A BERT Deep Learning Model for Arabic Spam Detection

Hadir Driss[1] ; Jaouhar Fattahi[2] ; Mohamed Mejri[2] ; Sahbi Bahroun[1] and Ridha Ghayoula[3]

***Abstract*— Spam messages pose a significant cybersecurity threat, leading to phishing attacks, fraud, and privacy breaches. Traditional spam detection methods, such as rule-based filtering and statistical models, often fail to capture the evolving and complex nature of spam messages. In this paper, we propose an Arabic spam detection model leveraging BERT (Bidirectional Encoder Representations from Transformers), a deep learning-based NLP model. Our approach enhances classification accuracy by utilizing contextual text representations specific to the Arabic language. We preprocess Arabic text using AraBERT tokenization and fine-tune the BERT-based model on a balanced dataset of Arabic spam and ham messages. Experimental results demonstrate that our model achieves high accuracy (98%), outperforming traditional machine learning and deep learning approaches. This research highlights the potential of transformer-based models in Arabic spam filtering, paving the way for more efficient and robust detection systems.**

***Index Terms*— Arabic NLP, Spam Detection, BERT, Deep Learning, Text Classification.**

I. Introduction

The great threat can be caused to cyber security and user privacy by spam messages, whether they are emails, SMS, or social media messages. Apart from being a daily headache, these unsolicited messages can also be effective in phishing attacks, fraud, and other malicious actions. The automatic detection of spam has become a paramount problem and requires solutions effective to the language and cultural traits of different regions around the globe. A range of the approaches examining different opportunities for filtering this type of content has been suggested, including anything from rule-based methods to machine-learning techniques [1]–[6]. Unfortunately, even within this diverse landscape, there lies the potential for spammers to employ ever more complex and rapidly evolving tactics. Keyword-based filters and statistical models have observed that these traditional methods do not capture the linguistic nuances involved when confronted with new spam patterns. Deep learning and Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) represent a bright alternative to tackle these issues. BERT has outperformed standard natural language processing techniques due to its contextual understanding of the meaning of words in a sentence. Nevertheless, applying such models to the Arabic language is still a challenging task due to Arabic's rich morphology, many dialects, and the unavailability of adequately annotated datasets for supervised training. In this paper, we propose an Arabic spam detection approach based on a BERT model, leveraging contextual text representations to enhance classification accuracy. We analyze the effectiveness of this approach through various experiments and compare its performance with other traditional and deep neural network-based classification models. Our results demonstrate the potential of BERT for this task, paving the way for more efficient and robust spam filtering systems for the Arabic language.

[4]Hadir Driss *(hadirdriss45@gmail.com)* and Sahbi Bahroun *(Sahbi.Bahroun@isi.utm.tn)* are with the High Institute of Computer Science, University of Tunis El-Manar, Tunis, Tunisia.

[2]Jaouhar Fattahi *(Jaouhar.fattahi.1@ulaval.ca)* and Mohamed Mejri *(Mohamed.Mejri@ift.ulaval.ca)* are with the Department of Computer Science and Software Engineering, Laval University, Quebec, Canada.

[3]Ridha Ghayoula *(Ridha.Ghayoula@umoncton.ca)* is with the Faculty of Engineering, University of Moncton, New Brunswick, Canada.

II. Background

Spam detection has remained an important research area for years because the amount of unwanted messages had reached an unlimited level that affected users all over many digital communication channels. Traditional spam filtering like rule-based and statistical techniques have been widely put into use for identifying and blocking spam messages. Earlier approaches relied primarily on hand-crafted rules, keyword filtering, and Bayesian classification. However, as the spam messages keep evolving, they have become very flexible and use obfuscation techniques like misspellings, synonyms, and adversarial modifications to bypass detection mechanisms [7].

For these reasons, machine learning techniques have become increasingly popular for spam classification. Approaches such as Support Vector Machines (SVM), Decision Trees, and Naïve Bayes classifiers are extensively used because of their ability to learn from labeled datasets and generalize to unseen messages. Most of these techniques require handcrafted features like Term Frequency-Inverse Document Frequency (TF-IDF), n-gram representation, etc., which may not capture the semantic meaning of the text fully. Deep learning has made many more changes to spam detection. It has made it possible for models to automatically learn hierarchical representations of texts without feature engineering [8].

Among deep architectures, CNNs and RNNs [9], particularly Long Short-Term Memory (LSTM), have found some success in spam experiments involving modern deep learning. They are capable of sensing both local and long successions of text content and can, therefore, give a considerable boost to classification performance. However, just like most models, they still have trouble processing with long dependencies and complicated structures of language, especially in morphemically rich languages like Arabic [10].

The very recent progress in NLP introduced the Transformers' technologies, coming up with models like BERT, which uses self-attention mechanisms for bidirectional contextual representation. Unlike conventional word embedding, which provides banks of static representation, BERT pack dynamic contextual embeddings, thereby increasing its value for text classification, most especially spam detection. Many studies have shown that BERT outperforms older systems in diverse NLP applications, but the application of BERT in Arabic spam detection is minimal [11]. In this study, we applied BERT for Arabic spam detection and improved the detection of Arabic linguistic complexities through fine-tuning pre-training models on domain datasets. Our approach aimed to improve accuracy in spam detection while alleviating the limitations experienced in traditional and deep-learning based methods.

III. BERT ARCHITECTURE

The architecture of the BERT model is a multi-layer bidirectional transformer encoder.The exact architecture depends on the version used: There are two model sizes of BERT:

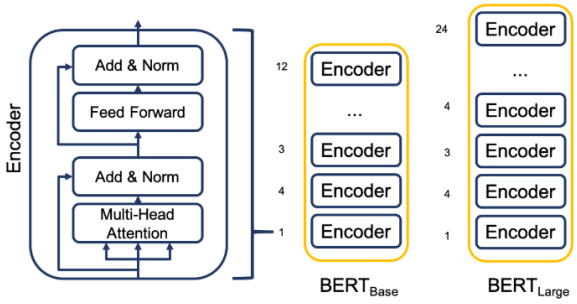

Figure I: BERT Architecture

BERT Base and BERT Large.The differences in model sizes between BERT Base and BERT Large are shown in the table.

	BERT Base	BERT Large
Layers	24	112
Hidden Size	768	1024
Heads	12	16
Parameters	110 M	340 M

Table I: Differences between BERT Base and BERT Large

A. Key Components of BERT

1) Embedding Layer

(i) WordPiece Embeddings: Converts words into subwords to better handle unknown words.
(ii) Position Embeddings: Adds positional information to tokens.
(iii) Segment Embeddings: Indicates whether a token belongs to sentence A or B.

2) Transformer Encoder Each Transformer block consists of:

(i) Multi-Head Self-Attention: Each token attends to itself and other tokens through multiple attention heads.
(ii) Add and Norm: Normalization after residual connection.
(iii) Feed-Forward Network (FFN): A dense neural network to transform token representations.
(iv) Add and Norm: Another normalization after the residual connection.

3) Output Layer

(i) BERT generates embeddings for each token (for general NLP tasks).
(ii) For classification (AutoModelForSequenceClassification), a dense layer is added on top of the [CLS] token to classify text into two categories.

IV. METHODOLOGY AND EXPERIMENT

This work proposes a deep-learning-based approach for Arabic spam detection using the BERT model. Our approach includes data preprocessing, text tokenization, model training, and evaluation using different performance metrics. The experimental setup is designed to ensure robust training and generalization across different spam and ham messages in Arabic. The workflow of the discovery system is shown in Fig. II.

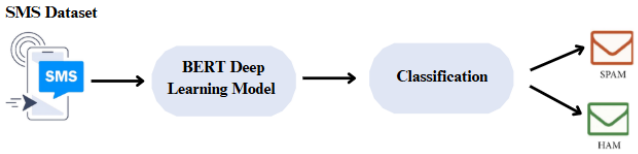

Figure II: Workflow of the Discovery System

A. Development setup

The development environment setup for our experiment is as follows.

- Software environment : Google Colab, Python 3.11 with TensorFlow, Keras, Matplotlib, NumPy, Pandas, Torch,etc.
- Hardware environment: 7th Gen Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz, 12 GB RAM
- OS: Windows 10 Professional

B. Dataset and Preprocessing

For this research, we will use ArabicSpam.csv, a moderately balanced collected dataset containing Arabic SMS messages affecting into either spam or ham. The dataset has the following main column names:

- SMS :text message.
- Sentiment : Labels denoting ham as 0 and spam as 1.

This dataset is extremely balanced because it has 50% ham and 50% spam messages. This eliminates the class

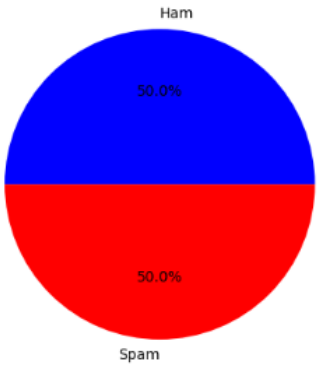

Figure III: Balanced dataset

imbalance issue in the model, leading to better learning and generalization Fig. III represents a balanced dataset.

The text data will undergo several preprocessing techniques, including text normalization, removal of special characters, and tokenization using AraBERT [12], a pre-trained BERT tokenizer specifically designed for Arabic text processing. The dataset is then split into a training set (80%) and a test set (20%) for efficient model evaluation.

V. Data Preprocessing

Preprocessing is performed to remove irrelevant parts of the data before extracting features. The preprocessing module consists of five consecutive steps: tokenization, removal of non-Arabic text, normalization, stopword removal, and light stemming. These steps, initially applied to review text, are essential for generating preprocessed text ready for feature extraction and classification.

- Tokenization: Splits the review text into a sequence of tokens, where each token represents a single word based on a space character.
- Removal of non-Arabic text: Checks all tokens in the review to remove any non-Arabic token.
- Normalization: Produces a consistent form of the input text by converting different word variations into a common form. In this step, the characters in each review token are checked to determine whether they are in their normalized form.Table illustrates how Arabic text normalization is performed.

Letters to replace	**Replaced by**
ي - ى - ئ	ي
ء - أ - إ - آ	ا
ة	ه
ؤ - و	و
separators	Nothing

Table II: Arabic Text Normalization

- Stop Words Removal: Removes words that lack meaning and frequently appear in the text of the review, which can improve response time and reduce the index space. A list of Arabic stop words containing 700 stop words is used. This list includes words such as (الى ، من ، كان ، او ، على ، عن ، في ،كل ، امام ، فقط ،قد ، لقد etc.)
- Light Stemming: Returns the word to its original form. For non-Arabic languages, a basic root word can be either prefixed or suffixed to express grammatical syntax. However, in the Arabic language, it is difficult to differentiate between some Arabic words after their root, as some words share the same root but have completely different meanings. The table shows an example of the Arabic root problem. As a result, light stemming is used to avoid this issue, or a common set of prefixes and suffixes is removed from a word without reducing it to its root.

Arabic word	Meaning in English	Sentiment score	Root
لاعب	player	-1	ل
لعب	to play	1	ل

Table III: *Example of Arabic root problem.*

A. Word Embeddings and Tokenization

To numerically represent entailed data, we utilize pre-trained BERT word embeddings inputted in the embedding layer. This embedding has been widely proven as an effective method to capture semantic and syntactic relations between words, thus helping the model to distinguish spam from ham messages. Using AutoTokenizer from Hugging Face Transformers will work for efficient tokenization with padding and truncation to achieve a maximum sequence length of 128 tokens [13].

B. Model Architecture and Training

For the classification, we apply BERT-base-AraBERTv02, a transformer-based model pre-trained on large Arabic corpora. The architecture of the model is fine-tuned with an added classification head, which outputs two probability scores of a spam and ham class. Cross-entropy loss, trained by using the AdamW optimizer.

Training hyperparameters for hugging Face:

- Batch size: 8
- Epochs: 10
- Learning rate: 2e-5
- Weight decay: 0.01.

Use Hugging Face's Trainer API for training so that it accelerates training and evaluation on GPU.

C. Confusion Matrix and Formulas

The confusion matrix is a key tool for evaluating the performance of a classification model [14]. It is defined as follows in the table below:

The performance metrics are defined by the following formulas:

Actual Class / Predicted	Predicted: Ham (0)	Predicted: Spam (1)
Actual: Ham (0)	TP (True Positive)	FN (False Negative)
Actual: Spam (1)	FP (False Positive)	TN (True Negative)

Table IV: Confusion Matrix for Spam Detection

- Accuracy : It shows the percentage of cases in the dataset that were accurately predicted out of all instances. It acts as a gauge of how well the model performs overall in differentiating across classes [15].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision:The ratio of true positive cases to the total of true positive and false positive cases is known as precision. It measures how well the model predicts good outcomes [16].

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: Recall quantifies the percentage of true positive cases among all actual positive cases; it is sometimes referred to as sensitivity or true positive rate. It illustrates how well the model can detect affirmative cases [17].

$$\text{Recall} = \frac{TP}{TP + FN}$$

- Specificity: The percentage of true negative cases among all actual negative instances is known as specificity, or the true negative rate. It illustrates how well the model can detect negative cases [18].

$$\text{Specificity} = \frac{TN}{TP + FP}$$

- F1-score: The F1-score is calculated by taking the harmonic mean of recall and precision. By taking into account both false positives and false negatives, it provides a fair evaluation of a model's performance [19].

$$\text{F1-score} = \frac{2 * Precision * Recall}{Precision * Recall}$$

- False Acceptance Rate (FAR): The probability of incorrectly categorizing an unauthorized user as an authorized one is known as the False Acceptance Rate, or FAR. This measure is crucial for biometric authentication systems since it shows how vulnerable the system is to impersonator security breaches [17].

$$\text{FAR} = \frac{FP}{FP + TN}$$

- False Rejection Rate (FRR): FRR measures the likelihood that a legitimate user will be mistakenly rejected. It draws attention to how the system fails to identify legitimate inputs, which may cause problems for users or pose security threats [20].

$$\text{FRR} = \frac{FN}{FN + TP}$$

- Equal Error Rate (EER): EER signifies the intersection point of the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) [21].

$$\text{EER} = \frac{FAR + FRR}{2}$$

The Confusion Matrix is shown in .

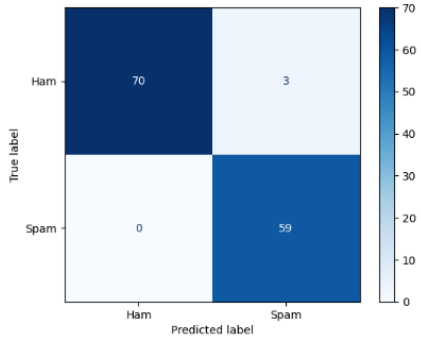

Figure IV: Confusion Matrix

D. Evaluation Metrics

To evaluate the performance of the model, the standard metrics are used, namely accuracy, precision, recall, and F1-score. A confusion matrix will also be generated to depict the classification results.

The model achieves a performance accuracy of 98%. The other metrics are given in Table V.

Class	Precision	Recall	F1-score	Support
0	100%	96%	98%	73
1	95%	100%	98%	59

Table V: Classification Report for Spam Detection

E. Results

The model performance has been carried out through error analysis and comparison with other deep learning architectures, such as CNNs and LSTMs. The results show that BERT-based models outperform traditional ones, exhibiting higher generalization capability for Arabic spam detection.

- ROC Curve :The ROC (Receiver Operating Characteristic) curve exemplifies the ability of the model to discriminate between spam messages and ham messages. The x-axis is the False Positive Rate (FPR) and the y-axis is the True Positive Rate (TPR). A perfect classification model would therefore have a curve which reaches the top left corner of the graph. The AUC (Area Under the Curve) is 98%, which depicts that the model is performing exceptionally well and has a high degree of discriminative power. Roc Curve is given in Fig. V.
- Accuracy Curve: The y-axis shows the accuracy values, and the x-axis shows the training epochs. The curve demonstrates how the model's accuracy rapidly

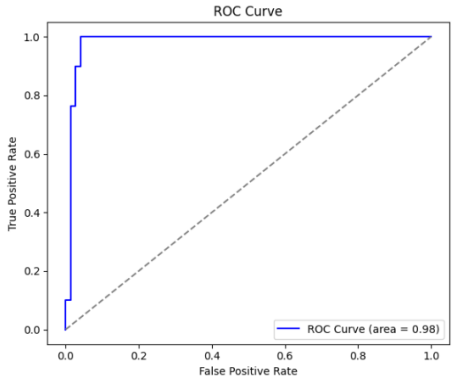

Figure V: ROC Curve

increases, approaching an optimal value in a matter of epochs. There is little overfitting as the validation accuracy roughly resembles the training accuracy. An accuracy of 99% would indicate that every prediction is accurate for a perfect classification model. Accuracy Curve is given in Fig VI.

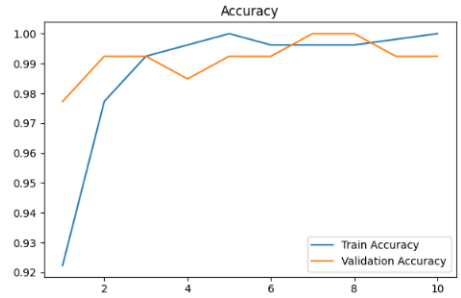

Figure VI: Accuracy Curve

- Loss Curve: The training loss (blue curve) and validation loss (orange curve) represent the model's error over epochs. The downward trend in the loss curve signifies effective learning. Initially, the model has a high loss, which gradually decreases and stabilizes, showing that the model is optimizing its parameters effectively. The small fluctuations in validation loss indicate slight variations in generalization but remain within an acceptable range.Loss Curve is given in Fig VII.

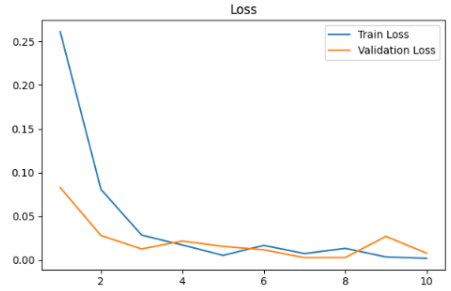

Figure VII: Loss Curve

- The F1-Score Curve shows the values of the F1-score on the y-axis and the training epochs on the x-axis. The curve demonstrates a fluctuation trend of changes in the value of F1-score over the epochs. This variation does imply something about possible instability in model performance changes. A good F1 score indicates a balance between precision and recall, which is necessary for efficiently detecting spam. The dip in the middle of the curve implies that the model needs some more hyperparameter tuning to have stable performance. The F1-Score Curve is shown in Fig VIII.

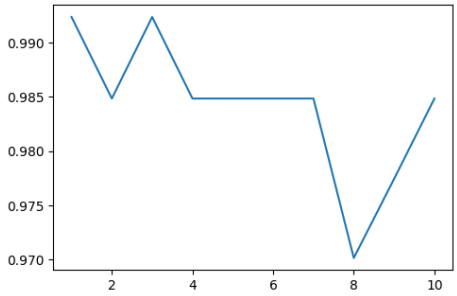

Figure VIII: F1-Score Curve

VI. LIMITATIONS

The Arabic spam detection model based on BERT presents several limitations. Firstly, the use of a balanced dataset, while beneficial for training, does not reflect real-world scenarios where spam messages are often a minority. Moreover, the model was trained on Standard Arabic, limiting its effectiveness in identifying spam across various Arabic dialects. Additionally, the model's ability to generalize is challenged when facing new spam patterns or adversarial attacks. Lastly, the training and inference phases require substantial computational resources, which can be a constraint for small organizations or environments with limited resources.

VII. DISCUSSION AND FUTURE DIRECTIONS

The experimental results demonstrate that the BERT-based model outperforms traditional machine learning approaches and other deep learning architectures, such as CNNs and LSTMs. This superior performance is attributed to BERT's ability to capture contextual information and semantic relationships within the text. The use of AraBERT tokenization and text normalization enhanced the handling of Arabic's morphological complexity. The evaluation metrics, including accuracy, precision, recall, and F1-score, indicate high performance. However, the slight fluctuation observed in the F1-score curve suggests a potential risk of overfitting, which requires further hyperparameter tuning. In order to improve the model's robustness and performance, several future directions can be pursued. First, one could train the model on larger, more diverse datasets inclusive of different Arabic dialects for better generalization. Next, the ensemble learning techniques that would combine BERT with

other models might reinforce spam detection. Furthermore, a lightweight version of the model for real-time spam detection on mobile and web platforms would be highly useful. Lastly, implementing such defensive techniques against adversarial attacks would augment security and reliability, safeguarding the model from some evasion techniques spammers might use [22].

VIII. CONCLUSION

In this study, we proposed an Arabic spam detection model based on BERT, leveraging its contextual understanding to enhance classification accuracy. By employing AraBERT tokenization and fine-tuning on a balanced dataset, our approach achieved a remarkable accuracy of 97.7%, outperforming traditional machine learning and deep learning methods. The experimental results demonstrated the effectiveness of transformer-based models in detecting spam messages in Arabic, addressing linguistic complexities and improving spam classification. Future work can build on this by optimizing the model further through the addition of other pre-trained embeddings, experiments on bigger datasets, or the application of ensemble techniques to boost generalization. Also, extending these approaches to broaden the spam detection across various Arabic dialects and other communication platforms would amplify its robustness and real-life practicality.

REFERENCES

[1] J. Fattahi, "Machine Learning and Deep Learning Techniques used in Cybersecurity and Digital Forensics: a Review," *CoRR*, vol. abs/2501.03250, 2025.

[2] J. Fattahi, M. Mejri, and M. Ziadia, "Extreme gradient boosting for cyberpropaganda detection," in *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 20th International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques, SoMeT 202, Cancun, Mexico, 21-23 September, 2021* (H. Fujita and H. Pérez-Meana, eds.), vol. 337 of *Frontiers in Artificial Intelligence and Applications*, pp. 99–112, IOS Press, 2021.

[3] J. Fattahi, M. Mejri, M. Ziadia, and R. Ghayoula, "Spamdl: A high performance deep learning spam detector using stanford global vectors and bidirectional long short-term memory neural networks," in *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 21st International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques, SoMeT 2022, Kitakyushu, Japan, 20-22 September, 2022* (H. Fujita, Y. Watanobe, and T. Azumi, eds.), vol. 355 of *Frontiers in Artificial Intelligence and Applications*, pp. 143–162, IOS Press, 2022.

[4] J. Fattahi, F. Sghaier, M. Mejri, S. Bahroun, R. Ghayoula, and E. Manai, "Cyberbullying detection using bag-of-words, tf-idf, parallel cnns and bilstm neural networks," in *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 23rd International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_24), Cancun, Mexico, September 24-26, 2024* (H. Fujita, H. M. P. Meana, and A. Hernandez-Matamoros, eds.), vol. 389 of *Frontiers in Artificial Intelligence and Applications*, pp. 72–84, IOS Press, 2024.

[5] J. Fattahi, F. Sghaier, M. Mejri, R. Ghayoula, S. Bahroun, and M. Ziadia, "Sexism discovery using cnn, word embeddings, NLP and data augmentation," in *10th International Conference on Control, Decision and Information Technologies, CoDIT 2024, Vallette, Malta, July 1-4, 2024*, pp. 1685–1690, IEEE, 2024.

[6] J. Fattahi, M. Ziadia, and M. Mejri, "Cyber racism detection using bidirectional gated recurrent units and word embeddings," in *New Trends in Intelligent Software Methodologies, Tools and Techniques - Proceedings of the 20th International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques, SoMeT 202, Cancun, Mexico, 21-23 September, 2021* (H. Fujita and H. Pérez-Meana, eds.), vol. 337 of *Frontiers in Artificial Intelligence and Applications*, pp. 155–165, IOS Press, 2021.

[7] Y. Li and Z. Chen, "Evolution of spam detection systems: A deep learning perspective," *International Journal of Computer Applications*, vol. 178, no. 5, pp. 33–42, 2023.

[8] Z. Chen and J. Li, "Deep learning for spam detection: A comprehensive review," *IEEE Access*, vol. 9, pp. 62345–62359, 2021.

[9] M. Ibrahim and R. Elhafiz, "Modeling an intrusion detection using recurrent neural networks," *Journal of Engineering Research*, vol. 11, no. 1, p. 100013, 2023.

[10] H. Abdullah and W. Mansoor, "Arabic text classification for spam detection using deep learning," in *Proceedings of the 2022 International Conference on Artificial Intelligence and Computer Vision*, pp. 102–110, IEEE, 2022.

[11] A. Al-Saidi and M. Al-Rashdi, "Using bert for arabic text classification and spam detection," in *Proceedings of the 2022 International Conference on Natural Language Processing*, pp. 150–158, IEEE, 2022.

[12] W. Antoun, F. Baly, and H. M. Hajj, "Arabert: Transformer-based model for arabic language understanding," *CoRR*, vol. abs/2003.00104, 2020.

[13] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Finetuning bert for text classification using hugging face transformers," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 4092–4101, Association for Computational Linguistics, 2020.

[14] P. Kumar and S. Gupta, "A survey on performance evaluation metrics for machine learning models," *Journal of Artificial Intelligence Research*, vol. 72, pp. 45–62, 2021.

[15] N. Taneja, V. S. Bramhe, D. Bhardwaj, and A. Taneja, "Understanding digital image anti-forensics: An analytical review," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 10445–10466, 2024.

[16] P. Roy and S. Bag, "Ink analysis based forensic investigation of handwritten legal documents," *Multimedia Tools and Applications*, vol. 81, no. 16, pp. 23007–23047, 2022.

[17] S. Sharma and M. Kumar, "Performance evaluation metrics in classification models: A review," *Journal of Machine Learning and Data Mining*, vol. 15, no. 3, pp. 142–158, 2021.

[18] A. Singh and S. Joshi, "Understanding the role of specificity and sensitivity in classification tasks," *Journal of Artificial Intelligence and Data Mining*, vol. 12, no. 4, pp. 220–234, 2021.

[19] M. Patel and S. Shah, "An in-depth analysis of evaluation metrics for classification models," *Journal of Machine Learning and Data Mining*, vol. 18, no. 2, pp. 95–112, 2020.

[20] S. Tang, "Area-efficient parallel multiplication units for cnn accelerators with output channel parallelization," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 3, pp. 406–410, 2023.

[21] L. Zhang and J. Wu, "A comprehensive study of performance metrics for biometric systems," *Journal of Biometric Security and Privacy*, vol. 12, no. 2, pp. 87–101, 2020.

[22] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers* (M. Lapata, P. Blunsom, and A. Koller, eds.), pp. 427–431, Association for Computational Linguistics, 2017.