

Artificial intelligence for automatic moderation of textual content in online chats and social networks

Solomiia Liaskovska¹, Rex Bacarra², Yevhen Martyn³, Volodymyr Baidych⁴,
Jamil Abedalrahim Jamil Alsayaydeh⁵

¹Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine

²Department of General Education and Foundation, Rabdan Academy, Abu Dhabi, United Arab Emirates

³Department of Information Technologies and Electronic Communications Systems, Lviv State University of Life Safety Lviv, Ukraine

⁴Department Information Technologies Khmelnytskyi National University, Khmelnytsk, Ukraine

⁵Department of Engineering Technology, Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

Article Info

Article history:

Received Mar 11, 2024

Revised Feb 17, 2025

Accepted Mar 5, 2025

Keywords:

Artificial intelligence

Language classification

Machine learning algorithms

Neural networks

Spam detection

ABSTRACT

The article explores fundamental techniques for converting text into numerical data for machine learning algorithms. It meticulously examines various methods, including word vector representation via neural networks like Word2Vec, and explains the principles behind linear models such as logistic regression and support vector machines. Convolutional neural networks (CNN) and long short-term memory (LSTM) methods are also discussed, covering their components, mechanisms, and training processes. The research extends to developing and testing software for spam detection, hate speech identification, and recognizing offensive language. Using two datasets—one for labeled text messages and another for Twitter posts—the study analyzes data to address challenges like imbalanced data. A comparative analysis among linear models, deep neural networks, and single-layer models, using pre-trained bidirectional encoder representations from transformers (BERT) network, reveals promising results. The convolutional neural network stands out with a remarkable accuracy of 0.95. The study also adapts neural network architectures for hate speech and offensive language classification.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Jamil Abedalrahim Jamil Alsayaydeh

Department of Engineering Technology, Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka

76100 Melaka, Malaysia

Email: jamil@utem.edu.my

1. INTRODUCTION

Due to the vast volume of content created and distributed on the Internet, it has become increasingly challenging for moderators to promptly detect and remove harmful or objectionable content [1]–[3]. Additionally, the dynamic nature of online content and user interactions makes it difficult to establish static rules or guidelines for moderators to adhere to. Thus, the utilization of artificial intelligence for automatic moderation presents a pertinent and crucial solution to this issue. Our objective is to investigate the effectiveness of employing neural networks for regulating text content, specifically in detecting spam, hate speech, and abusive language using machine learning algorithms. This involves analyzing the features of machine learning methods for natural language processing tasks, such as text classification, developing a model for identifying spam messages based on deep neural networks, comparing the results obtained with

classical machine learning methods, creating a text classification model for detecting hate speech and offensive language based on the trained bidirectional encoder representations from transformers (BERT) transformer model, and analyzing the accuracy and efficiency of the developed model. To accomplish this, we propose the development of a software implementation of machine learning algorithms designed to detect spam and hate speech in online communication. Our research integrates methods and algorithms of natural language processing commonly utilized for text classification.

Thus, in order to utilize the most appropriate and up-to-date information, as well as effective methods, it is necessary to verify the relevance of the data involved in our research. Among the subject areas to which scientific works correlating with our chosen scientific direction can belong are computer science, engineering, and mathematics. It is evident that the choice is quite obvious and aligns with our scientific objectives to investigate the effectiveness of using neural networks to regulate textual content using machine learning methods.

Works [4]–[8] delve into the methods and means of detecting social spam, which encompasses illegal text content [9]–[13], offensive language, hate speech, cyberbullying, and disinformation. In [1], the need for the development of effective methods for detecting social spam is emphasized, wherein the support vector method (SVM), random forest, and naive Bayesian algorithm are employed to address this crucial social problem. They propose a concept for developing more accurate and context-dependent hostile language detection systems [14]. Such machine learning algorithms as the support vector method, random forest, Bayesian classifier, k-nearest neighbor's method, and multilayer perceptron are explored in [3]. In [4], various speech models-transformers like BERT, XLNet, and RoBERTa are compared in their efficacy in detecting clickbait headlines. Additionally, [6] underscores the necessity for interdisciplinary cooperation among information technology professionals, sociologists, and legal experts to devise effective and ethical solutions to the problem of harmful content.

Various works offer solutions to the spam problem. In [5], a model based on the convolutional neural network (CNN) architecture is proposed. Research [7] claims that the deep learning long short-term memory (LSTM) model outperforms other models in terms of all metrics: accuracy, reliability, recall, and F1-score. In [8], the emphasis is on the problem of classification accuracy of emotional coloring in text data, and it is found that the best model is a random forest, achieving an accuracy of over 80%. Research [15]–[19] underscores the importance of classification accuracy and spam message detection using transformer models and ensemble learning [20]. Proposed transformer models, including BERT and eXtreme Gradient Boosting (XGBoost), are utilized for spam classification and detection.

Another group of works proposes directions and methods for solving the spam problem [20]–[24]. The authors of [16] introduce a new approach to text classification based on CNN and Bidirectional LSTM models, which, in their opinion, better capture semantic information and demonstrate increased accuracy for tweet classification. The work [25] suggests an approach that combines a pre-trained transformer model with a CNN, while [21], [22], [26] presents a spam detection system in the Twitter network in real-time alongside sentiment analysis using machine learning and deep learning methods. The authors of [17] proposes a new approach to improving spam detection using a deep recurrent neural network, while [18]–[20] presents a binary classifier based on machine learning. In article [9], based on a comprehensive review of methods and evaluation metrics for detecting socially unacceptable statements, it is concluded that future research should focus on developing more reliable and accurate methods capable of coping with the dynamics of text data flows on online platforms [27], [28]. Online communication abuse takes many forms – it can be cyberbullying, misinformation, spam, and more. Online propaganda deserves particular attention – through widespread use of fake accounts on social media, various political or public figures and organizations can disseminate desired information to shape public opinion. This creates a challenge for information filtering and control [29]–[31]. Typically, owners of various online forums or chats use people to monitor published content, but this method has obvious drawbacks – a person physically cannot review the content of hundreds of messages posted in a short period, especially during mass spam attacks on users. Large platforms such as Twitter [23], [26] or Facebook use software tools to detect content that violates platform rules, but they also have limitations – these tools often do not consider the message context or the cultural background of its author, thus they frequently block content that does not violate community guidelines [14], [18], [20], [22]. This issue is highly relevant for social network Instagram. Therefore, modern automated content moderation tools are not flawless, and despite existing solutions that use artificial intelligence to detect illicit content, this area requires further research [32]–[37].

2. METHOD

In this scientific work, authors solve the task of text classification using machine learning algorithms [38]–[40]. Such algorithms are not able to work directly with text data, for this they need to be converted into a numerical format - vectors. Therefore, we analyzed the main methods [41]–[43] of text data representation [44]: one-hot vector, bag of words, term frequency–inverse document frequency (TF-IDF),

n-grams, as well as vector representation of words and its implementation through the word2vec model [45]–[48]. In particular, two model architectures are involved in the research [49]–[52]: Continuous bag of words (CBOW) and Skip-gram, linear machine learning algorithms and neural networks. There are several methods, the most common of which is “bag of words.” The bag of words marks the presence of a word in input documents compared to all words in the dataset. Therefore, its implementation requires a dictionary of all used words and an indicator of word presence [53], [54]. All data inputted into machine learning models will thus be represented as numerical vectors (1):

$$[x_1, x_2, x_3 \dots x_n], \quad (1)$$

The previous method can be improved by representing each word in the vector not just as 0 or 1, but rather by its count in the document or its frequency relative to the total number of words in the text. The main drawback of this approach is that words appearing in every document will have the highest frequency and create informational noise. To address this issue, term frequency–inverse document frequency (TF-IDF) exists – a metric that determines the significance of a word for a specific document against its significance for the entire corpus. It is logical to assume that a word appearing in all input data will have a low value for a specific document, whereas a word appearing in only one document will better describe it. TF-IDF is calculated for each word, and the higher the value of the metric, the more significant the word is for the document. The formula for the metric is as (2):

$$tf - idf = tf * \log \left(\frac{n}{f} \right) \quad (2)$$

tf – term frequency, n – total number of documents, f – number of documents containing the word. The essence of our research is the use of machine learning algorithms to detect spam.

Dense vectors or context vectors are vectors used to describe a word based on its relationships with other words. Given a sentence, we can take a specific window around the chosen word with a size of n words to represent its context. Words that have similar contexts – meaning they share the same surrounding words as word x , will be considered synonyms or semantically similar to word y . Then, for the chosen word, we can form a vector $[x_1, x_2, x_3 \dots x_n]$ where each variable represents the frequency of each word's occurrence in the corpus within the vicinity of the chosen word. Since words are represented as vectors, we can measure the similarity between words using the formula of the dot product, specifically finding the cosine similarity (3):

$$\cos \alpha = \frac{a*b}{|a|*|b|} \quad (3)$$

where a and b are vector representations of words. Word2vec is a two-layer neural network that processes text by “vectorizing” words. It takes a textual corpus as input and produces a set of dense vectors representing words in that corpus. There are two main architectures: CBOW and skip-gram.

Authors used Python programming language, libraries for machine learning, natural language processing and data visualization NLTK, sklearn, matplotlib.pyplot, seaborn, neattext as tools for research. As a development environment, Google Colab was used an interactive online environment for performing data analysis and visualization tasks, which allows you to break the code into separate parts, run them independently of each other, visualizes the process of code execution in real time and gives the opportunity to immediately see the result execution of the desired part of the program, which greatly simplifies their writing and debugging.

2.1. Classifier of linear models, LSTM, spam based CNN and BERT

2.1.1. Data analysis and pre-processing

The dataset used for model training consists of 5,574 text messages, which are labeled as spam and non-spam. Figure 1 presents an overview of the dataset, including general statistics such as word frequency and class distribution. This visualization helps to understand the nature of the data and its balance, which is critical for training effective classification models.

Stop-words are words that are present in the text, but by themselves do not make sense, such as conjunctions, prepositions, other official parts of speech, and exclamations. Also, stop words usually include words that are found in almost all corpora of a certain language. By throwing them out, you can get rid of unnecessary noise and give more weight to words that are more important and have a significant impact on the content of the document. The NLTK library contains a built-in list of stop words for each language.

label	ham	spam
count	4825	747
unique	4516	653
top	Sorry, I'll call later Please call our customer service representativ...	
freq	30	4

Figure 1. Description and visualization of the data set

Since the specific meaning of words is relatively unimportant for spam detection, stemming can be used. It is much faster and easier to implement. There are several stemming implementations in the NLTK library. The target variable in the dataset takes two string values. Since the models can only work directly with numerical data, we encode the value of the target variable according to the binary classification problem. Using the built-in *train_test_split()* function of the sklearn library, we split the data set into training and test samples. Tokenization is the process of dividing a document into word components - tokens, after tokenization, we will convert documents into numerical vectors using the methods of “bag of words”, n-grams and TF-IDF. To do this, we will use the sklearn library package for extracting features from text data and classes for vectorization. Let's involve instances of the CountVectorizer and TfidfTransformer classes to create new datasets for each feature extraction method. For each of the data sets, we train two linear models: logistic regression and the support vector method. We will use the following metrics to evaluate the models:

- Accuracy-score – the ratio of the number of correctly predicted classes to the number of all predicted data, characterizes the accuracy of the model;
- Precision-score – the ratio of the number of correctly predicted positive (y=1) data to the number of all predicted positive data, which characterizes the error with which the model can accept data marked as negative as positive;
- Recall – the ratio of the number of correctly predicted positive data to the sum of the number of correctly predicted positive and falsely predicted negative data, characterizes the model's ability to determine positive data;
- F1-score – a metric used to calculate the ratio of the proportion of objects that were classified by the model as positive and really were positive to the proportion of found positive data from all positive data in the set, calculated by the formula:

$$F = \beta^2 + 1 * \frac{Precision * Recall}{\beta^2 Precision * Recall}$$

where β – is the weight for metrics.

After training the model and testing the model, the following metrics were obtained:

- Accuracy-score=0.952, precision-score=0.97, recall=0.93, f1-point=0.95 for logistic regression trained on “bag of words”;
- Accuracy-score=0.94, precision-score=0.964, recall=0.91, f1-point=0.936 for the method of support vectors trained on the “bag of words”;
- Accuracy-score=0.94, precision-score=0.988 recall=0.89, f1-score=0.93 for logistic regression trained on the “bag of unigram and bigram”;
- Accuracy-score=0.947, precision-score=0.988, recall=0.9, f1-score=0.94 for the method of support vectors trained on the “bag of unigram and bigram”;
- Accuracy-score=0.95, precision-score=0.976, recall=0.922, f1-score=0.95 for logistic regression trained on TF-IDF vectors;
- Accuracy-score=0.9679144385026738, precision-score=0.98, recall=0.955, f1-score=0.966 for the support vector method trained on TF-IDF vectors.

To visualize the quality of the models, we will output the error matrix for each data set for each model. Figure 2 demonstrates where Figure 2(a) shows error matrices for logistic regression and Figure 2(b) shows the support vector method that we can see that both models for bag-of-words data are equally good at identifying non-spam messages. But the linear regression method is better at directly classifying spam itself.

Figure 3 demonstrates that the above matrices and we can conclude that for data sets containing unigram and bigram. Figure 3(a) shows error matrices for logistic regression (left). Figure 3(b) shows the support vector method the models give almost identical results, but in turn predict less false positive data.

We can conclude that the determination of the message label as spam or not spam does not improve much when taking into account the additional context. Because when using the n-grams dataset, the overall accuracy of the model did not improve, it even decreased slightly, but at the same time the model does less

spam detection errors. Fewer false-positive data and the lowest accuracy are demonstrated by models trained on data in the TF-IDF indicator format.

For TF-IDF format data in Figure 4 demonstrates that the support vector method better classifies messages containing spam, while allowing fewer errors than logistic regression. Figure 4(a) shows error matrices for logistic regression. Figure 4(b) shows the support vector method.

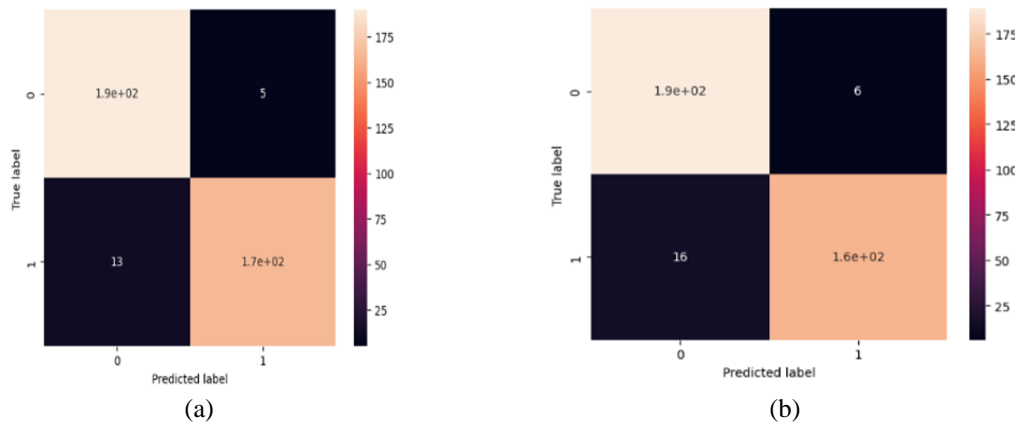


Figure 2. Error matrices for logistic regression (a) and the support vector method and (b) for the “bag of words” data set

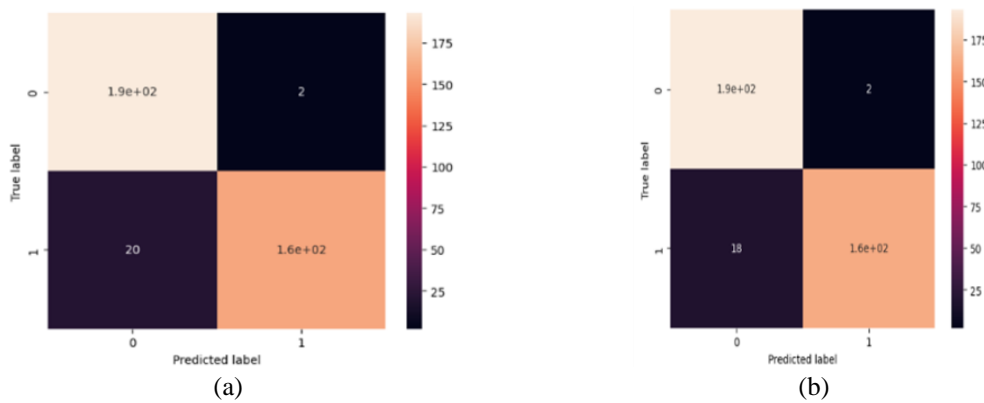


Figure 3. Error matrices for logistic regression (left) (a) and the support vector method and (b) (right) for the “bag of words” data set

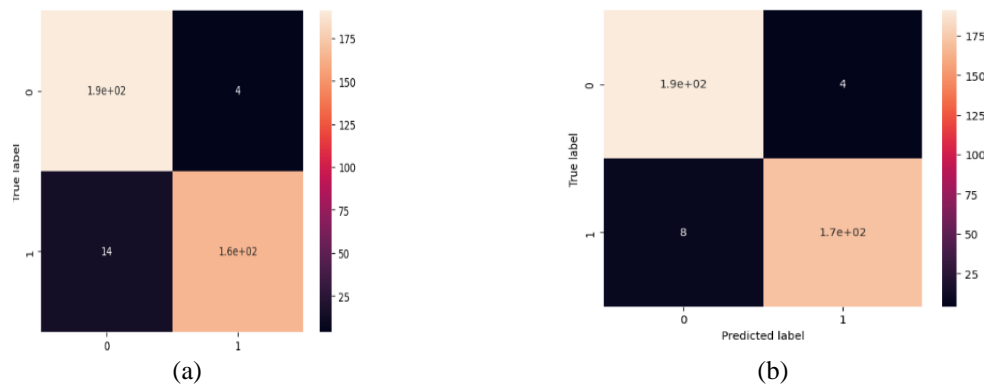


Figure 4. Error matrices for logistic regression (a) and the support vector method (b) for the TF-IDF dataset

3. RESULTS AND DISCUSSION

This section describes using CNN and LSTM neural network architectures for spam classification. In order to train neural networks on our data, it is necessary to bring all vectors, that is, all documents, to a fixed length. As the dimension value of the vectors, authors take the length of the largest document. Authors change the dimensionality of the vectors using the `pad_sequences()` function of the Keras library, the `padding=` argument will have the value “post”, which indicates the zero values of the function, due to which we expand the vector and add them to the end.

3.1. Description of the network architecture

Embedding layer: input data to the network is a sequence of words, which are represented as integers word indexes in the dictionary. The Embedding Layer transforms these integers into vectors of given dimension containing the representation of the word through its contextual relationship with other words. 80 is the size of the length of the input vector of tokens, and the dimension of dense vectors is 100;

- LSTM: A layer of an LSTM network that can store long-term dependencies in sequential data. It consists of neurons for processing sequential input data and saving information about the state of the neural network;
- GlobalMaxPooling1D: a layer acting as a filter for features generated by LSTM, its output is the maximum value from each vector of features;
- Dropout and batch normalization: a dropout layer is applied after the LSTM layer to filter the number of firing neurons to prevent overtraining. After that, batch normalization is applied to standardize the input data to the previous layer;
- Dense: a fully connected layer accepts LSTM output data after processing by several layers, contains 80 neurons, for nonlinear transformation of the input data by the rectified linear unit (ReLU) activation function;
- Dropout: repeated removal of a part of neurons;
- Dense: an output dense layer with one neuron and a sigmoid activation function, used to calculate the output probability that an object belongs to a class.

Figure 5 presents the performance of the LSTM-based network. Figure 5(a) shows the accuracy curve for both training and validation sets, indicating consistent improvement and good generalization. Figure 5(b) illustrates the loss curve, which steadily decreases, suggesting that the model is not overfitting and is learning effectively over time.

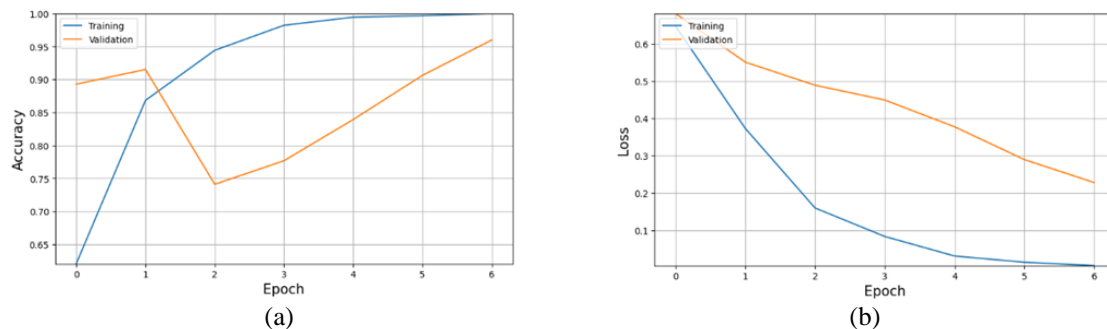


Figure 5. Graph of network losses and accuracy based on LSTM model (a) accuracy plot and (b) loss plot

From the given graphs, we can say that the network is not overtrained, because the losses on the validation data are constantly decreasing. We initialize the convolutional neural network. Figure 6 demonstrates the architecture of a convolutional neural network.

Let's perform the following description of the network architecture:

- Embedding layer: the layer vectorizes the input data into dense context vectors. The size of the input vectors is 80, and the dimension of the dense vectors is 50. Convolutional layer: A convolutional layer containing 64 filters is applied, with a one-dimensional kernel of dimension 3 and a ReLU activation function. This layer performs a convolution operation on the input sequence thus forming a feature map for the vector. GlobalMaxPooling1D: The input data are feature maps from the convolutional layer, the current layer in turn selects the maximum value from each feature map, thus reducing the data volume and highlighting the most important information. Dropout and Batch Normalization: removing part of the neurons and standardizing the input data to speed up and regulate the network. Dense: the output data

after “screening” and standardization passes through a fully connected dense layer with 256 neurons and ReLU activation function. This layer performs a non-linear transformation of the input data, enabling the network to learn complex data relationships. Dropout and batch normalization: repeatedly removing part of the neurons and standardizing the input data to speed up and regulate the network. Figure 7(a) demonstrates that plot the accuracy and Figure 7(b) demonstrates that losses of the convolutional network.

This model is also not overtrained; one can say that the accuracy values for the training data are relevant for the entire dataset since they coincide with the model's accuracy for the validation data. The accuracy of the convolutional neural network exceeds the accuracy of the long short-term memory network. Figure 8 demonstrates a comprehensive comparison, let's output the confusion matrices for the deep networks and calculate the key metrics.

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 80, 50)	342300
conv1d (Conv1D)	(None, 78, 64)	9664
global_max_pooling1d (GlobalMaxPooling1D)	(None, 64)	0
dropout_4 (Dropout)	(None, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 64)	256
dropout_5 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 256)	16640
dropout_6 (Dropout)	(None, 256)	0
batch_normalization_3 (Batch Normalization)	(None, 256)	1024
dropout_7 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257

Figure 6. Architecture of a convolutional neural network

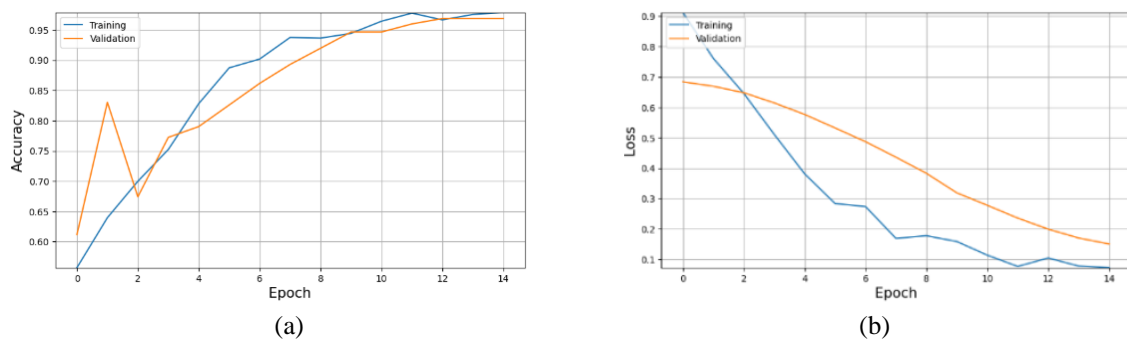


Figure 7. Plot of accuracy and loss for a convolutional neural network (a) accuracy and (b) loss

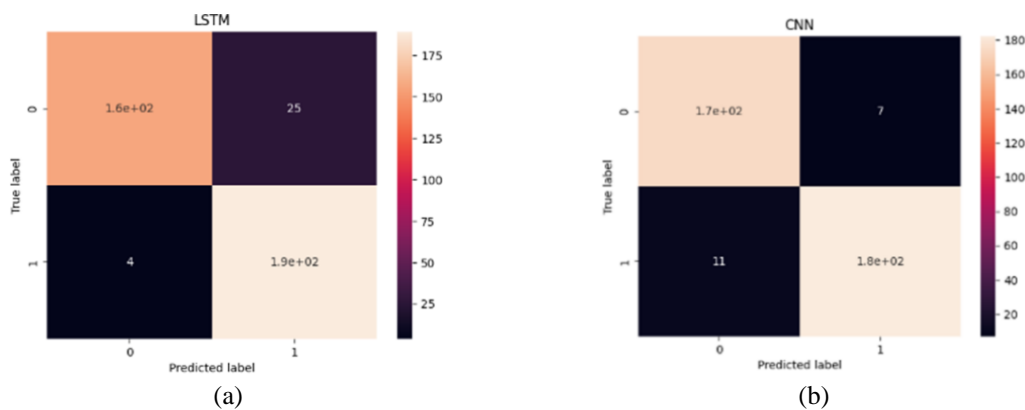


Figure 8. The confusion matrices for (a) LSTM and (b) CNN

Metrics for the models: Accuracy Score: 0.92, Precision Score: 0.88, Recall: 0.98, F1-Score: 0.928 for LSTM. Accuracy Score: 0.95, Precision Score: 0.96, Recall: 0.94, F1-Score: 0.95 for CNN.

3.2. Parameters analysis, and results

BERT is a language representation model designed to create bidirectional representations for deep neural networks on raw, unannotated text by combining left and right contexts in all layers. All models in the BERT family use a partial implementation of transformer models, namely encoders, as the network's output is a language model. BERT is pre-trained on “dirty” text data, so there is no need to perform preprocessing that was used for linear models and neural networks. Dense layer: A fully connected layer consisting of one neuron and a logistic activation function to return the probability of an object belonging to a class. Next step is to train the model and output the confusion matrix Figure 9 demonstrates loss and accuracy plots similarly to deep networks.

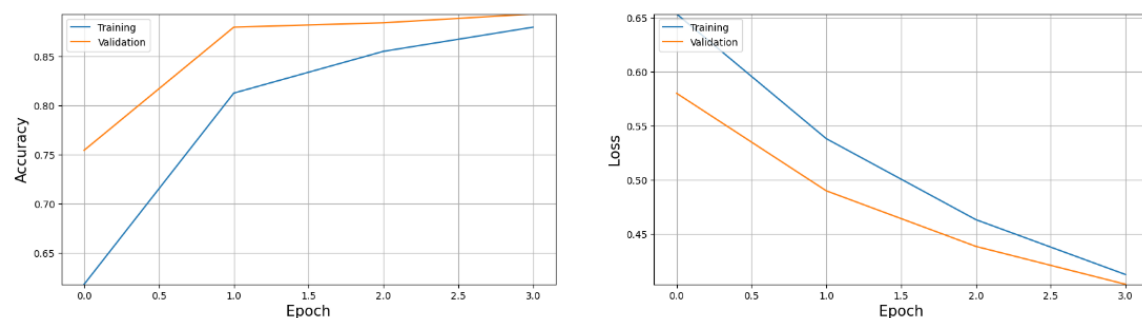


Figure 9. The loss and accuracy plot for the BERT-based model

BERT is trained using masked token prediction and next sentence prediction. Through this training process, BERT acquires contextual, latent representations of tokens based on their context. Figure 10 demonstrates the loss and accuracy plot for the BERT-based model.

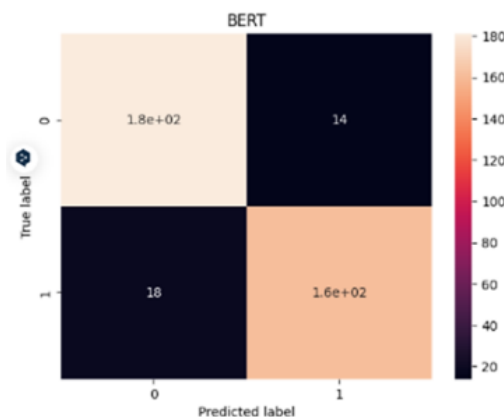


Figure 10. The loss and accuracy plot for the BERT-based model

3.3. Comparison and discussion

For classification, we will use the architectures of models for spam classification, but since we have three classes in the dataset, for each of the networks, we need to change the output layer and the loss function. As the output layer, we used a dense connected layer with one neuron and a logistic activation function, thus obtaining the probability of an object belonging to the target class. After making changes to the architecture, we will train deep learning models and demonstrate the loss and accuracy graph. Figure 11 demonstrates accuracy and loss plot for the LSTM-based model. Figure 12 demonstrates accuracy and loss plot for the CNN-based model

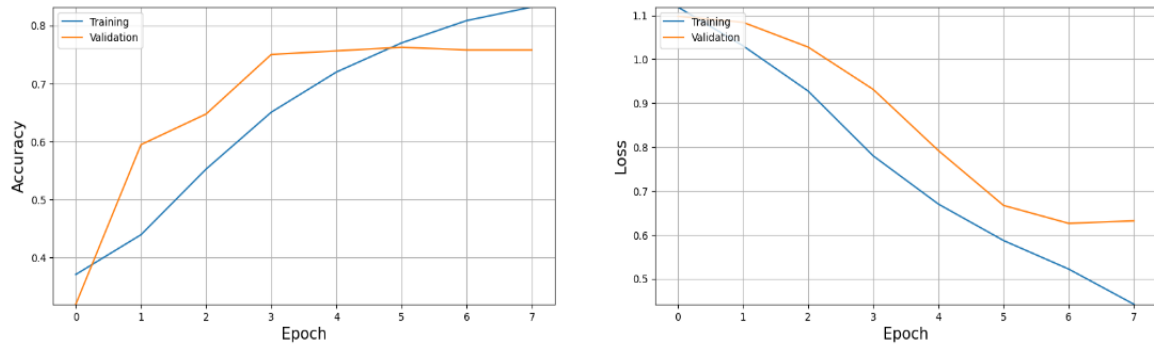


Figure 11. Accuracy and loss plot for the LSTM-based model

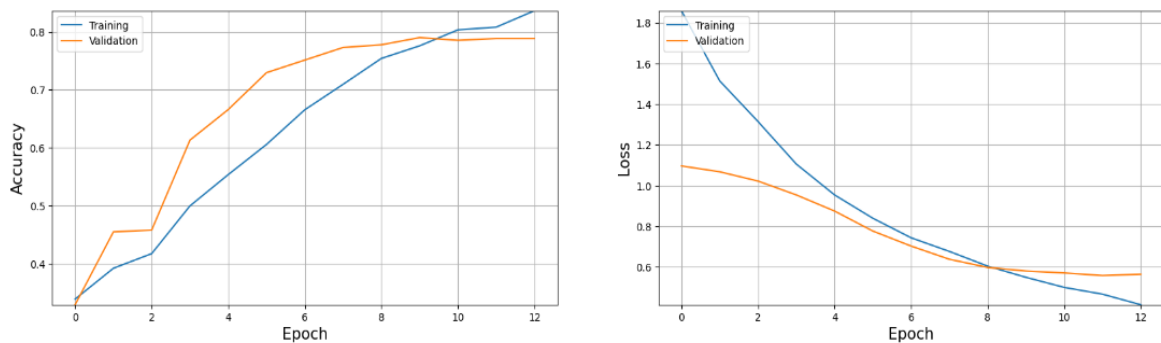


Figure 12. Accuracy and loss plot for the CNN-based model

To analyze the accuracy of the models, we will apply a confusion matrix. Figure 13 demonstrates confusion matrices for LSTM Figure 13(a) and CNN Figure 13(b). The results of the confusion matrices allow us to understand the performance of the models. Both models make the most errors when classifying hate speech, often mistaking it for offensive language. Specifically, the accuracy in classifying posts containing hate speech is lower than the accuracy in classifying posts containing offensive language or “normal” posts. The models perform best at distinguishing offensive language from regular posts. Overall, considering all error values, it can be concluded that this implementation of the convolutional neural network handles the task better than the long short-term memory network.

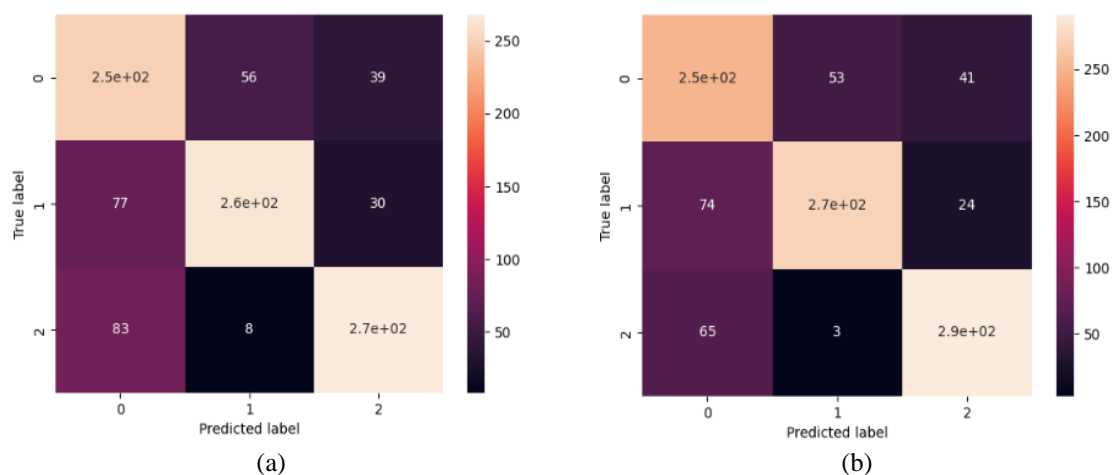


Figure 13. Confusion matrices for (a) LSTM and (b) CNN

Let's compute the metrics of the models for each class for comparison, and we can see the results in Tables 1 and 2. These tables present the performance of the LSTM and CNN models across hate speech, offensive language, and regular posts. Table 3 further demonstrates the classification metrics for the BERT-based model, allowing for a comprehensive comparison among all three approaches. Table 1 shows metrics for each class of publication for the LSTM model. Table 2 shows for each class for publication for the CNN model.

Table 1. Metrics for each class of publications for the LSTM model

	Accuracy-score	Precision-score	Recall	F1-score
Hate speech	0.75	0.78	0.62	0.69
Offensive language	0.75	0.8	0.71	0.75
Regular posts	0.75	0.8	0.745	0.77

Table 2. Metrics for each class of publications for the CNN model

	Accuracy-score	Precision-score	Recall	F1-score
Hate speech	0.76	0.72	0.68	0.70
Offensive language	0.76	0.82	0.74	0.78
Regular posts	0.76	0.81	0.81	0.81

Table 3. Metrics for each class of publications for the model based on BERT

	Accuracy-Score	Precision-score	Recall	F1-score
Hate speech	0.3	0.33	0.3	0.31
Offensive language	0.3	0.35	0.32	0.33
Regular posts	0.3	0.32	0.33	0.32

The proposed architecture based on BERT fields rather poor results. We can assume that this is related to the nonlinearity of dependencies in textual data since our network is essentially equivalent to a linear model with a single Embedding layer. In the context of spam detection, a comparative study was conducted for linear models, deep neural networks, and single-layer models using the pre-trained BERT network. Additional datasets were created for different text representation techniques, namely bag-of-words, n-grams, and TF-IDF. As a result, three pairs of logistic regression and support vector machine models were trained. All models achieved reasonably high overall accuracy, with logistic regression performing better in identifying spam for the standard bag-of-words, while the support vector machine had higher metrics for TF-IDF. The lowest overall accuracy was observed for the TF-IDF data format, although the gap in all metrics for the three datasets is not significant. Deep models and the BERT-based model were then trained. The convolutional neural network model demonstrated the highest accuracy with a value of 0.95.

For the classification of hate speech and offensive language, we used the same neural network architectures as for spam, adapting their output layer for multi-class classification tasks. Again, the convolutional neural network achieved the highest accuracy - 0.76, while the BERT-based model showed very low results - 0.3.

4. CONCLUSION

The main techniques for representing text in numerical format for machine learning algorithms were investigated, analyzing their characteristics, working principles, advantages, and disadvantages. The method of word vector representation using neural networks, exemplified by the word2vec model, was detailed. For the chosen linear models—logistic regression and support vector machines—an explanation of their working principles and mathematical foundations was provided. The description of convolutional neural networks and the long short-term memory method included their basic architectural components, operational principles, and training processes. The specificity of using convolutional layers for textual data was also discussed.

A dataset was selected for each classification task. The research work includes a detailed description of the data preprocessing and feature extraction process using various methods. Corresponding implementations of machine learning algorithms were trained for each dataset, and model performance results were demonstrated. It was found that logistic regression and support vector machines can classify spam with high accuracy, and different data representations minimally affect the model results. From the research findings, it was concluded that detecting spam in messages is weakly dependent on the semantic content of the text; frequently used words can be crucial indicators of spam.

ACKNOWLEDGMENTS

The authors extend their appreciation to Universiti Teknikal Malaysia Melaka (UTeM) and to the Ministry of Higher Education of Malaysia (MOHE) for their support in this research.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Solomiia Liaskovska	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Rex Bacarra	✓	✓			✓	✓	✓			✓	✓		✓	✓
Yevhen Martyn	✓				✓	✓	✓	✓	✓	✓		✓	✓	
Volodymyr Baidych		✓	✓	✓	✓	✓		✓	✓		✓			
Jamil Abedalrahim	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Jamil Alsayaydeh														

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, J.A.J.A., upon reasonable request.

REFERENCES




- [1] B. Abu-Salih *et al.*, "An intelligent system for multi-topic social spam detection in microblogging," *Journal of Information Science*, vol. 50, no. 6, pp. 1471–1498, Dec. 2024, doi: 10.1177/01655515221124062.
- [2] T. Davidson, D. Warmesley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," *Prepr. arXiv.1703.04009*, Mar. 2017.
- [3] S. Abro, S. Shaikh, Z. Hussain, Z. Ali, S. Khan, and G. Mujtaba, "Automatic hate speech detection using machine learning: A comparative study," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 8, pp. 484–491, 2020, doi: 10.14569/IJACSA.2020.0110861.
- [4] Y. Kim, "Convolutional neural networks for dentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.
- [5] P. Rajapaksha, R. Farahbakhsh, and N. Crespi, "BERT, XLNet or RoBERTa: The best transfer learning model to detect clickbaits," *IEEE Access*, vol. 9, pp. 154704–154716, 2021, doi: 10.1109/ACCESS.2021.3128742.
- [6] V. U. Gongane, M. V. Munot, and A. D. Anuse, "Detection and moderation of detrimental content on social media platforms: current status and future directions," *Social Network Analysis and Mining*, vol. 12, no. 1, Dec. 2022, doi: 10.1007/s13278-022-00951-3.
- [7] W. H. Bangyal *et al.*, "Detection of fake news text classification on COVID-19 using deep learning approaches," *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 1–14, Nov. 2021, doi: 10.1155/2021/5514220.
- [8] F. Gholami, Z. Rahmati, A. Mofidi, and M. Abbaszadeh, "On enhancement of text classification and analysis of text emotions using graph machine learning and ensemble learning methods on non-English datasets," *Algorithms*, vol. 16, no. 10, Oct. 2023, doi: 10.3390/a16100470.
- [9] S. Kaddoura, G. Chandrasekaran, D. Elena Popescu, and J. H. Duraisamy, "A systematic literature review on spam content detection and classification," *PeerJ Computer Science*, vol. 8, Jan. 2022, doi: 10.7717/peerj-cs.830.
- [10] M. Sumathi and S. P. Raja, "Machine learning algorithm-based spam detection in social networks," *Social Network Analysis and Mining*, vol. 13, no. 1, Aug. 2023, doi: 10.1007/s13278-023-01108-6.
- [11] A. B. Singh, K. M. Singh, Y. J. Chanu, K. Thongam, and K. J. Singh, "An improved image spam classification model based on deep learning techniques," *Security and Communication Networks*, vol. 2022, pp. 1–11, Aug. 2022, doi: 10.1155/2022/8905424.

- [12] Z. Zhang, Z. Deng, W. Zhang, and L. Bu, "MMTD: A multilingual and multimodal spam detection model combining text and document images," *Applied Sciences*, vol. 13, no. 21, Oct. 2023, doi: 10.3390/app132111783.
- [13] I. Ahmad, M. Yousaf, S. Yousaf, and M. O. Ahmad, "Fake news detection using machine learning ensemble methods," *Complexity*, vol. 2020, pp. 1–11, Oct. 2020, doi: 10.1155/2020/8885861.
- [14] C. Intelligence and Neuroscience, "Retracted: Real-time Twitter spam detection and sentiment analysis using machine learning and deep learning techniques," *Computational Intelligence and Neuroscience*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2022/5211949.
- [15] A. Ghourabi and M. Alohalay, "Enhancing spam message classification and detection using transformer-based embedding and ensemble learning," *Sensors*, vol. 23, no. 8, Apr. 2023, doi: 10.3390/s23083861.
- [16] H. Huan, Z. Guo, T. Cai, and Z. He, "A text classification method based on a convolutional and bidirectional long short-term memory model," *Connection Science*, vol. 34, no. 1, pp. 2108–2124, Dec. 2022, doi: 10.1080/09540091.2022.2098926.
- [17] A. Mosavi, S. Shamshirband, E. Salwana, K. wing Chau, and J. H. M. Tah, "Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning," *Engineering Applications of Computational Fluid Mechanics*, vol. 13, no. 1, pp. 482–492, 2019, doi: 10.1080/19942060.2019.1613448.
- [18] M. F. Abdul Kadir, A. F. A. Abidin, M. A. Mohamed, and N. A. Hamid, "Spam detection by using machine learning based binary classifier," *Indonesian Journal of Electrical Engineering and Computer ScienceE*, vol. 26, no. 1, pp. 310–317, Apr. 2022, doi: 10.11591/ijeecs.v26.i1.pp310-317.
- [19] E. M. Dogo, T. Makaba, O. J. Afolabi, and A. C. Ajibo, "Combating road traffic congestion with big data: A bibliometric review and analysis of scientific research," in *Towards Connected and Autonomous Vehicle Highways*, 2021, pp. 43–86. doi: 10.1007/978-3-030-66042-0_4.
- [20] D. Nallaperuma *et al.*, "Online incremental machine learning platform for big data-driven smart traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4679–4690, 2019, doi: 10.1109/TITS.2019.2924883.
- [21] N. V. Babu and E. G. M. Kanaga, "Sentiment Analysis in Social Media Data for Depression Detection Using Artificial Intelligence: A Review," *SN Computer Science*, vol. 3, no. 1, p. 74, Jan. 2022, doi: 10.1007/s42979-021-00958-1.
- [22] N. M. Samsudin, C. F. binti Mohd Foozy, N. Alias, P. Shamala, N. F. Othman, and W. I. S. Wan Din, "Youtube spam detection framework using naïve bayes and logistic regression," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1508–1517, Jun. 2019, doi: 10.11591/ijeecs.v14.i3.pp1508-1517.
- [23] G. K. Wadhvani, P. K. Varshney, A. Gupta, and S. Kumar, "Sentiment analysis and comprehensive evaluation of supervised machine learning models Using Twitter data on Russia–Ukraine War," *SN Computer Science*, vol. 4, no. 4, Apr. 2023, doi: 10.1007/s42979-023-01790-5.
- [24] A. K. Rajpoot, P. Nand, and A. I. Abidi, "Development of textual analysis using machine learning to improve the sentiment classification," *Journal of Physics: Conference Series*, vol. 2062, no. 1, Nov. 2021, doi: 10.1088/1742-6596/2062/1/012014.
- [25] R. T. Mutanga, N. Naicker, and O. O., "Hate speech detection in Twitter using transformer methods," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, pp. 614–620, 2020, doi: 10.14569/IJACSA.2020.0110972.
- [26] M. M. Khan, K. Shahzad, and M. K. Malik, "Hate speech detection in Roman Urdu," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 1, pp. 1–19, Jan. 2021, doi: 10.1145/3414524.
- [27] M. F. Juna and M. Hayaty, "The observed preprocessing strategies for doing automatic text summarizing," *Computer Science and Information Technologies*, vol. 4, no. 2, pp. 119–126, Jul. 2023, doi: 10.11591/csit.v4i2.p119-126.
- [28] N. Afifie *et al.*, "Evaluation Method of Mesh Protocol over ESP32 and ESP8266," *Baghdad Science Journal*, vol. 18, p. 1397, Dec. 2021, doi: 10.21123/bsj.2021.18.4(Suppl.).1397.
- [29] J. Shen, R. H. Deng, Z. Cheng, L. Nie, and S. Yan, "On robust image spam filtering via comprehensive visual modeling," *Pattern Recognition*, vol. 48, no. 10, pp. 3227–3238, Oct. 2015, doi: 10.1016/j.patcog.2015.02.027.
- [30] K. Cao *et al.*, "Optimization control of adaptive traffic signal with deep reinforcement learning," *Electronics*, vol. 13, no. 1, Jan. 2024, doi: 10.3390/electronics13010198.
- [31] L. Chen, N. Song, and Y. Ma, "Harris hawks optimization based on global cross-variation and tent mapping," *The Journal of Supercomputing*, vol. 79, no. 5, pp. 5576–5614, Mar. 2023, doi: 10.1007/s11227-022-04869-7.
- [32] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
- [33] A. Goswami *et al.*, "Sentiment analysis of statements on social media and electronic media using machine and deep learning classifiers," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–18, Mar. 2022, doi: 10.1155/2022/9194031.
- [34] J. A. J. Alsayaydeh, W. A. Indra, A. W. Y. Khang, V. Shkaruplyo, and D. A. P. P. Jkatisan, "Development of vehicle ignition using fingerprint," *ARNP Journal of Engineering and Applied Sciences*, vol. 14, no. 23, p. 23, 2019.
- [35] J. Angskun, S. Tipprasert, and T. Angskun, "Big data analytics on social networks for real-time depression detection," *Journal of Big Data*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00622-2.
- [36] T. Siddiqui, S. Hina, R. Asif, S. Ahmed, and M. Ahmed, "An ensemble approach for the identification and classification of crime tweets in the English language," *Computer Science and Information Technologies*, vol. 4, no. 2, pp. 149–159, Jul. 2023, doi: 10.11591/csit.v4i2.p149-159.
- [37] H. M. Saleh, "An efficient feature selection algorithm for the spam email classification," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 9, no. 3, pp. 520–531, Aug. 2021, doi: 10.21533/pen.v9i3.2202.
- [38] V. Shkaruplyo, I. Blinov, A. Chemeris, V. Dusheba, J. A. J. Alsayaydeh, and A. Oliinyk, "Iterative Approach to TLC Model Checker Application," in *2021 IEEE 2nd KhPI Week on Advanced Technology (KhPIWeek)*, 2021, pp. 283–287. doi: 10.1109/KhPIWeek53812.2021.9570055.
- [39] C. Shang and F. You, "Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era," *Engineering*, vol. 5, no. 6, pp. 1010–1016, 2019, doi: 10.1016/j.eng.2019.01.019.
- [40] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, pp. 1–48, Jan. 2019, doi: 10.1145/3278607.
- [41] H. Yang, Q. Liu, S. Zhou, and Y. Luo, "A spam filtering method based on multi-modal fusion," *Applied Sciences*, vol. 9, no. 6, Mar. 2019, doi: 10.3390/app9061152.
- [42] M. Tubishat, F. Al-Obeidat, A. S. Sadiq, and S. Mirjalili, "An improved dandelion optimizer algorithm for spam detection: Next-generation email filtering system," *Computers*, vol. 12, no. 10, Sep. 2023, doi: 10.3390/computers12100196.
- [43] Z. F. Sokhangoe and A. Rezapour, "A novel approach for spam detection based on association rule mining and genetic algorithm," *Computers & Electrical Engineering*, vol. 97, Jan. 2022, doi: 10.1016/j.compeleceng.2021.107655.
- [44] V. V. Shkaruplyo, I. V. Blinov, A. A. Chemeris, V. V. Dusheba, and J. A. J. Alsayaydeh, "On applicability of model checking technique in power systems and electric power industry," in *Systems, Decision and Control in Energy III*, 2022, pp. 3–21. doi: 10.1007/978-3-030-87675-3_1.




- [45] J. A. J. Alsayaydeh, W. A. Indra, A. W. Y. Khang, V. Shkaruplyo, and D. A. P. P. Jkatisan, "Development of vehicle ignition using fingerprint 2," *ARNP Journal of Engineering and Applied Science*, vol. 14, no. 23, p. 23, Apr. 2019.
- [46] C. Colón-Ruiz and I. Segura-Bedmar, "Comparing deep learning architectures for sentiment analysis on drug reviews," *Journal of Biomedical Informatics*, vol. 110, Oct. 2020, doi: 10.1016/j.jbi.2020.103539.
- [47] B. M. Gurusamy, P. K. Rengarajan, and P. Srinivasan, "A hybrid approach for text summarization using semantic latent dirichlet allocation and sentence concept mapping with transformer," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 6, pp. 6663–6672, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6663-6672.
- [48] Y. Y. Tan, C.-O. Chow, J. Kanesan, J. H. Chuah, and Y. Lim, "Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning," *Wireless Personal Communications*, vol. 129, no. 3, pp. 2213–2237, Apr. 2023, doi: 10.1007/s11277-023-10235-4.
- [49] A. Rastgoo and H. Khajavi, "A novel study on forecasting the airfoil self-noise, using a hybrid model based on the combination of CatBoost and Arithmetic Optimization Algorithm," *Expert Systems with Applications*, vol. 229, Nov. 2023, doi: 10.1016/j.eswa.2023.120576.
- [50] P. Dhal and C. Azad, "Hybrid momentum accelerated bat algorithm with GWO based optimization approach for spam classification," *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 26929–26969, Sep. 2023, doi: 10.1007/s11042-023-16448-w.
- [51] L. Geng, S. Zhang, J. Tong, and Z. Xiao, "Lung segmentation method with dilated convolution based on VGG-16 network," *Computer Assisted Surgery*, vol. 24, no. sup2, pp. 27–33, Oct. 2019, doi: 10.1080/24699322.2019.1649071.
- [52] O. Polska, R. Kudermetov, J. Abedalrahim, J. Alsayaydeh, and V. Shkaruplyo, "QoS-aware web-services ranking: Normalization techniques comparative analysis for LSP method," *ARNP Journal of Engineering and Applied Sciences*, vol. 16, no. 2, pp. 248–254, 2021.
- [53] N. F. Binti Abdul Rahim *et al.*, "Channel congestion control in VANET for safety and non-safety communication: A review," in *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*, Dec. 2021, pp. 1–6. doi: 10.1109/ICRAIE52900.2021.9704017.
- [54] I. Fedorchenko, A. O. Oliynyk, J. Alsayaydeh, A. Kharchenko, A. Stepanenko, and V. Shkaruplyo, "Modified genetic algorithm to determine the location of the distribution power supply networks in the city," *ARNP Journal of Engineering and Applied Sciences*, vol. 15, no. 23, pp. 2850–2867, 2020.

BIOGRAPHIES OF AUTHORS






Solomiia Liaskovska    received a master's degree in computer science from Lviv Polytechnic National University and earned a Ph.D. in the design of multiparameter systems through computer technology. She is currently an associate professor at the Department of Artificial Intelligence Systems, Lviv Polytechnic National University. Modern software tools and products play a crucial role in her theoretical research. In addition to the software products utilized in her Ph.D. research, such as AI tools and the MATLAB scientific computer graphics system, she also employs software and programming in Python, using the Matplotlib package for data analysis in her scientific research. She can be contacted at email: solomiya.y.lyaskovska@lpnu.ua.






Rex Bacarra    Ph.D., is a distinguished educator, speaker, and writer with a doctorate in philosophy from De La Salle University, a top QS-ranked institution in the Philippines. He has extensive experience teaching and serving in administrative roles at institutions such as Geneva Business School, York St. John Business School, and the American College of Dubai, where he was dean of general education. Recognized for his contributions, Dr. Bacarra has been named one of the 100 most influential Filipinos in the Gulf for five consecutive years and is a sought-after speaker on leadership, classroom engagement, and gamified learning. His writings have appeared in publications like Khaleej Times and The Philippine Daily Inquirer, with research interests in ethics, teaching, and organizational development. Contact: rbacarra@ra.ac.ae.






Yevhen Martyn    doctor of technical sciences in the field of applied geometry and engineering graphics, is a professor at the Department of Descriptive Geometry and Engineering Graphics, and also a professor at the Department of Information Technologies and Electronic Communications Systems of Lviv State University of Life Safety. He graduated from Lviv Polytechnic Institute (now National University "Lviv Polytechnic"), Lviv in 1973. He has been a doctor of technical sciences since 2000 and a professor at the Department of Descriptive Geometry and Engineering Graphics since 2003. He can be contacted at email: evmartun@gmail.com.



Volodymyr Baidych    received a bachelor's degree in computer science at Lviv Polytechnic National University, Ukraine in 2023. He is currently studying at Khmelnytskyi National University at the Information Technologies Department. His research interests include machine learning, data mining and preprocessing, natural language processing, and web service composition. The main theme of his current projects is using artificial intelligence to analyse text information. He can be contacted at email: vvbaydych@gmail.com.



Jamil Abedalrahim Jamil Alsayaydeh    (member, IEEE) received a degree in computer engineering from Zaporizhzhya National Technical University, Ukraine, in 2009, an M.S. degree in computer systems and networks from Zaporizhzhya National Technical University, Ukraine, in 2010, and a Ph.D. in engineering sciences with a specialization in automation of control processes from National Mining University, Ukraine, in 2014. He is currently a senior lecturer at the Department of Engineering Technology, Faculty of Electronic and Computer Engineering and Technology, Universiti Teknikal Malaysia Melaka (UTeM) since 2015. His teaching portfolio includes a range of courses such as computer network & security, internet technology & multimedia, software engineering, computer system engineering, data communications & computer network, computer network & system, real time system, programming fundamental, digital signal processing, and advanced programming. He is a research member at the Center for Advanced Computing Technology. His research interests are formal methods, simulation, internet of things, computing technology, artificial intelligence and machine learning, computer architecture, algorithms, and applications. Dr. Alsayaydeh has more than 66 research publications to his credit that are indexed in SSCI, SCIE, and Scopus, and have been cited by over 300 documents. He supervises undergraduate and postgraduate students and is a reviewing member of various reputed journals. Currently, he actively publishes research articles and receives grants from the government and private sectors, universities, and international collaborations. He is also a member of the Board of Engineers Malaysia (BEM). He can be contacted at email: jamil@utem.edu.my.