Tweet Prediction for Social Media using Machine Learning

Mohammed Fattah
Department of Information Technology, College of Computer Sciences and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia
mail

Mohd Anul Haq
Department of Computer Science College of Computer Sciences and Information Sciences, Majmaah University, Al Majmaah, 11952, Saudi Arabia
m.anul@mu.edu.sa (corresponding author)

Received: 17 April 2024 | Revised: 25 April 2024 | Accepted: 30 April 2024

 DOI: https://doi.org/10.48084/etasr.7524

ABSTRACT

Tweet prediction plays a crucial role in sentiment analysis, trend forecasting, and user behavior analysis on social media platforms such as X (Twitter). This study delves into optimizing Machine Learning (ML) models for precise tweet prediction by capturing intricate dependencies and contextual nuances within tweets. Four prominent ML models, i.e. Logistic Regression (LR), XGBoost, Random Forest (RF), and Support Vector Machine (SVM) were utilized for disaster-related tweet prediction. Our models adeptly discern semantic meanings, sentiment, and pertinent context from tweets, ensuring robust predictive outcomes. The SVM model showed significantly higher performance with 82% accuracy and an F1 score of 81%, whereas LR, XGBoost, and RF achieved 79% accuracy with average F1-scores of 78%.

Keywords-tweet prediction; emotion analysis; machine learning; hyperparameter tuning

I. INTRODUCTION

Understanding sentiments and behaviors on platforms like X (Twitter) is essential for gaining insights into public opinion and trends. However, predicting accurately tweets that capture these sentiments and emotions remains a challenging task. In the current study, focus is given on leveraging Machine Learning (ML) to address this challenge. The aim is to develop models capable of capturing the deep meanings and emotions embedded in tweets, thereby enhancing the reliability of predictions. Tweet prediction is advanced with the use of ML models, providing insights into their performance strengths and limitations. Such advancements hold significant potential for businesses, researchers, and social media platforms in real-world applications. Proper data preprocessing and hyperparameter tuning were carried out to systematically evaluate four tweet prediction ML models. A detailed comparison of these models is given, highlighting their respective strengths and limitations, contributing to the advance of tweet prediction techniques and providing valuable guidance for businesses, researchers, and social media platforms interested in sentiment analysis and trend forecasting on Twitter.

II. LITERATURE REVIEW

In [1], the authors utilized two deep learning hybrid LSTM models, Word2vec LSTM and GLoVe LSTM and compared their performance with leading ML modes. Surprisingly, ML-based models outperformed their DL counterparts in detecting insider threats deploying real-world datasets, emphasizing the need for further exploration to overcome the existing limitations. Authors in [2] focus on improving the accuracy of Intrusion Detection Systems (IDS) for identifying DoS and DDoS attacks on IoT devices. Anomalies in the existing techniques prompted the adoption of a Convolutional Neural Network (CNN) for Enhanced Data rates for GSM Evolution (EDGE) computing. The proposed approach achieved remarkable accuracies of 99.34% and 99.13% for binary and multiclass classifications on the NSL-KDD dataset. Authors in [3] addressed the security and privacy risks associated with the growth of Internet of Things (IoT) devices, stressing the vulnerability to botnet attacks. The study introduces DBoTPM, a novel deep-neural network-based model optimized for performance and computational efficiency through hyperparameter tuning and dropout techniques. Evaluation results demonstrate DBoTPM's effectiveness in predicting botnet attacks, utilizing two real datasets for enhanced accuracy and faster response. Authors in [4] discussed the security challenges introduced by the widespread use of IoT devices, acknowledging their benefits in reducing human efforts and increasing resource efficiency. They developed two innovative Deep Neural Network (DNN) models, DNNBoT1 and DNNBoT2, specifically designed to detect and classify well-known IoT botnet attacks like Mirai and BASHLITE.

Leveraging Principal Component Analysis (PCA) for feature extraction, they aimed to enhance the accurate classification of botnets in IoT environments employing proper data preprocessing and hyperparameter tuning. Their thorough evaluation revealed that these DNN models outperformed others in terms of accuracy and efficiency, based on proper data preprocessing and optimization with rigorous hyperparameter tuning [5-8]. Authors in [9] aimed to develop an ML model that accurately predicts the geographic location of Twitter users based on their tweets. Potential applications, such as demographic analysis, targeted advertising, and personalized user experiences were considered. The primary objectives included proposing and evaluating various ML algorithms for location prediction and assessing their performance using metrics like accuracy, precision, recall, and f1-score. The results revealed that the Random Forest (RF) classifier demonstrated the best performance metrics for a specific dataset, making it the most suitable algorithm for predicting the geographic location of Twitter users. Authors in [10] analyzed the correlation between traffic volume and tweet counts, proposing an optimization framework that extracts traffic indicators based on tweet semantics. These indicators were then incorporated into traffic prediction engaging Linear Regression (LR). The experimental results, utilizing data from the San Francisco Bay area of California, demonstrated the effectiveness of the proposed framework, highlighting the novel approach of integrating social media data for enhancing longer-term traffic prediction.

Author in [11] explored the widespread economic crime of fraud and deception in contemporary societies, both Western and Arab. Focus was placed on the intelligence of fraudulent criminals and the characteristics of victims, particularly in Algerian society. Authors in [12] delved into the dual nature of social media platforms, serving as a convenient means for daily communication and business transactions while also becoming a breeding ground for criminal activities. Focusing on WeChat, utilized by 846 million users globally, the study aimed to provide a thorough overview of fraud occurrences in China, particularly within the WeChat application.

III. METHODS

A. Used Libraries and Tools

The investigation made use of a set of libraries to support its analysis. The Pandas library was employed for managing and exploring datasets. Scikit-learn was utilized for constructing and training the ML models. Visualizations, including histograms, were created deploying the Matplotlib library and Seaborn was applied to enhance the aesthetics of these visualizations. For tokenization purposes, TFAutoModel and AutoTokenizer from Transformers were specifically used. The Datasets module facilitated the effective handling of datasets and Sklearn train_test_split aided in splitting the data appropriately. TensorFlow's, EarlyStopping, and ExponentialDecay schedules were utilized during the model training phase. Cross-validation was conducted using KFold from Sklearn and the investigation incorporated GridsearchCV for hyperparameter search, exploring parameters like learning rate, max_depth, C, etc. for the different ML models. These libraries played integral roles in different aspects of the investigation, contributing to data management, model development, and hyperparameter optimization.

B. Data Preprocessing and EDA

The current investigation utilized a dataset from a Kaggle competition designed to introduce data scientists to NLP [14]. Focused on Twitter data, the competition participants tasked with distinguishing tweets about real disasters from those that were not. Each entry in the sets contained a tweet text, an associated keyword (possibly blank), and the tweet's location (possibly blank). The objective was to predict if a tweet was related to a real disaster, with 1 indicating yes and 0 indicating no. The challenge highlighted the difficulty in discerning literal from metaphorical language in emergency-related tweets. A dataset of 10,000 hand-classified tweets was provided to encourage the development of accurate ML models. The dataset comprised 5 columns, including 'id,' serving as a unique identifier for each tweet. The 'text' column contained the actual content of the tweets, providing textual information for analysis. The 'location' column denoted the geographical origin of the tweets, however, it could be left blank. Similarly, the 'keyword' column captured specific keywords associated with the tweets, but like the location, it might also be blank. The primary focus of the investigation was the 'target' column, indicating whether a tweet was related to a genuine disaster (designated as 1) or not (designated as 0). In the initial phase of the investigation, the dataset was loaded, and Exploratory Data Analysis (EDA) was conducted to understand the information better [1–4]. This involved checking for imbalances in target labels, addressing missing values and handling any duplicate data [5–8]. The dimensions of the dataset were 7613×5 for training and 3263×4 for testing. The number of the non-null values is displayed in Table I. The length of each text of the training data was analyzed with a histogram with 30 bins and a kernel density estimate (Figure 1). The count plot depicting the distribution of the target variable in the training data is shown in Figure 2.

TABLE I. INITIAL DATASET EDA SUMMARY

Data	ID	Keyword	Location	Text	Target
Train	7613	7552	5080	7613	int64
Test	3263	3237	2158	3263	Not Applicable (Test Data)

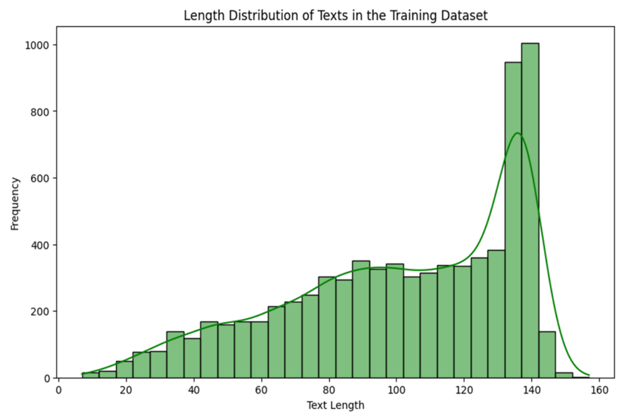

Fig. 1. The length of each text of the training data.

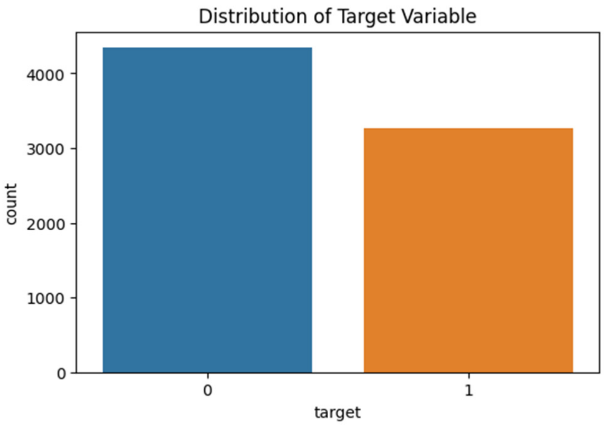

Fig. 2. The distribution of the target variable in the training data.

Initialization of the TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer using the TfidfVectorizer() was performed. The vectorizer was then fitted to the 'text' column of the 'train' DataFrame using fit(). Subsequently, the 'text' column in both training and test datasets were transformed into TF-IDF feature matrices utilizing transform(). The resulting TF-IDF features for training and testing were stored in the 'train_text_features' and 'test_text_features' variables, respectively. A histogram displaying the distribution of the text lengths in the 'train' data for 30 bins can be seen in Figure 3. The bars are stacked to represent different categories ('Not Disaster' and 'Disaster').

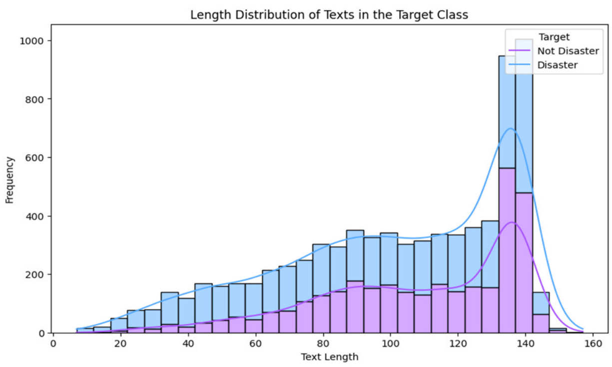

Fig. 3. Length distribution of texts in the target class.

A word cloud was created implementing the 'WordCloud' library, visualizing common words in the 'text_combined' variable, as detected in Figure 4. The plot was displayed with bilinear interpolation for smoother visuals. The LabelEncoder from the 'preprocessing' module was deployed to convert the 'target' column in the 'train' DataFrame into numerical labels. Then, data splitting was performed to split the data into training and validation sets with 20% of the data reserved for validation and a fixed random state for reproducibility. In the Hyperparameter Search (Section V), the investigation aimed to optimize the models' performance. The Keras Tuner library was employed to explore key hyperparameters, such as learning rate, dropout rate, and the number of units in dense layers. Different learning rate schedules were also considered for a comprehensive exploration of the hyperparameter space.

Fig. 4. The word cloud is generated from the text data.

C. The Models

1) Logistic Regression (LR)

Initially, LR, a widely used method for predictive analytics and classification tasks, was applied. LR calculates the likelihood of an event occurring based on a given dataset of independent variables. In this approach, the dependent variable ranges from 0 to 1, representing the outcome as a probability. To transform the odds, which is the probability of success divided by the probability of failure, the logit formula was employed:

$$Logit\ (P) = \frac{1}{1+\exp(-p)} \quad (1)$$

The logit function transforms the probability P into a log-odds scale, where the odds are the ratio of the probability of an event occurring to the probability of it not occurring. The logit transformation maps probabilities from the interval [0,1] to the entire real line, making it suitable for modeling binary outcomes in LR.

$$In\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k \quad (2)$$

where In is the natural logarithm, p is the probability of an event, $X_1, \ldots X_k$ are predictor variables, and $\beta 0, \beta 1, \ldots, \beta k$ are coefficients that determine the impact of each predictor variable.

2) Random Forest

In RF, Decision Trees (DTs) are generated using subsets of the dataset, building a diverse ensemble of DTs that collaborate to make predictions. Unlike a single DT, RF employs feature randomness by selecting a subset of features for each DT, ensuring diversity and reducing similarity among the trees. The prediction for an observation in RF is determined by the majority vote from all the individual trees. The final predicted class for an observation is the mode of the predicted classes from all the individual DTs. By aggregating predictions from multiple trees, RF enhances prediction accuracy and reliability compared to a single DT.

3) XGBoost

XGBoost, a fast and efficient gradient-boosted DT implementation, is achieved through the XGBoost package. It deploys gradient boosting, an ensemble technique that iteratively corrects errors made by earlier models. This process continues until no further improvements are achievable. AdaBoost is a notable example of boosting, emphasizing challenging data points in predictions. XGBoost distinguishes itself from other ML algorithms with its scalability, speed, and performance optimization. It efficiently handles large datasets using parallel processing and tree pruning, while its built-in support for missing values simplifies data preprocessing. XGBoost's regularization and feature importance ranking make it robust and versatile, rendering it a preferred choice for various ML tasks [7, 8].

4) Support Vector Machine (SVM)

SVM is a supervised learning algorithm primarily applied for classification tasks. It finds the optimal hyperplane that separates data into different classes, maximizing the margin between them. SVM can handle non-linear data by using kernel functions to transform the input space, making it effective for complex classification problems with intricate decision boundaries. To make sure the model works well and does not get too focused on the training data, this study adopted the Gridsearchcv method, which helped adjust settings in the model, like the regularization parameter (C), kernel type, and kernel coefficient (gamma) for SVM, as well as the learning rate, number of estimators, and maximum depth for XGBoost.

5) Hyperparameter Tuning

Hyperparameters play a critical role in determining the performance and behavior of ML models. Proper tuning and selection of hyperparameters are essential steps in the model development process to achieve optimal performance on unseen data. The present investigation also added penalties and constraints to LR. These changes aimed to find the right balance between making the model accurate on known data and ensuring it can do well on new, unfamiliar data. The present study opted for GridSearchCV, a greedy optimization-based approach, to tune hyperparameters due to its systematic and exhaustive search capabilities, ascertaining optimal model performance by exploring a wide range of parameter combinations.

TABLE II. HYPERPARAMETERS OF ALL MODELS

Model	Hyperparameters	Tuned Hyperparameters
LR	penalty: [l1, l2, elasticnet]	L2
	C: [0.001, 0.01, 0.1, 1, 10, 100]	0.1
	solver: [newton-cg, lbfgs, saga, liblinear]	newton-cg
	max_iter: [100, 500, 1000]	500
RF	n_estimators: [50, 100, 200]	150
	max_depth: [None, 10, 20]	10
	min_samples_split: [2, 5, 10]	5
	min_samples_leaf: [1, 2, 4]	2
XGBoost	learning_rate: [0.01, 0.1, 0.2]	0.01
	n_estimators: [50, 100, 200]	150
	max_depth: [3, 5, 7]	10
SVM	C: [0.1, 1, 10]	0.7
	kernel: [rbf, poly, linear]	poly
	gamma: [scale, auto]	auto

IV. EVALUATION METRICS

The performance of ML models in detecting fraud was evaluated utilizing four measures: precision, recall, f1-score, and accuracy. Accuracy refers to the fraction of fraud instances correctly recognized out of all instances. Recall is the proportion of accurately identified fraud instances compared to the total number of fraud samples. Precision is the ratio of correctly identified fraud instances to the total instances marked as fraud. Equations (3)-(5) provide specific calculations for accuracy, recall, and precision. [28]

$$\text{Accuracy}=\frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad (5)$$

where TP represents true positive, TN is true negative, FP is false positive, and FN is false negative. Accuracy measures the overall correctness of the model's predictions by considering both positive and negative classifications. Equation (3) expresses the ratio of correct predictions to the total number of instances. While accuracy provides a general measure of model performance, it may not be sufficient in cases of imbalanced datasets where additional metrics like precision, recall, and F1 are considered for a more comprehensive evaluation. Precision assesses the model's ability to accurately identify positive values among all samples predicted as positive. A high precision value indicates a low rate of false positives, highlighting the reliability of positive predictions made by the model. Recall measures the quantity of actual positive instances that the model successfully detects. A high recall value reveals that the model is effective at minimizing false negatives, essential in scenarios where missing positive instances is a significant concern. Recall is valuable in applications where the importance is on capturing as many positive instances as possible. F1-score is the harmonic mean of precision and recall, providing a balanced measure that considers both false positives and false negatives. It is particularly useful in binary classification settings where class imbalance is present, offering a single value that captures the model's accuracy in identifying both positive and negative instances.

$$\text{F1} = \frac{\text{Precision+Recall}}{\text{Precision} \times \text{Recall}} \quad (6)$$

V. RESULTS AND DISCUSSION

The LR model achieved an accuracy of 79%, with a precision of 77% for classifying non-disaster tweets (Class 0) and 85% for disaster tweets (Class 1), with corresponding recalls of 92% and 62%. Its F1-score was 84% for Class 0 and 72% for Class 1. The XGBoost model displayed similar accuracy and performance metrics, with precision, recall, and F1 scores of 78%, 82%, and 83% for Class 0, and 89%, 66%, and 73% for Class 1, respectively. RF achieved 79% accuracy, demonstrating precision, recall, and F1-scores of 76%, 93%, and 84% for Class 0, and 86%, 61%, and 71% for Class 1. SVM yielded the highest accuracy at 82%, with precision, recall, and F1-scores of 81%, 90%, and 85% for Class 0, and 84%, 71%, and 77% for Class 1. Overall, the SVM model exhibited superior performance in predicting disaster tweets among the evaluated models, see Figures 5-6.

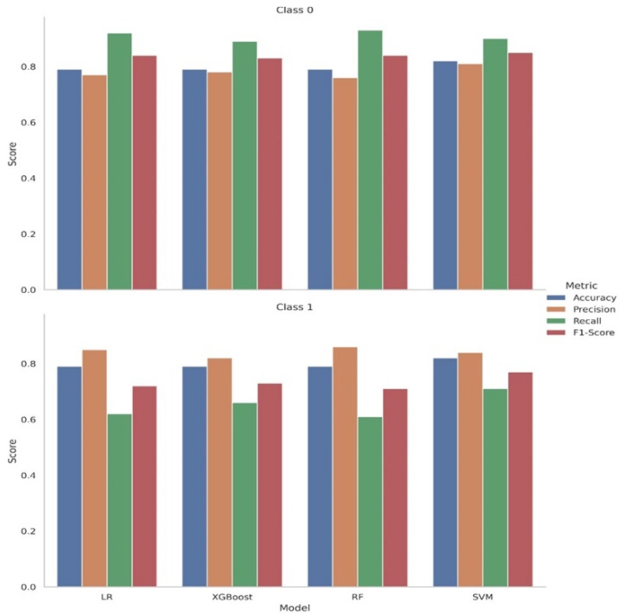

Fig. 5. Classwise performance of the four ML models in predicting disaster-related tweets.

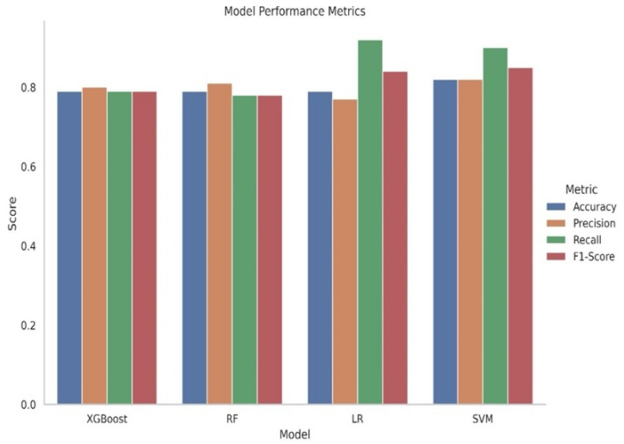

Fig. 6. Performance of the four ML models in predicting disaster-related tweets

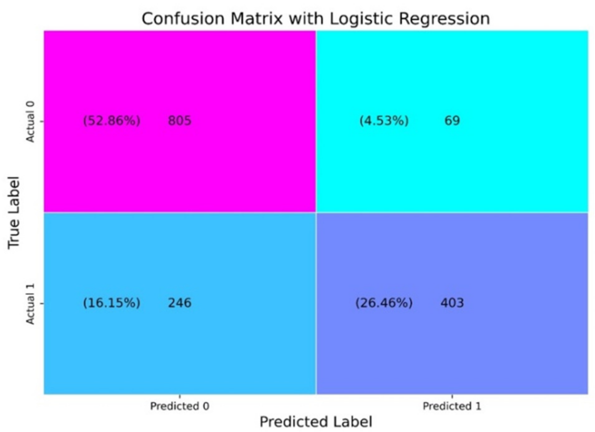

Fig. 7. Confusion matrix of the LR model for predicting disaster-related tweets.

In the confusion matrix analysis (Figures 7-10), and out of the 1523 instances, the LR model exhibited 805 TN, 69 FP, 246 FN, and 403 TP. The XGBoost model showed 779 TN, 95 FP, 220 FN, and 429 TP, while the RF demonstrated 811 TN, 63 FP, 254 FN, and 395 TP. Lastly, the SVM model manifested 787 TN, 87 FP, 189 FN, and 460 TP.

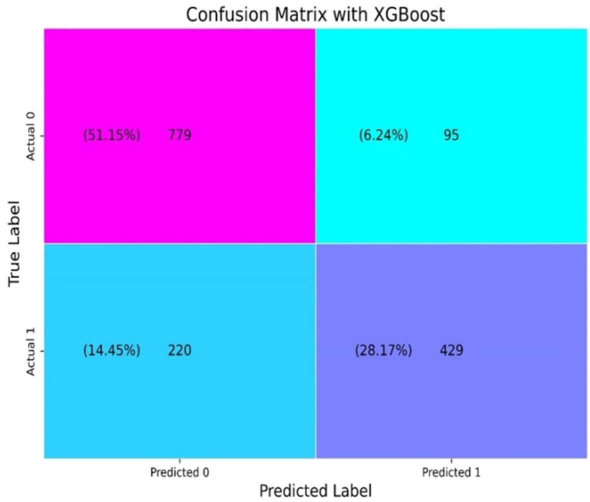

Fig. 8. Confusion matrix of the XGBoost model.

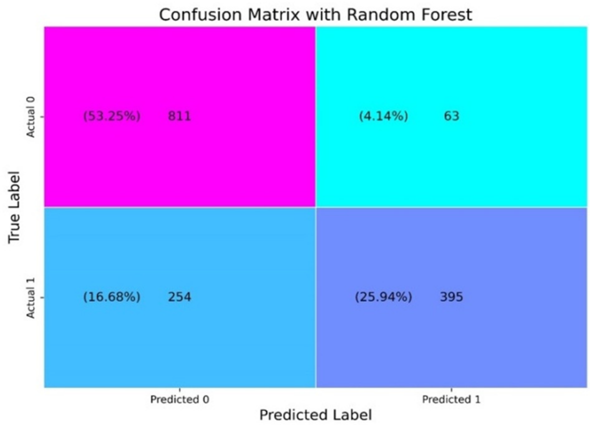

Fig. 9. Confusion matrix of the RF model.

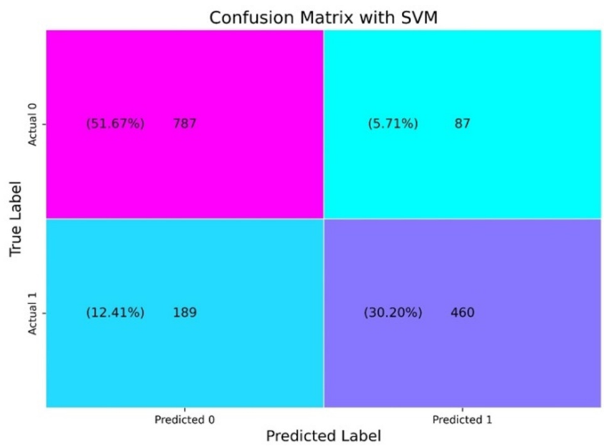

Fig. 10. Confusion matrix of the SVM model.

VI. LIMITATIONS AND STRENGTHS

The study has a few limitations that need to be considered. Firstly, since this study uses a dataset from a Kaggle competition, its findings might be influenced by the specific characteristics of that dataset like the tweet origin geolocation. Generalizing the results to different areas might require additional testing on various datasets. Also, this study's focus on distinguishing disaster-related tweets might not cover some cases where tweets could have both disaster-related and non-disaster-related elements. Moreover, the current study mainly explores traditional ML models, potentially missing out on the benefits of newer deep learning techniques. Not extensively experimenting with deep learning architectures might limit individuals' understanding of the full capabilities of these models. The computational efficiency of the models varies: LR and SVM are generally efficient with moderate computational demands, RF can be computationally intensive, and XGBoost strikes a balance with efficient boosting algorithms that optimize computational resources. In terms of scalability, LR and SVM scale well to large datasets, RF can be challenging due to its ensemble nature, while XGBoost is designed for efficient handling of large datasets.

VII. CONCLUSIONS

This study endeavored to develop models capable of discerning tweets related to real-world situations. Leveraging a Kaggle competition dataset and utilizing libraries like pandas and scikit-learn, this study's focus encompassed data preparation, inspection, and model construction. While the findings of this research showcase the efficacy of various ML models, especially SVM with an accuracy of 82%, it is crucial to note that performance might vary across different datasets. The absence of exploration into advanced deep-learning methods further highlights avenues for future research. Future endeavors should encompass testing these models across diverse datasets, integrating advanced techniques [12, 13, 15], and addressing the nuances posed by figurative language in tweets. Despite these limitations, this research serves as a foundational step toward proficiently predicting disaster-related tweets. Continual refinement remains imperative to adapt to the evolving landscape of social media and ML [16-22].

REFERENCES

[1] M. A. Haq, M. A. R. Khan, and M. Alshehri, "Insider Threat Detection Based on NLP Word Embedding and Machine Learning," *Intelligent Automation & Soft Computing*, vol. 33, no. 1, pp. 619–635, Jan. 2022, https://doi.org/10.32604/iasc.2022.021430.

[2] M. A. Haq, M. Abdul, and T. AL-Harbi, "Development of PCCNN-Based Network Intrusion Detection System for EDGE Computing," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1769–1788, 2021, https://doi.org/10.32604/cmc.2022.018708.

[3] M. A. Haq, "DBoTPM: A Deep Neural Network-Based Botnet Prediction Model," *Electronics*, vol. 12, no. 5, Jan. 2023, Art. no. 1159, https://doi.org/10.3390/electronics12051159.

[4] M. A. Haq and M. Abdul, "DNNBoT: Deep Neural Network-Based Botnet Detection and Classification," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1729–1750, 2021, https://doi.org/10.32604/cmc.2022.020938.

[5] M. A. Haq, "SMOTEDNN: A Novel Model for Air Pollution Forecasting and AQI Classification," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1403–1425, 2021, https://doi.org/10.32604/cmc.2022.021968.

[6] M. A. Haq, "CDLSTM: A Novel Model for Climate Change Forecasting," *Computers, Materials & Continua*, vol. 71, no. 2, pp. 2363–2381, 2021, https://doi.org/10.32604/cmc.2022.023059.

[7] M. A. Haq, A. K. Jilani, and P. Prabu, "Deep Learning Based Modeling of Groundwater Storage Change," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 4599–4617, 2021, https://doi.org/10.32604/cmc.2022.020495.

[8] M. A. Haq *et al.*, "Analysis of environmental factors using AI and ML methods," *Scientific Reports*, vol. 12, no. 1, Aug. 2022, Art. no. 13267, https://doi.org/10.1038/s41598-022-16665-7.

[9] H. Samadi and M. A. Kollathodi, "A comprehensive comparison and analysis of machine learning algorithms including evaluation optimized for geographic location prediction based on Twitter tweets datasets," *Cogent Engineering*, vol. 10, no. 1, Dec. 2023, Art. no. 2232602, https://doi.org/10.1080/23311916.2023.2232602.

[10] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence, "Improving Traffic Prediction with Tweet Semantics," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, Aug. 2013, pp. 1387–1393.

[11] Y. Labidi, "Sociological analysis of victims of fraud & deception - A study of a sample of victims in Algerian society," *ARID International Journal of Social Sciences and Humanities*, vol. 6, no. 11, pp. 173–187, Jan. 2024, https://doi.org/10.36772/arid.aijssh.2024.6117.

[12] X. Li, "Analysis of Criminal Activities Exploiting Social Media: With Special Regards to Criminal Cases of Wechat Fraud in Chinese Jurisdiction," *Journal of Legal Studies*, vol. 26, no. 40, pp. 19–36, Dec. 2020, https://doi.org/10.2478/jles-2020-0009.

[13] M. Madhukar and S. Verma, "Hybrid Semantic Analysis of Tweets: A Case Study of Tweets on Girl-Child in India," *Engineering, Technology & Applied Science Research*, vol. 7, no. 5, pp. 2014–2016, Oct. 2017, https://doi.org/10.48084/etasr.1246.

[14] A. Howard, Devrishi, P. Culliton, and Y. Guo, "Natural Language Processing with Disaster Tweets." 2019, [Online]. Available: https://kaggle.com/competitions/nlp-getting-started.

[15] B. Ahmed, G. Ali, A. Hussain, A. Baseer, and J. Ahmed, "Analysis of Text Feature Extractors using Deep Learning on Fake News," *Engineering, Technology & Applied Science Research*, vol. 11, no. 2, pp. 7001–7005, Apr. 2021, https://doi.org/10.48084/etasr.4069.

[16] A. Bathula, S. Muhuri, S. kr. Gupta, and S. Merugu, "Secure certificate sharing based on Blockchain framework for online education," *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16479–16500, May 2023, https://doi.org/10.1007/s11042-022-14126-x.

[17] A. Bathula, S. K. Gupta, S. Merugu, and S. S. Skandha, "Academic Projects on Certification Management Using Blockchain- A Review," in *2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC)*, Hyderabad, India, Sep. 2022, pp. 1–6, https://doi.org/10.1109/ICMACC54824.2022.10093679.

[18] S. Merugu, K. Jain, A. Mittal, and B. Raman, "Sub-scene Target Detection and Recognition Using Deep Learning Convolution Neural Networks," in *ICDSMLA 2019*, 2020, pp. 1082–1101, https://doi.org/10.1007/978-981-15-1420-3_119.

[19] M. Suresh, A. S. Shaik, B. Premalatha, V. A. Narayana, and G. Ghinea, "Intelligent & Smart Navigation System for Visually Impaired Friends," in *Advanced Computing*, 2023, pp. 374–383, https://doi.org/10.1007/978-3-031-35641-4_30.

[20] S. Merugu, M. C. S. Reddy, E. Goyal, and L. Piplani, "Text Message Classification Using Supervised Machine Learning Algorithms," in *ICCCE 2018*, Singapore, 2019, pp. 141–150, https://doi.org/10.1007/978-981-13-0212-1_15.

[21] A. Alabdulwahab, M. A. Haq, and M. Alshehri, "Cyberbullying Detection using Machine Learning and Deep Learning," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, pp. 424–432, Oct. 2023, https://doi.org/10.14569/IJACSA.2023.0141045.

[22] S. Kumar *et al.*, "Multilayer Neural Network Based Speech Emotion Recognition for Smart Assistance," *Computers, Materials & Continua*, vol. 74, no. 1, pp. 1523–1540, 2022, https://doi.org/10.32604/cmc.2023.028631.