classification approach for analyzing contents and hypertext structures of public World Wide Web sites using Naive-Bayes and Two-layer Convolutional Neural Networks is proposed in [9]. Text information collected from website accounts is handled to control what kind of information can be obtained. The collection of parameters of websites are classified and evaluated by statistical clustering, Self-organizing map (SOM). Then, the HTML data of websites are classified by Naive-Bayes. Finally, the obtained features are faded into the layers of the Convolutional Neural Networks model.

The survey of [10] examined web page classification models under three main categories of textual, visual, and text-image combination methods. Textual content classification is generally based on the idea of creating a feature vector, counting the frequencies at which the terms of a word appear in the text, and applying these vectors to train the classifier. In textual content classification, it is stated that contextual information surrounding metadata and terms are ignored, and the structure of the text in HTML tags and hyperlinks is not sufficiently studied. In visual content classification methods, it is mentioned that feature extraction is predominantly based on computational and problem-specific analysis. In addition, it is emphasized that there is no specific model on how ignoring neither texts nor images affects classification accuracy, and there is also no study that measures the contribution of each feature to classification accuracy in web page classifications. It is also stated that there is a certain research gap regarding how to effectively apply structural information in web page classification. It is emphasized that CNN, RNN, and reinforcement learning methods are not applied sufficiently and that developing a detailed test environment is open to research in the evaluation of web page classifiers.

As mentioned before, due to the wide variety of noisy information and semi-structured data embedded in the web page, web page classification is a more complex problem than text classification. In [11], Word2vec and Skip-Gram models were combined to estimate the probability of the word in the front and back ranges based on the probability of the current target word. Results of [11] indicated that CNN, which combines the features of web page structure and semantic plain text, has the best performance. Also, the experimental results of the JFCNN (Joint Features Convolutional Neural Networks) method applied according to web page structure and semantic features were compared with TFCNN (Text Features Convolutional Neural Networks). As a result, JFCNN has an F1 Score of over 94% in each category and provides an improvement F1 score value between 4% and 6% compared to the single category.

This work of [12] was conducted to ensure that old sites automatically translated into the new site builder and then allow users to update the appearance of their websites effortlessly. The question of how to classify HTML documents according to their semantic functions with machine learning approaches was the main concern of this study. In other words, when a website is received as input, the site is automatically created according to semantic sections such as "gallery", "blog post", "contact information" using machine learning algorithms. In [12], a prototype with a multi-layer perceptron (multi-layer feed-forward neural network) is presented. The performance of the developed prototype was compared against the random forest, gradient boosting machines, and a neural network. According to the accuracy results for the prototype, the best model is the random forest model.

III. WEB PAGE CLASSIFICATION

The main purpose of this study is to investigate the possibility of building an efficient classification model that fully supports the Turkish language. The process of classifying the web pages is done according to their text

content. In data pre-processing, the initial document is prepared for training and classification processes. The main component of webpages classification systems are shown in Fig. 1., and their details are explained sequentially.

A. Extracting Web Page's Contents

Generally, HTML documents are semi-structured files that contain tags used to configure and organize information to display via the web browser. In this step, the URLs and the text of the web pages bodies are extracted to be used in the classification.

B. Text Preprocessing

It is obvious that pre-processing is essential and must be carried out before applying deep learning techniques. Respectively, removing punctuations, and ineffective words, also, converting all words to lowercase letters, and reaching the root phrase of the word are performed as shown in Figure 1.

1) Elimination of Useless Part: It is the process applied to separate the characters that do not contain any meaning are eliminated. The main purpose here is to reach useful words only. Therefore, all punctuation marks, spaces, numbers, and characters are eliminated.

2) Lowercase: All text is converted to lowercase letters, taking into account the alphabet of the language, in order to avoid expressions that may be perceived differently due to capital and lowercase letters.

3) Stemming: Here, the suffixes are removed from the word and the simplest form of the word is found. While obtaining this form of the word is called lemmatization, the state after the construction suffixes is discarded is called root, and this form is called stemming.

C. Feature Extraction/ BERT Word Embedding

Word Vectorization Algorithms in general and word embedding, in particular, are models in which words are represented according to their meanings. Today, the high rate of textual content in the concept of Big Data and traditional methods used in natural language processing studies have been inadequate. Word representations created by deep learning have achieved high success in solving problems in natural language processing. Word representation is the process of training the texts in the data set through artificial neural networks and assigning a numerical value(s) to each word. A data set consisting of texts is given to training models that perform word representations.

In other words, the information contained in the web pages is resumed in numerical vectors(features). Word vectorization is a transformation process applied before the classification process. In this step, i.e., transformation, the text is transformed into numerical vectors. This can be done using different methods, such as the proximity of words or the frequency of words in the dataset. As a result of this process, the text data is transformed into numerical values that can be used by classifiers.

In 2018, Google introduced the BERT model (Bidirectional Encoder Representations from Transformers). BERT has the ability to understand the relationship of any word with its surrounding words on both sides, right and left. In more detail, the sentence is divided into tokens, and a predefined token is added to the beginning and end of the sentence. Then, if the sentence is shorter than the maximum length, the gaps are filled, while if it is long, the text is expressed using a limited number of words. Overall, attention masks are created, and the input text is represented by tensor objects at the end of the process. Apart from being bi-directional, BERT is trained with two techniques, i.e., Masked Language Modeling (MLM) and Next Sentence Prediction (NSP).

D. CNN Text Classification

Text classification is the process of automatically selecting the most relevant class candidate for the input sample (a text, document, or sentence). The classification model presented in this paper is a multi-class text classifier. We report a series of experiments conducted on Convolution Neural Network (CNN) by training it on different datasets.

In the last decade, CNN was able to achieve impressive performance in many fields such as image classification and captaining. This encourages researchers to investigate the capability of CNN models when processing text. As expected, CNN was able to outperform other approaches, especially for some languages such as English. In this study, the performance of some state-of-the-art CNN models that were built to support the Turkish language was investigated, to find out the most suitable one for classifying Turkish webpages. The details of these models are summarized.

1) The first studied model which we call "CNN model 1" is shown in Fig. 2, and mainly consisted of three parts, the convolutional layer, the pooling layer, and the fully connected layer. In the convolutional layer, the input is filtered, and feature maps are obtained. Feature maps are sampled by the pooling layer, and more general and faster learning of the network is provided. Finally, each neuron in the fully connected layer generates an output based on all inputs from the previous layer. Each layer extracts attributes based on the numerals result of the previous layer and can learn the attribute hierarchy by combining and training all layers [13].

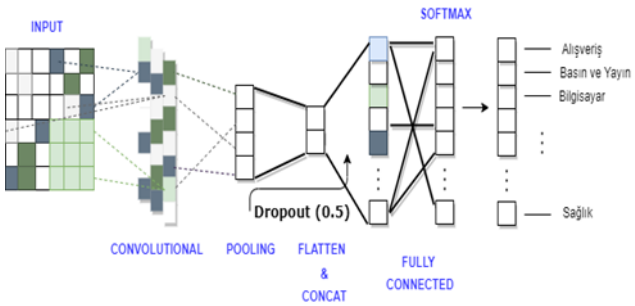

Fig. 2. A visualization of the first CNN model [13]

2) CNN Model 2: Our second CNN model [4] initiates embedding with random values, and its architecture is shown in Fig. 3. Here, the character-level placements are applied as input for the first layer, and this layer preserves the vector representation of each word. The input layer receives the input data both during training and evaluation. The vector representation of each word is then kept on the embedding layer. A dropout value of 0.5 is used for the dropout layer. Therefore, every time data is introduced, 50% of connections from the upper embedding layer will be discarded [4].

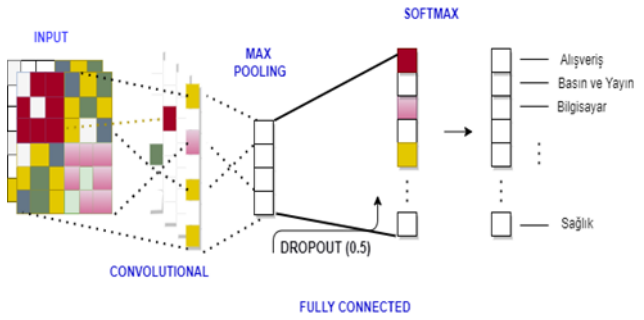

Fig. 3. A visualization of the second CNN model[4].

3) CNN Model 3: The Third CNN model was proposed in [14], and as shown in Fig. 4., it consists of an embedding layer, then, a pooling layer followed by two dense layers. In this model, an embedding layer was developed for natural language processing and used as the model's first layer. The second layer is a pooling layer where feature maps are sub-sampled. The model has two dropout layers to prevent overfitting, and two activation functions were used in the dense layer. One of them is the Rectified Linear unit (ReLU), which is nonlinear and has the advantage of not having back propagation errors. On the other hand, the other one has a Sigmoid function which can map the entire number line into a small range such as [0, 1] while also signifying high confidence for large positive or negative numbers.

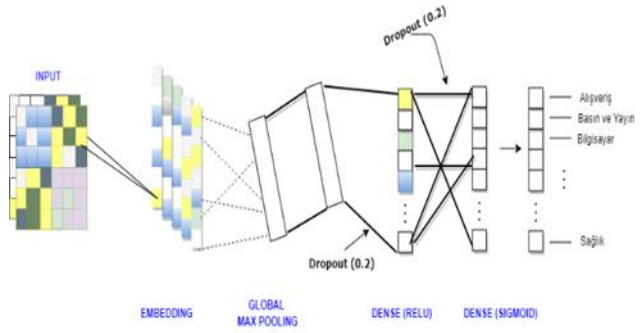

Fig. 4. A visualization of the third CNN model architecture [14].

IV. EXPERIMENTS

In this paper, the performance of three state-of-the-art CNN models that were originally developed for classifying the Turkish language text was examined in the first experiment using all the 22,346 Turkish webpages. In order to ensure the quality of the obtained results, in the second experiment, three sub-datasets, which contain samples of 5, 8, and 14 categories respectively were used to compare the performance of the mentioned models.

A. Dataset

We used the dataset of Turkish webpages introduced in [15]. This dataset contains 22,346 Turkish web pages. Then, as shown in Table 1, these pages were manually annotated and classified into 14 categories.

TABLE 1 THE DETAILS OF THE USED DATASET [15].

Topic	**#of samples**
Ekonomi_ve_İş_Dünyası	7145
Kaynaklar	3241
Bilgisayar	2491
Toplum	2239
Sağlık	1440
Kültür ve Sanat	1356
Alışveriş	1276
Eğlence ve Yaşam	714
Bilim	613
Spor	603
Basın ve Yayın	463
Oyunlar	288
Çocuklar ve Gençler	269
Ev	208

B. Setups of Implementation and Experiments

In order to build an efficient text classification system and especially if the input is webpages, preprocessing is essential. The text is extracted first from the web pages, and the "Keras Tokenizer" is used to tokenize the entire data. Then a padding is added if needed to unify the sequences of all texts. We split the data into training and test sets. For BERT, the batch size parameter was set to 64 and the number of epochs set to 3. Note that it is also possible to use pre-trained representations such as Glove or Word2Vec as transfer learning.

The performance evaluations were made using some appropriate criteria for classification, i.e., Accuracy, Precision, Recall, and F1.

1) Experiment 1:

The first experiment was conducted with the original dataset, i.e., 22,346 Turkish web pages. Also, in addition to the Accuracy, the Precision, Recall, and F1 scores were calculated. The results are shown in Fig. 5, and it is clear that the third model has obtained a slightly better result comparing to the other two model.

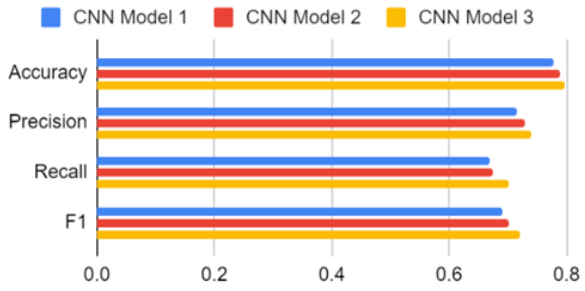

Fig. 5. The accuracy, precision, recall, and F1 score of the three models.

2) Experiment 2:

The second experiment was conducted using three sub-datasets, which contain samples of 5, 8, and 14 categories respectively, to investigate the performance stability of the mentioned models. Results are shown in Table 2 and Fig. 6 and Fig. 7. Also, the average of the results is shown in Fig. 8. Similar to the first experiment, still using all datasets the three models can obtain acceptable performance, however, the third model was able to achieve slightly better than the other two models.

TABLE 2 THE ACCURACY, PRECISION, RECALL AND F1 SCORE FOR ALL DATASETS AND MODELS.

Dataset (Category)	CNN Model	Accuracy	Precision	Recall	F1
1st Dataset (All)	Model 1	0.7792	0.7154	0.6698	0.6919
	Model 2	0.7883	0.7303	0.6732	0.7006
	Model 3	0.7964	0.7396	0.7029	0.7207
2nd Dataset (Top 8)	Model 1	0.8199	0.7944	0.7971	0.7957
	Model 2	0.8220	0.8015	0.7985	0.8001
	Model 3	0.8282	0.8129	0.7742	0.793
3rd Dataset (Top 5)	Model 1	0.8453	0.8256	0.8127	0.8191
	Model 2	0.8531	0.8373	0.8266	0.8319
	Model 3	0.8671	0.857	0.8382	0.8474

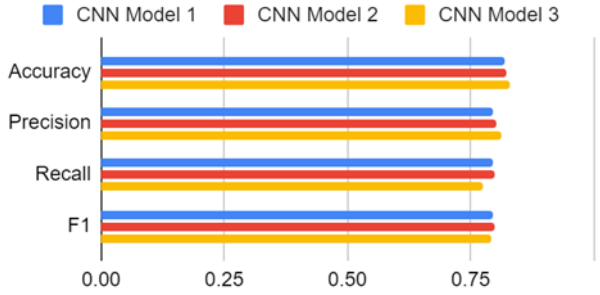

Fig. 6. The accuracy, precision, recall, and F1 score of the three models using a dataset of top 8 categories

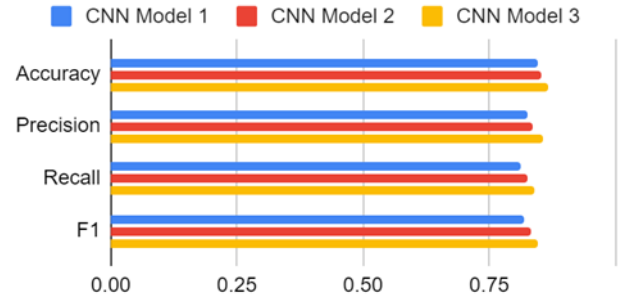

Fig. 7. The accuracy, precision, recall, and F1 score of the three models using a dataset of top 5 categories.

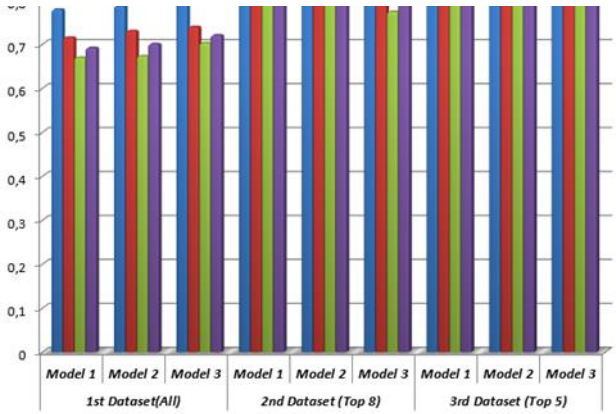

Fig. 8. The Average of Accuracy, Precision, Recall, and F1 of the three models using the three sub-datasets.

V. CONCLUSIONS

This paper highlights the importance of both textual CNN techniques for building an efficient classifier for Turkish web pages. For this purpose, the BERT word embedding which is based on our preliminary investigation provides the best results for categorizing web pages.

The most appropriate CNN model for classifying Turkish web pages was found after an intensive investigation of three state-of-the-art models. Then, a new efficient classifier that is mainly consisted of pre-processing flowed by the BERT, and finally the third studied model was developed and able to outperform the performance of the other studied models. Based on the findings of this paper, the CNN models can be further evaluated using larger datasets, as well as integrating the developed classifier with topic-specific web crawlers to establish some standard benchmarks can be the direction for future works

REFERENCES

[1] C. Balim, K. Özkan, "Functional classification of web pages with deep learning," IEEE 27th Signal Processing and Communications Applications Conference (SIU), 2019

[2] M. Z. Amin, N. Nadeem, "Convolutional neural network: text classification model for open domain question answering system." arXiv preprint arXiv:1809.02479, 2018.

[3] Y Wei., W. Wang, B. Wang, B. Yang, Y. Liu, (2018) A Method for Topic Classification of Web Pages Using LDA-SVM Model. In: Deng Z. (eds) Proceedings of 2017 Chinese Intelligent Automation Conference. CIAC 2017. Lecture Notes in Electrical Engineering, vol 458. Springer, Singapore.

[4] F. Kurt, Investigating the performance of segmentation methods with deep learning models for sentiment analysis on Turkish informal texts, Master's 585 thesis, MIDDLE EAST TECHNICAL UNIVERSITY (2018).

[5] A. Balagopalan, (San Franciso, CA, US), S. Hardik (Sunnyvale, CA, US), G. Carolina (San Francisco, CA, US) 2019 SYSTEMS AND METHODS OF TOPIC MODELING FOR LARGE SCALE WEB PAGE CLASSIFICATION United States 20190180327

[6] S. Sharmila, P. Joeg, and S. Vanjale. "Web document classification using support vector machine." IEEE International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)., 2017.

[7] Z. Qiuhan, W. Yang, and R. Hua. "Design and Research of Composite Web Page Classification Network Based on Deep Learning." 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)., 2019

[8] M. Kishan. "Content Based Hierarchical URL Classification with Convolutional Neural Networks." IEEE International Conference on Information Technology (ICIT)., 2019

[9] L. Xueyan, and R. Uda. "Classification of Web Site by Naive-Bayes and Convolutional Neural Networks." Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication. 2018

[10] H. Mahdi. (2020). Web page classification: a survey of perspectives, gaps, and future directions. Multimedia Tools and Applications. 79. 11921-11945. 10.1007/s11042-019-08373-8.

[11] H. Li, Z. Zhang and Y. Xu, "Web Page Classification Method Based on Semantics and Structure," 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 2019, pp. 238-243, doi: 10.1109/ICAIBD.2019.8837027.

[12] http://urn.fi/URN:NBN:fi:aalto-201905123012, Classification of Web Elements Using Machine Learning; Verkkosivuelementtien luokittelu koneoppimisen avulla, Virtanen, Erkka, 2019-05-06.

[13] K. Yoon. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 10.3115/v1/D14-1181.Letter Symbols for Quantities, ANSI Standard Y10.5-1968.

[14] S. Alqaraleh, and M. IŞIK. "Efficient Turkish tweet classification system for crisis response." Turkish Journal of Electrical Engineering & Computer Sciences 28.6 (2020): 3168-3182.

[15] S. Ş. Hüsem and A. Gülcü, "Categorizing the Turkish web pages by data mining techniques," 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 255-260, doi: 10.1109/UBMK.2017.8093385.