FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 8 No. 6, December, 2024, pp 170 - 176
DOI: https://doi.org/10.33003/fjs-2024-0806-2799

MACHINE LEARNING ALGORITHMS FOR TELEGRAM SPAM FILTERING

***[1]Hassan, A., [1]Ayuba, Y., [2]Wajiro, M. A. and [1]Ahmed, M. Z.**

[1]Department of Computer Engineering, University of Maiduguri, Maiduguri, Borno State.
[2]Directorate of Information and Communications Technology, Ramat Polytechnic, Maiduguri.

*Corresponding authors' email: abubakarhassan@unimaid.edu.ng

ABSTRACT

With unprecedented usage of social media applications to interact in virtual communities, bad entities can now use these platforms to spread their malicious activities such as spam, hate speech, and even phishing to a very large population. Especially, Telegram is suitable for these kinds of activities because it is a new instant messaging (IM) application which is becoming increasingly used by bloggers and social media users today around the world that was developed in 2013 Pavel Durov. As a result, it becomes necessary for social media platforms to develop algorithms to filter these malicious contents. This paper employs Machine learning algorithms to filter spam messages in Telegram. Experiments were carried out in Jupyter Notebook (Python 3) environment using dataset obtained from Kaggle. Five machine learning models were applied, namely, Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), CatBoosting, Support Vector Machine (SVM) and K-Nearest Neighbours (KNN). Simulation results demonstrated that SVM Algorithm obtains superior performance than the other machine learning techniques employed for the study and achieved a classification accuracy of 94%. This shows that SVM model proves promising for Spam filtering task in Telegram if adopted.

Keywords: Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), CatBoost, Support Vector Machine (SVM), k-Nearest Neigbour(kNN), Telegram Spam Filtering

INTRODUCTION

Telegram, Facebook and other social media applications have provided great platforms for social interactions, information gathering and sharing for the benefit of humanity. Despite this, such unprecedented convenience supports activities of bad agencies which include virus spreading, malicious messages, fake news and fraudulent link farming (Alkadri et al., 2022).

In the context of social media, spam can be seen as fraudulent, undesirable, or irrelevant messages which include microblogs, fraudulent links, contents, fake connections, scams, etc. Spammers can generate money by directly spreading fake news to advertise business services and products or gain popularity through creating connections with other social media users. Spam messages distort the quality of communication services offered by social networking platforms. It corrupts social interactions and pollutes human perceptions of online messages. The user experience will significantly reduce when people are exposed to extremely unwanted messages, that can lead to loss of subscribers for online social network companies. Hence, it is necessary for social media platforms to develop algorithms to filter spam messages (Alkadri et al., 2022) .

Machine Learning (ML) algorithms prove a promising approach to spam filtering in E-mails, SMS, and social media platforms, leveraging the computational capabilities to analyse complex data and identify patterns embedded in them. Digital technology companies such as Google and Microsoft have employed ML algorithms for e-mail spam filtering. However, their performance in Telegram platform remains underexplored (Dada et al., 2019; Hassan et al., 2024). Telegram is one of the new cloud-messenger which is becoming increasingly popular among media users and bloggers around the world. It was developed in 2013 by Pavel, and it has good communication features such as security and anonymity. Hence, this paper aims to investigate the efficiency of machine learning approaches for spam filtering in Telegram platform.

Related Works

Some recent research on spam filtering in social media platforms using machine learning approaches are presented in this section.

Authors in (Balfagih et al., 2022), employed machine learning models to detect spam on Saudi tweets datasets in Arabic. They used eight (8) Twitter datasets to train and evaluate five different machine learning algorithms in WEKA environment: Naive Bayes (NB), K-Nearest Neighbour(K-NN), Random Forest (RF), Multi-Layer Perceptron (MLP) neural network, and WisSARD. The RF algorithm demonstrated superior performance compared to other machine learning techniques employed.

Different machine learning models were used, such as Support Vector Machine (SVM), Naive Bayes, and Logistic Regression for detection of Arabic spam on the Twitter platform by (Alkadri et al., 2022). Experimental results show that the SVM technique outperformed the other machine learning models used.

Ghanem & Erbay, (2020) proposed a BERT model which is based on a context-dependent representation of text. Twitter datasets were used to test the model. Simulation results show that the proposed technique performs better than conventional weighting techniques, traditional word embedding based algorithms as well as the existing state of the art models used for twitter spam detection.

Dar et al., (2023) developed a machine learning model for policy-based Urdu tweet spam detection. The model consists of TF-IDF, Count Vectorizer, and classifiers such as multinomial naïve Bayes, support vector classifier, RBF, logical regression, and BERT. Experimental results demonstrated that the logistic regression model has obtained the highest accuracy, with an F1-score of 0.70 and an accuracy of 99.55%.

Researchers in (Hassan et al., 2024), proposed an ensemble machine learning model to detect spam in Telegram platform using Random Forests and Logistic Regression as base learners. Experimental results showed that the proposed

ensemble model and the Random Forests algorithm achieved 94% accuracy compared to the Logistic Regression model (93%) on the benchmark dataset.

These research works demonstrate the importance of leveraging machine learning techniques to detect spam messages in social media platforms. Despite that much progress has been made in this direction, the performance of ML algorithms in the context of Telegram is still underexplored (Hassan et al., 2024). Hence, this paper seeks to apply machine learning algorithms for spam filtering in Telegram platform.

MATERIALS AND METHODS

Dataset Description

The Telegram spam dataset used in this study was obtained from Kaggle. The dataset contains 20,000 messages which can be classified into spam or ham (70-30%).

Data Preprocessing

Techniques were employed to remove noise from the dataset that could affect the system's accuracy and perform data cleansing. The techniques include Tokenization, normalization, removing repeated chars, removing punctuations, removing stop words. Pre-processing cleans and normalises the text data to ensure that it is in a consistent format.

Feature Extraction

Extracting suitable features is the first step in employing machine learning algorithm for classification problems. This facilitates the identification of spam contents and their transformation into numerical feature vectors. Content features that represent the text included in the Telegram messages were extracted. Language features are used, namely Term-Frequency-based (TFIDF). TFIDF is the most popular feature extraction technique. It normally transforms all sentences as a vector of term frequencies (TF) and assigns a score for each word in the text based on the number of times its occurrence and the probability it can be found in texts. The relative importance of a term in a document compared to other words in the corpus can be shown by TF-IDF.

Machine Learning Algorithms

Machine learning (ML) is a branch of mathematics, computing and statistics which deals with the design of algorithms that can learn(Dada et al., 2022; Hassan et al., 2024). Five (5) different ML algorithms were employed in this study which include Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), CatBoost, Support Vector Machine (SVM) and K-Nearest Neighbor(K-NN).

Extreme Gradient Boosting (XGB)

XGB is a tree-based ensemble algorithm which employs a gradient boosting machine learning technique for accomplishing regression and classification tasks. XGB uses level0algorithms to grow trees. It is different from the RF algorithm in the way it grows, orders, and combines the results. It employs a variety of algorithms for split finding. Trees grow in leaf-wise manner when histogram is used. The method works by bucketing features values into group of bins to construct features in histogram. The splitting is performed on the bins instead of on the features. The bucket bins are constructed before each tree is built. As a result, it speeds up the training which in turn reduces the computation complexity(Alzamzami et al., 2020).

Light Gradient Boosting Machine (LGBM)

LightGBM is a fast, distributed, high-performance gradient boosting technique developed by Microsoft Inc based on decision tree algorithm. It is applied to solve ranking and classification other machine learning problems. LightGBM is essentially an ensemble algorithm which combines the predictions of multiple decision trees by adding them together to make the final prediction that generalizes better. It trains the multiple tree models in an additive manner, with each new tree model being trained to predict the residuals (i.e., errors) of the prior models. The LightGBM (LGBM) algorithm provides

built-in support for categorical features, eliminating the need for preprocessing or the use of one-hot encoding methods for categorical variables. Within the realm of LGBM, the optimisation of hyperparameters, the proficient management of categorical features, and the understanding of the impact of different parameters on model performance are frequently emphasised by researchers and practitioners. The use of this technique extends to a wide array of machine learning applications, among others, categorisation, statistical modelling and rating (Chen et al., 2019; Dada et al., 2024).

CatBoosting Ensemble Method

The CatBoost algorithm, also known as categorical boosting, belongs to the gradient boosting family within the domain of machine learning. The method was specifically designed to effectively handle categorical data, making it suitable for both quantitative and qualitative variables. CatBoost is a machine learning algorithm that has been designed with the specific purpose of efficiently handling categorical information. This eliminates the need for complex preprocessing methods such as one-hot encoding. The algorithm utilises various methodologies, including target encoding and ordered boosting, to proficiently manage categorical variables within its internal processes. Like other gradient boosting strategies, CatBoost builds an ensemble of decision trees in a sequential way to minimise the loss function. CatBoost has been purposefully developed with a focus on enhancing performance and optimising memory usage. The acceleration of the training process is achieved by employing methodologies such as oblivious trees. CatBoost integrates a variety of built-in regularisation methods to address the problem of overfitting. The model integrates both L1 and L2 regularisation techniques to effectively handle the intricacy of the system. While CatBoost often exhibits robust performance using its default configurations, it provides users with a wide range of hyperparameters that may be customised to accommodate the distinct attributes of their datasets and goals. Commonly employed attributes include the pace of learning, the depth of trees, and the number of trees (Dada et al., 2024).

Support Vector Machine (SVM)

SVM algorithm is employed for both linear and nonlinear data classification. It uses a nonlinear mapping to convert the primary training set into an upper-level size. SVM explores for the linear optimal separating hyperplane in this new size as a decision border by which the tuples of one class from another are being split. The data from two classes can be separated by a hyperplane which uses a proper nonlinear mapping to an upper dimension. This hyperplane is used to form support vectors that are important training vectors and margins. Contrary to the other methods, they are highly robust for overfitting(Maikano, 2024; Oyewola & Dada, 2022).

K-Nearest Neigbour (K-NN)

The k-NN algorithm is a classic classification model based on the principle of nearest learning examples in the feature space. It learns by analogy which means the comparison of a provided test tuple with training tuples which are similar. These tuples must be the closest ones to the unknown tuple. A distance metric like Euclidean distance describes the "closeness". To classify *k*-nearest neighbour, the tuple that is not known is selected as the most common class among its *k*-nearest neighbours. The rate of *k* can be determined experimentally. The dataset is used by the k-NN to fill a sample of the search space with only instances of a known class. For this reason, this algorithm is referred to as a "lazy learning" algorithm. Since the k-NN algorithm is a slow learner, it does not have a training stage, and when it does, it completes it in a short amount of time. The testing stage, on the other hand, is time and memory intensive. Despite the need for a training dataset, no definite learning or model formation occurs during the training stage. (Dada et al., 2022; Oyewola & Dada, 2022).

Experimental Settings

The experimental and parameter settings for the study are shown in Table 1. The process of training a model entails the selection of appropriate values for each weight and bias parameter based on labelled samples. These factors play a crucial role in refining the effectiveness of the model. The models were trained with the pandas, NumPy and scikit-learn tools for machine learning computation in Python.

Table 1: Experimental Settings and Parameter Tuning of XGB, LGBM, CatBoost, SVM and k-NN.

Model	Hyperparameter	Values
XGBoost	n_estimators learning_rate	100
	max_depth	0.1
		5
LightGBM	n_estimators learning_rate	100
	max_depth	0.1
		5
CatBoost	n_estimators learning_rate	100
	max_depth	0.1
	verbose	5
		0
SVM	kernel	linear
	probability	True
K-NN	n_neighbors	5

Evaluation Metrics

Standard evaluation metrics were used to measure the performance of the proposed machine learning techniques such as accuracy, F1-score, recall, and precision. True Positive (TP) indicates the number of messages that are classified correctly into the spam class. The definitions of other quantities like True negative (TN), False positive (FP), and False negative (FN) are derived from TP. Accuracy, precision, recall, and F1-score were computed by the formula given the equations below.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

RESULTS AND DISCUSSION

This section presents the results and examines the significant discoveries derived from the simulations. The experiment was carried out in Python (jupyter Notebook) environment. The Telegram spam dataset was used for the purpose of training and testing the classifiers.

Performance Metrics

As seen from Table 2, it can be noticed that XGB, LGBM and CatBoost models achieved better precision results for spam and better recall scores for ham. While SVM obtained superior F-1 score for spam and ham respectively and better accuracy result. It can also be seen that *k*-NN was struggling in all performance metrics: classification accuracy, precision, recall and *F*1-Score.

Table 2: Classifiers' performance on the Telegram Spam dataset.

Classifier		Precision	Recall	F-Measure	Accuracy
XGBoost	ham	0.89	0.98	0.93	0.90
	spam	0.94	0.70	0.80	
LGBM	ham	0.89	0.98	0.93	0.90
	spam	0.94	0.69	0.80	
CatBoost	ham	0.89	0.98	0.93	0.90
	spam	0.94	0.69	0.80	
SVM	ham	0.95	0.97	0.96	0.94
	spam	0.92	0.87	0.89	
K-NN	ham	0.79	0.85	0.82	0.73
	spam	0.54	0.43	0.48	

Confusion Matrix

A confusion matrix is a table which provides the performance summary of a machine learning classification algorithm by comparing its predicted labels to the true labels. It shows the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) of the classifier's predictions. The confusion matrices of XGB, LGBM, CatBoost, SVM and k-NN are depicted in Figures (1- 5) respectively. It is clear that XGB, LGBM, and CatBoost performed better for ham prediction. While SVM achieved superior performance for spam prediction.

Figure 1: Confusion Matrix of XGB

Figure 2: Confusion Matrix of LGBM

Figure 3: Confusion Matrix of CatBoost

Figure 4: Confusion Matrix of SVM

Figure 5: Confusion Matrix of k-NN

ROC Curve

The performance analysis of the ML models is also illustrated using ROC curves to provide insight into the trade-offs between sensitivity (recall) and specificity. It plots True Positive Rate (Recall) against the False Positive Rate (1-Specificity). The Area Under the Curve (AUC) indicates the extent of separability and measures how good a model is at classifying between positive and negative classes. Figures (6-10) illustrate the ROC of all the models. The AUC values fall within the interval between 0 and 1. The closer the value is to one, the more intelligent the model. SVM obtained better results than the other models while k-NN performed the least.

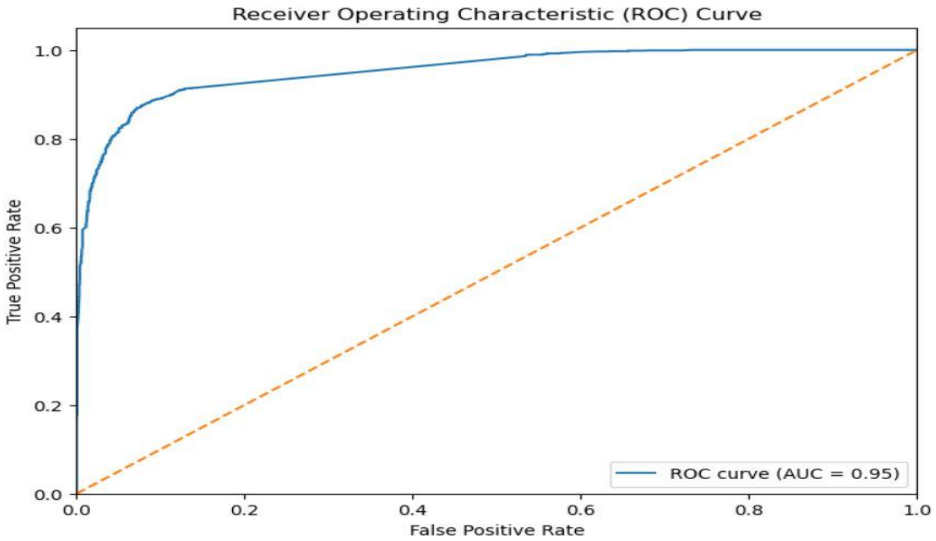

Figure 6: ROC Curve of XGM

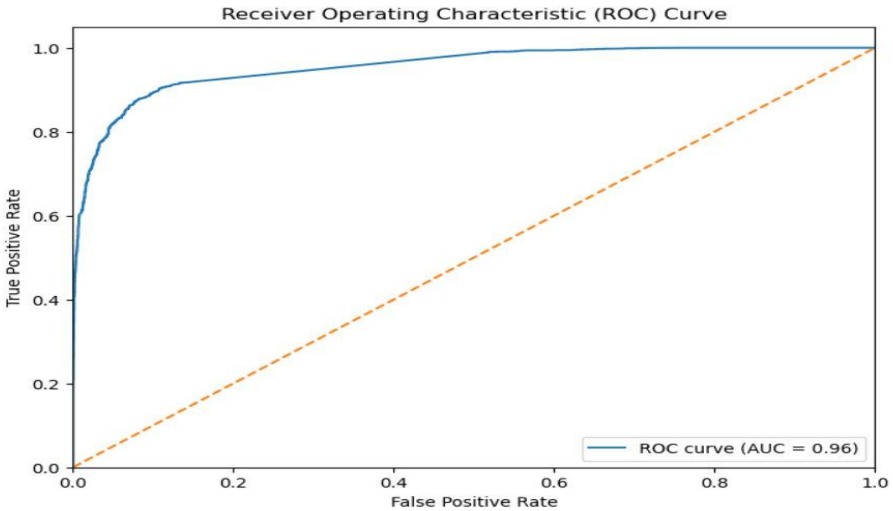

Figure 7: ROC Curve of LGBM

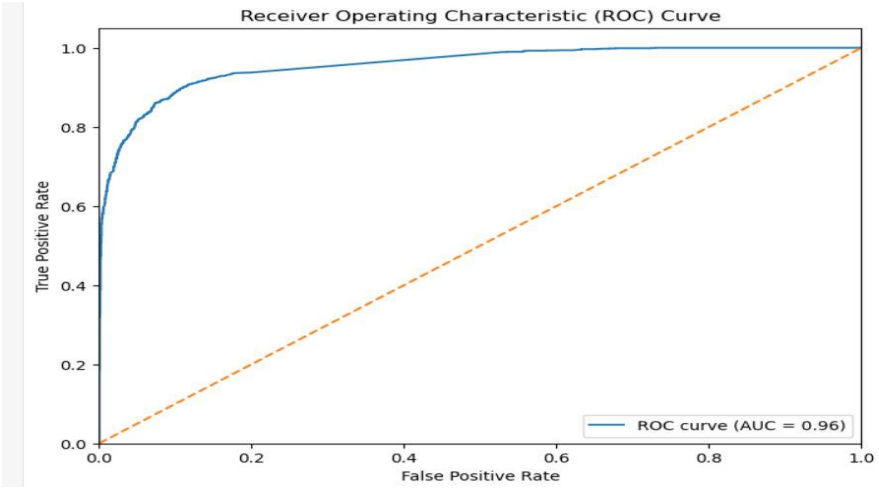

Figure 8: ROC Curve of CatBoost

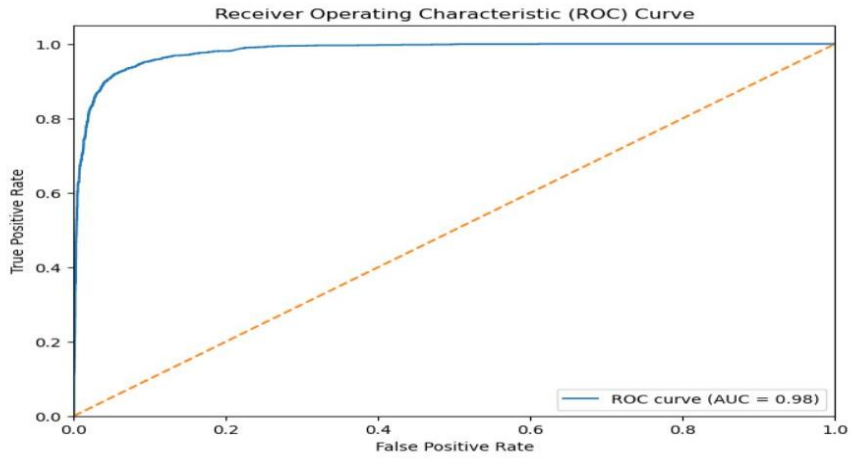

Figure 9: ROC Curve of SVM

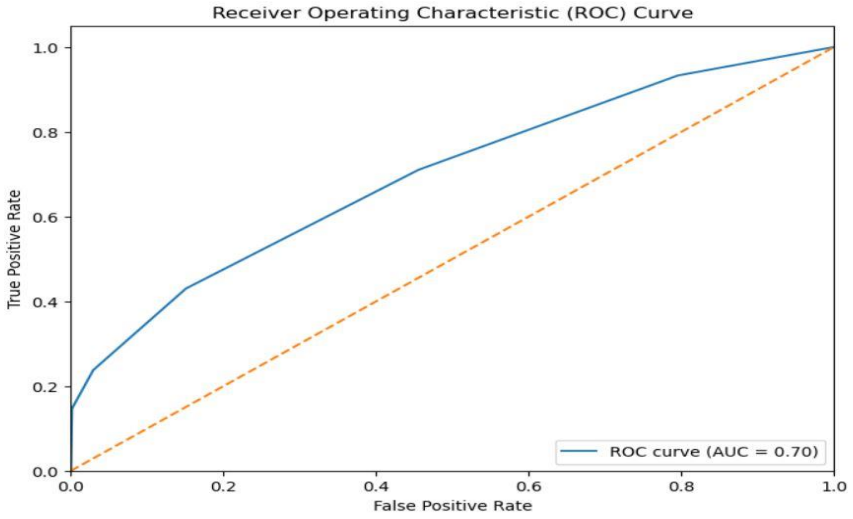

Figure 10: ROC Curve of k-NN

CONCLUSION

Machine learning models for filtering spam messages in Telegram platforms have been proposed in this paper. The Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), CatBoosting, Support Vector Machine (SVM) and k-Nearest Neighbours (kNN) models were employed using dataset obtained from Kaggle in Jupyter Notebook (Python3) environment. Experimental results illustrate the superiority of SVM compared to the other algorithms used for Telegram Spam Filtering applied in this study. Simulations data illustrate that SVM proves promising technique which can be employed for Telegram spam filtering. Future work will focus on improving the classification accuracy of SVM model by integrating soft computing and computational intelligence algorithms such as Whale Optimization Algorithm (WOA).

REFERENCES

Alkadri, A. M., Elkorany, A., & Ahmed, C. (2022). Enhancing Detection of Arabic Social Spam Using Data Augmentation and Machine Learning. *Applied Sciences (Switzerland)*, *12*(22). https://doi.org/10.3390/app122211388

Alzamzami, F., Hoda, M., & Saddik, A. El. (2020). Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation. *IEEE Access*, *8*, 101840–101858. https://doi.org/10.1109/ACCESS.2020.2997330

Balfagih, A. M., Keselj, V., & Taylor, S. (2022). N-gram and Word2Vec Feature Engineering Approaches for Spam Recognition on Some Influential Twitter Topics in Saudi Arabia. *Journal of Advances in Information Technology*, *13*(6), 562–568. https://doi.org/10.12720/jait.13.6.562-568

Chen, T., Xu, J., Ying, H., Chen, X., Feng, R., Fang, X., Gao, H., & Wu, J. (2019). Prediction of Extubation Failure for Intensive Care Unit Patients Using Light Gradient Boosting Machine. *IEEE Access*, *7*, 150960–150968. https://doi.org/10.1109/ACCESS.2019.2946980

Dada, E. G., Bassi, J. S., Chiroma, H., Abdulhamid, S. M., Adetunmbi, A. O., & Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, *5*(6). https://doi.org/10.1016/j.heliyon.2019.e01802

Dada, E. G., Birma, A. I., & Gora, A. A. (2024). Ensemble machine learning algorithm for cost-effective and timely detection of diabetes in Maiduguri, Borno State. *Journal of the Nigerian Society of Physical Sciences*, 2175. https://doi.org/10.46481/jnsps.2024.2175

Dada, E. G., Oyewola, D. O., & Yakubu, J. H. (2022). Power Consumption Prediction in Urban Areas using Machine Learning as a Strategy towards Smart Cities. *Arid Zone Journal of Basic and Applied Research (AJBAR)*, *1*(1), 11–24.

Dar, M., Iqbal, F., Latif, R., Altaf, A., & Jamail, N. S. M. (2023). Policy-Based Spam Detection of Tweets Dataset. *Electronics (Switzerland)*, *12*(12). https://doi.org/10.3390/electronics12122662

Ghanem, R., & Erbay, H. (2020). Context-dependent model for spam detection on social networks. *SN Applied Sciences*, *2*(9). https://doi.org/10.1007/s42452-020-03374-x

Hassan, A., Abatcha, M., & Dada, E. G. (2024). Ensemble Machine Learning Algorithm for Telegram Spam Detection. *Arid-Zone Journal of Basic & Applied Research*, *3*(4), 87–95. https://doi.org/10.55639/607.060504

Maikano, F. A. (2024). MACHINE LEARNING APPROACHES FOR CYBER BULLYING DETECTION IN HAUSA LANGUAGE SOCIAL MEDIA: A COMPREHENSIVE REVIEW AND ANALYSIS. *MACHINE LEARNING APPROACHES. FJS FUDMA Journal of Sciences (FJS*, *8*(3), 344–348. https://doi.org/10.33003/fjs-2024-0803-2517

Oyewola, D. O., & Dada, E. G. (2022). Machine Learning Methods for Predicting the Popularity of Movies. *Journal of Artificial Intelligence and Systems*, *4*(1), 65–82. https://doi.org/10.33969/ais.2022040105

©2024 This is an Open Access article distributed under the terms of the Creative Commons Attribution 4.0 International license viewed via https://creativecommons.org/licenses/by/4.0/ which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is cited appropriately.