IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 January 2023, accepted 26 January 2023, date of publication 3 February 2023, date of current version 8 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242290

RESEARCH ARTICLE

A Machine Learning-Sentiment Analysis on Monkeypox Outbreak: An Extensive Dataset to Show the Polarity of Public Opinion From Twitter Tweets

STAPHORD BENGESI[1], TIMOTHY OLADUNNI[2], RUTH OLUSEGUN[1], AND HALIMA AUDU[1]

[1]Department of Computer Science, Bowie State University, Bowie, MD 20715, USA
[2]Department of Computer Science, Morgan State University, Baltimore, MD 21251, USA

Corresponding author: Staphord Bengesi (Sbengesi@bowiestate.edu)

This work was supported in part by the Department of Computer Science, Bowie State University.

ABSTRACT Research on sentiment analysis has proven to be very useful in public health, particularly in analyzing infectious diseases. As the world recovers from the onslaught of the COVID-19 pandemic, concerns are rising that another pandemic, known as monkeypox, might hit the world again. Monkeypox is an infectious disease reported in over 73 countries across the globe. This sudden outbreak has become a major concern for many individuals and health authorities. Different social media channels have presented discussions, views, opinions, and emotions about the monkeypox outbreak. Social media sentiments often result in panic, misinformation, and stigmatization of some minority groups. Therefore, accurate information, guidelines, and health protocols related to this virus are critical. We aim to analyze public sentiments on the recent monkeypox outbreak, with the purpose of helping decision-makers gain a better understanding of the public perceptions of the disease. We hope that government and health authorities will find the work useful in crafting health policies and mitigating strategies to control the spread of the disease, and guide against its misrepresentations. Our study was conducted in two stages. In the first stage, we collected over 500,000 multilingual tweets related to the monkeypox post on Twitter and then performed sentiment analysis on them using VADER and TextBlob, to annotate the extracted tweets into positive, negative, and neutral sentiments. The second stage of our study involved the design, development, and evaluation of 56 classification models. Stemming and lemmatization techniques were used for vocabulary normalization. Vectorization was based on CountVectorizer and TF-IDF methodologies. K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest, Logistic Regression, Multilayer Perceptron (MLP), Naïve Bayes, and XGBoost were deployed as learning algorithms. Performance evaluation was based on accuracy, F1 Score, Precision, and Recall. Our experimental results showed that the model developed using TextBlob annotation + Lemmatization + CountVectorizer + SVM yielded the highest accuracy of about 0.9348.

INDEX TERMS Count vectorizer, machine learning algorithm, monkeypox, sentiment analysis, twitter, TF-IDF, TextBlob, Vader.

I. INTRODUCTION

Monkeypox is a viral disease caused by the monkeypox virus (MPXV), belonging to the same family of viruses that causes smallpox, known as the variola virus [1]. The World Health Organization (WHO) declared the spread of Monkeypox a global health emergency [2] due to its sporadic outbreak. The department of health and human services secretary of the United States, Xavier Becerra declared this virus a public health emergency on August 4, 2022 [3], because of the increased number of cases reported in the US.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

Monkeypox was first discovered in 1958 when the colonies of monkeys in a research institute in Denmark developed a pox-like disease. The first case in a human was confirmed in 1970 in the Republic of Congo [4]. Recently, cases have been reported from over 73 countries, and the record shows that the total number of cases reported worldwide as of September 23, 2022, was 65,415, with 24,846 cases from the United States of America [5]. The most trusted diagnosis for the virus is the polymerase chain reaction (PCR) test, and the available solution remains the development and administration of vaccines.

Studies have shown that Twitter can be an excellent data source for analyzing events worldwide, including health-related issues [6]. For example, since the eruption of COVID-19, social media platforms such as Facebook, Instagram, Pinterest, and Twitter have been the most active means of expressing opinions and sharing information among users [6]. Analyzing content posts is a way to understand the perception of human thought and emotions, as well as reveal the current mood and disposition of the broader human population.

Society's reliance on social media for information is enormous, unlike conventional news sources. The volume of data accessed daily led to the adoption of natural language processing (NLP) for text analytics [7]. This information may include social trends, governmental policies, public health, and other related matters. Companies also use social media to promote their products and services due to its low cost, easy access, and connectivity within the social media network. Therefore, social media platforms become a repository for information sources, reviews, and open communication where users' experiences are shared.

Understanding public's perception on infectious diseases is critical for the government and policymakers in formulating policies and mitigation strategies to control its spread. This study aims to identify people's sentiments expressed on Twitter about monkeypox disease. Our study applied natural language processing techniques to the datasets to make them suitable for our experiment. We annotated the preprocessed data using VADER and TextBlob and then vectorized them using CountVectorizer and TF-IDF. We adopted different machine learning algorithms to classify the sentiment into positive, negative, and neutral. The best-performing model was identified and optimized.

The rest of the research work is organized as follows: Section II presents a literature review of related work, Section III discusses the research methodology, Section IV presents the experimental results, Section V discusses our contributions, section VI concludes the research work, and section VII presents the future work.

II. LITERATURE REVIEW

With the large number of users on social media platforms, public opinions have become imperative to consider as a tool in decision-making. They provide insight into how people react to a particular topic. There have been several studies on infectious disease sentiment analysis such as the study performed by Neha et al [8]. Their study revealed the sentiment analysis of people during the coronavirus pandemic using deep learning algorithms: Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). These algorithms were used to develop model to predict the impact of the pandemic on the general populace. Another study similar to that of Neha et al. [8] was the study conducted by Chakraborty et al [9] which proposed a model based on deep learning classifiers and Gaussian functions to classify the sentiments of the public from tweets related to COVID-19 from the beginning of the virus outbreak through May 2020. The result of their study emphasized the need for a monitoring mechanism to prevent the spread of negative information about the virus.

Shahi et al. [10] also conducted a sentiment analysis of COVID-19 prediction. In their study, they performed analysis using tweets available in the Nepali language employing two widely used text representation methods: TF-IDF and FastText to capture discriminating features within the dataset. They developed models that implemented nine machine-learning algorithms to extract hybrid features from their dataset. The study in [11] and [12] is an extension of their study, where a multichannel CNN was used to perform sentimental analysis based on hybrid features. Their result showed that the proposed model provided excellent performance when compared with the state-of-the-art methods. However, the study is limited in scope because it only considered tweets posted only in the Nepali language.

A further study by Chinnasamy et al. [13] also performed a sentiment analysis to classify people's opinions and reactions to the COVID-19 Vaccine. They proposed a model that classified their dataset into positive, negative, and neutral sentiments. In their study, raw tweets were stored and processed using NLP, and then deployed a supervised KNN algorithm as the learning algorithm. The result of their experiment showed that people had a more positive perspective of Pfizer than the rest of the other covid -19 vaccines. A further review by the authors in [14] examined tweet sentiments on COVID-19 vaccination hesitancy. Their results indicated that COVID-19 vaccine hesitancy has steadily declined over time.

Considering other existing work on infectious diseases outbreak, we reviewed a study conducted by Chung et al. [15]. The study presented a sentiment and emotional analysis of tweets extracted from the social media platform on the Ebola disease outbreak. In the study, the authors proposed a model known as eMood. The proposed model used a comprehensive lexicon to identify emotion categories using a linear regression technique. The result of their study indicated that there is a relationship between user thoughts and emotions.

The recent outbreak of monkeypox disease has attracted research in this domain. Thakur et al. [16] created the first open-source monkeypox dataset following its outbreak.

In their study, more than 255,000 English-language tweets related to the 2022 monkeypox outbreak were collected. Their dataset was extracted by searching for the keyword "monkeypox" using RapidMiner software. This study only focused on the dataset features' development and classifications. However, no analysis was carried out on the developed datasets.

As a way to distinguish between monkeypox and other pox-like diseases, Sitaula et al. [17] performed an intensive study on the detection of the monkeypox disease using deep learning and computer vision techniques such as the visual geometry group (VGG) and the ResNet. Their study provides 13 Pre-trained Deep Learning (DL) models for monkeypox detection. The models were compared and evaluated; the performance of their model showed it is a reliable method for monkeypox virus detection.

Monkeypox is often mistaken for warts- benign bumps found on the skin, which could result in a wrong diagnosis and treatment. Tackling this misrepresentation, Alakus et al [18] performed an analysis on the DNA sequences of HPV-causing warts and MPV-causing monkeypox disease. The classification of the sequences was conducted using a deep learning algorithm. An average accuracy of 96.08 percent was achieved in their study. The study showed that the two diseases can be classified using their DNA sequences, which can help prevent wrong diagnoses and treatments.

Another study was carried out by the author [19] on monkeypox analysis. The study hypothesized that information sharing, and data seeking can be obtained through Google trend and Reddit platforms. However, the result of the experiment suggested that there was no significant discussion related to monkeypox on both Google trend and Reddit as compared to the information available on other social media platforms.

Jahanbin et al. [20] also did a study on the prediction of the monkeypox outbreak using data collected from Twitter and web news mining. Their study used the Fuzzy Algorithm for Monitoring, Extraction, and Classification (FAMEC) methodology to predict the outbreak of monkeypox disease. The dataset was cleaned, classified, and evaluated based on the developed algorithm. Their study showed the FAMEC model has the potential and capability to track and monitor zoonotic diseases like monkeypox, but the data collected for analysis was limited to posts available at the beginning of the outbreak. Similarly, Mohbey et al. [21] provided an analysis of individuals' thoughts about monkeypox disease. In their study, they presented a hybrid deep learning technique based on CNN and Long short-term memory (LSTM) which generated three possible sentiments - positive, negative, and neutral from people's tweets on Twitter. They developed a CNN-LSTM model to determine their model's accuracy, but the scope of their analysis was limited to tweets in the English language which does not allow a larger perspective on the subject matter.

People leverage social media platforms to express their feelings and sentiments about public health situations. Therefore, it is critical to analyze public sentiment and its dynamics to reveal insights into current issues. Based on the previous research on monkeypox sentiment analysis, there seem to be gaps that needed to be discussed.

1. Previous studies on monkeypox analysis showed that most of the data collected for use were limited to the first detected outbreak cases of the virus. We believe that the monkeypox situation may have changed regarding the number of cases and public views posted on social media. Hence performing an analysis of the most recent cases is vital.
2. Also, most prior research on this topic was based on downloaded datasets from other public sources, which does not specify how their datasets were preprocessed. The data preprocessing step in data analysis is vital for achieving the most effective classification model. Hence, addressing each analysis step is critical and needs to be discussed.
3. Most of the previous research focused only on English-language tweets. We argue that extending the scope of our analysis across several other languages would help consider the larger community's opinions in making better decisions.

Given the gaps identified, this research aims to perform a sentiment analysis on the monkeypox outbreak with an up-to-date dataset collected from tweets posted on the Twitter social media platform. To generate insight into how public opinions can help policymakers and health authorities take proactive steps and decisions to control the outbreak of this disease, as well as enlighten the general populace on taking preventive measures amidst the crisis. Our study collected and preprocessed a multilingual dataset of 103 languages for a detailed analysis. Table 1 shows the distribution of our dataset based on the top 4 languages. We designed, developed, and evaluated 56 models. Modeling was based on a combination of Natural Language Processing and different learning algorithms. The best-performing classification model was identified.

III. METHODOLOGY

A. OVERVIEW

Our experimental framework shown in figure 1 began with the collection of data, translation, and preprocessing. During preprocessing, we removed retweets, punctuation marks, hashtags, user tags, stopwords, numbers, repeated words, and the emojis were converted to text. Stemming and lemmatization were applied to normalize the preprocessed dataset. VADER and TextBlob techniques were applied to compute sentiment scores for the dataset. Vectorization of tokens was achieved using CountVectorizer and TF-IDF techniques. The final step was to construct classification models using machine learning methods such as Random Forest, Naïve Bayes, K-Nearest Neighbor (KNN), Multilayer

TABLE 1. Language count.

Language	Count
English (en)	91,998
Portuguese (pt)	3,389
Undetermined (und)	2,536
Spanish (es)	2,219
French (fr)	2,181

Perceptron (MLP), support vector machine (SVM), and logistic regression.

B. DATA COLLECTION

This study extracted tweets using the Twitter API and Tweepy library. Tweepy is a python package for accessing Twitter API that allows developers to access Twitter content, such as tweets, retweets, and timestamps. A python script was used to search for all the tweets related to the keyword ‘‘**#monkeypox**’’. All text that met our criteria was extracted and stored in a comma-separated values (CSV) file. We considered a total of five features for our analysis: text, timestamps, author, source, and language. Google Translate API was used to translate all non-English tweets to English.

We collected over 500,000 tweets between July 2022 and September 2022; however, after preprocessing, we were left with 107,000 unique tweets. Table 1 shows our dataset's top five language counts; we had about 103 languages in total.

C. DATA PREPROCESSING

Data preprocessing is an integral part of natural language processing (NLP) that helps to reconstruct raw text into a meaningful format. A variety of tools and mechanisms were used in our study for preprocessing. Since our dataset is multilingual, it is crucial that all the data are in the same language. To accomplish this, we used the Google Translate API to translate all non-English tweets into English. All retweets, punctuation, hashtags, stop words, tokenization, stop words, repeated words, stemming, and lemmatization are then removed. We will discuss each task in the following manner.

1) RETWEET (RT) AND USER TAGS REMOVAL

Retweeting is resharing someone's tweets on Twitter. By sharing tweets, duplicates are created that can adversely affect model training and accuracy, so it was necessary to remove retweets. An ‘‘RT’’ indicates a retweet, while an ‘‘@Someone’’ indicates a user tag. They were omitted too.

2) EMOJI AND TEXT CONVERSION

People express their opinions and emotions using small digital images and icons called emojis. We converted these images into their corresponding textual format to improve our model training. Also, all the dataset texts were converted into lowercase to avoid double recognition of the same word.

3) HASHTAG, NUMERAL, AND PUNCTUATION REMOVAL

A hashtag is a common term for searching and retaining related content on social media [20]. Usually, the hash sign (#) precedes the keyword, and it is a powerful tool in social media but unnecessary for learning models; hence it was removed from the dataset. Numerical, repeated words and punctuation were removed using regular expressions (RegExp). This reduced memory consumption and accelerated the learning process.

4) STOPWORD REMOVAL

A stopword refers to words that don't add much meaningful information to a sentence, such as ‘to,’ ‘me,’ ‘my,’ ‘ours,’ etc. For this reason, they were removed using a Python package library named stopword, to avoid noise in our dataset.

5) TOKENIZATION

Tokens are created by splitting text into smaller chunks of individual words using the natural language toolkit. Sentiment analysis requires tokenization to simplify feature extraction.

6) LEMMATIZATION AND STEMMING

A tweet can have one word written differently with the same meaning since there is no standard format. Using lemmatization and stemming, we avoid such a scenario. Lemmatization converts all words into their dictionary-based form, commonly known as a lemma while stemming involves discarding the word's last few characters to get the word's meaningful base.

D. DATASET EXPLORATION

For familiarity and insight, we explored the dataset before training. High-frequency words were extracted and visualized. For text exploration, the next section discusses word frequency and word clouds.

1) WORD FREQUENCY

Following preprocessing, word frequency was used to explore the dataset. By analyzing word frequency, we were able to identify the most commonly used words. In Table 2, the ten most common keywords are listed. Monkeypox was the most common word because everyone who posted about this outbreak included monkeypox in their sentence.

The second phrase with the highest frequency was **not.** In our study, we did not consider ‘not’ as a stopword because removing it could cause the sentence to lose its meaning. Twitter users dispel the myth that monkeypox is spread by air contamination, such as COVID-19, while others claim that monkeypox is a sexually transmitted disease. These different perspectives are the reason why it is not considered. **Vaccine** emerged as the third most frequently used word, probably

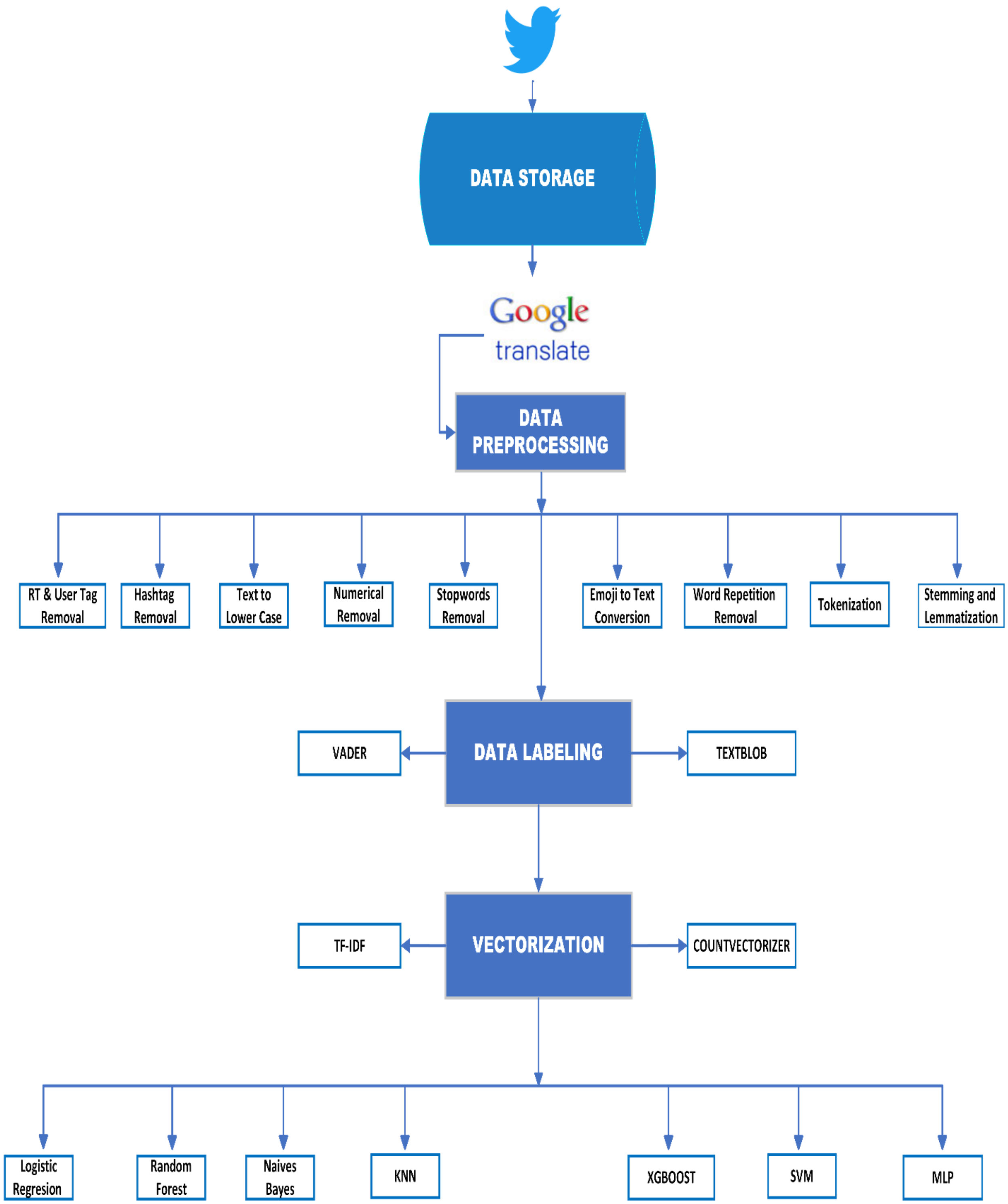

FIGURE 1. Experimental framework. The figure elucidates a step-by-step methodology for our experiment starting from data collection to pre-processing, labeling, vectorization, and classification algorithms applied with their respective components.

because Twitter users were discussing whether to accept or hesitate about monkeypox vaccination. Another word that was frequently used was **case**, possibly because many people were discussing new cases of monkeypox at the time.

Furthermore, it was found that many people were concerned about their health, so it may be possible that this is why **health** was listed as the fifth most frequent word in the search results. Some discussions also compared the covid-19

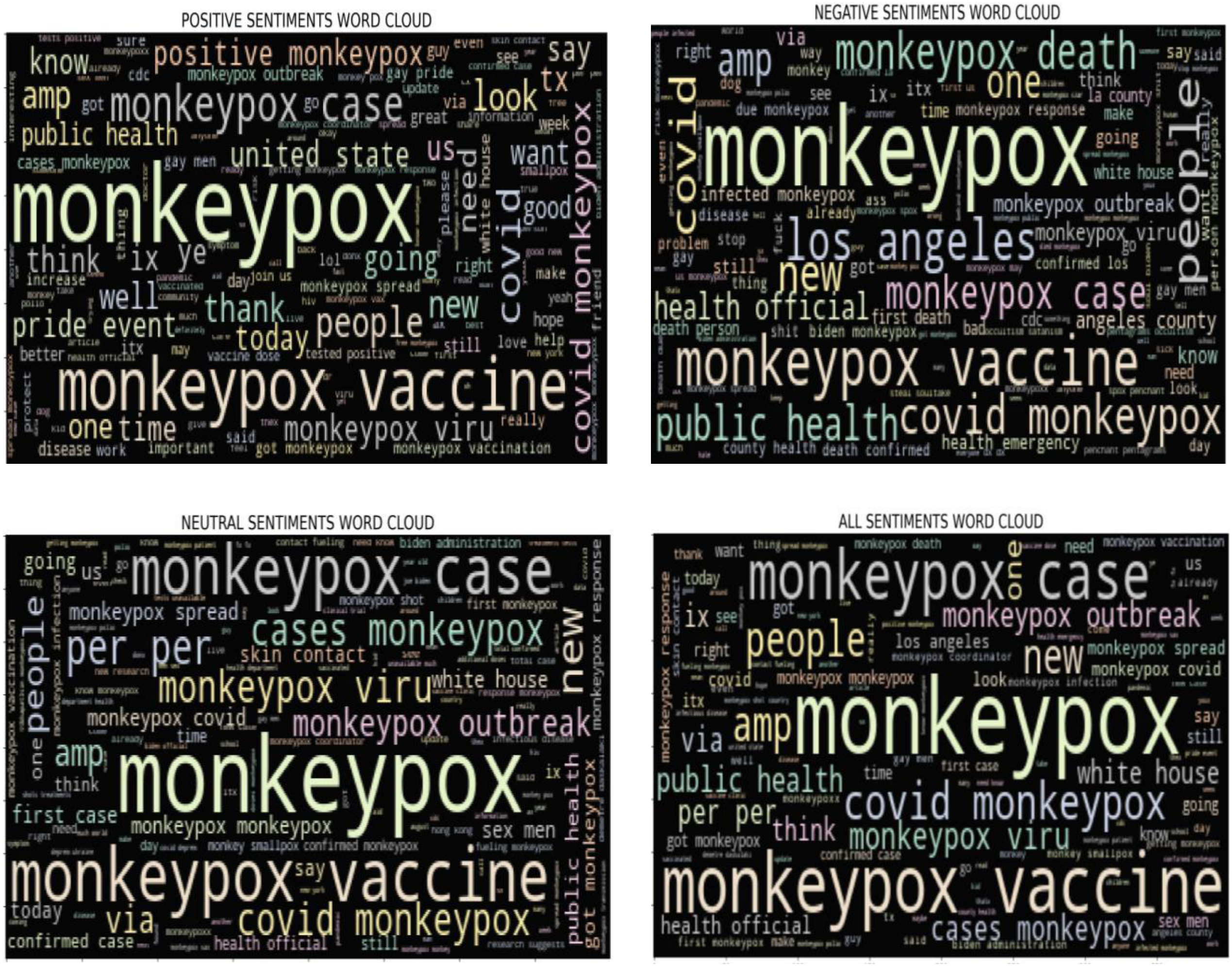

FIGURE 2. Word Cloud. Display of the most prevalent keywords based on positive, negative, neutral, and the entire dataset sentiment respectively.

TABLE 2. Word frequency.

Keyword	Frequency
Monkeypox	79,489
Not	10,750
Vaccine	8,741
cases	8,589
Health	6,745
Covid	6,400
new	6,086
people	4,528
first	4,458
Sex	3,898

pandemic with the most recent outbreak, which caused the **covid** word to appear as number six in the rank. **News** ranked seventh, possibly because of continued coverage of the outbreak. It was followed in the ranking by the keyword **people**; presumably, most posts discussed the number of people affected by the disease.

Several tweets termed this disease a sexually transmitted disease, possibly resulting in the **sex** keyword being ranked tenth in frequency.

2) WORD CLOUD

We attempted word cloud to further investigate the prior discussed exploration technique. Word cloud is a visualization technique of observing the most common word available in our dataset. We generated four visualizations: one for the entire dataset and three for the sentiment polarities (positive, negative, and neutral). It was observed that all the words which showed up in the keyword frequency popped up in the visualization shown by figure 2. Again, monkeypox, case, new, covid, people, and vaccine were among the most visible words in all visualization

E. DATA LABELING

Data labeling refers to adding a label(s) to raw data. Usually, it serves as a bridge between raw data and results obtained from a machine learning model. It helps a machine learning model identify the specific class of an object in a dataset. Due to the volume of data collected in our study, we applied an existing tool for annotation using VADER and TextBlob.

In VADER, we used the output compound score to determine the label, while we used the polarity score to determine the label in TextBlob. Positive, Negative, and Neutral were the three possible labels. Algorithm 1 shows detailed steps of our annotation.

Algorithm 1 Labeling Steps

```
Input: Unannotated Tweet Dataframe: $df_u$
Output: Annotated Tweet Dataframe: $df_a$
For each row in $df_u$:
  if row ['language'] != "English Language":
    row['text'] = Google translate(row['text'])
  Annotate1: TextBlob (row['text']):
    if $0 < polarity\ score \leq 1$:
      Annotated as Positive
    else if $-1 < polarity\ score < 0$:
      Annotated as Negative
    else:
      Annotated as Neutral
  Annotate2: VADER (row['text']):
    if $-0.05 < compound\ score < 0.05$:
      Annotated as Neutral
    else if $compound\ score \leq 0.05$
      Annotated as Negative
    else:
        Annotated as Positive
end:
return annotated Dataframe ['Annotate1', 'Annotate2']
```

1) VADER

VADER is an acronym for the Valence Aware Dictionary and Sentiment Reasoner. It is an open-source platform based on lexicon and rule-based, invented in 2014 [6]. It was used to analyze text based on three polarities (negative, neutral, and positive). VADER labels text based on the polarity score calculated by normalizing the sum of the positive, negative, and neutral scores of sentences/words. The normalization falls within the range of -1 (most extreme negative) and $+1$ (most extreme positive). Equation 1 summarizes the labeling process in Vader according to Hutto et al [22], [23].

$$\text{Label} = \begin{cases} Positive & if\ score \geq 0.05 \\ Negative & if\ score \leq -0.05 \\ Neutral & if\ -0.05 < score < 0.05 \end{cases} \quad (1)$$

As an outcome of the VADER application, out of 107,336 tweets in our study, about 27.3% were positive, 25.7% were negative, and 47% were neutral tweets. Table 3 shows the frequency of each polarity.

TABLE 3. VADER annotation score.

Polarity	Count	Percentage
Positive	29,295	27.3%
Negative	27,628	25.7%
Neutral	50, 473	47%

TABLE 4. VADER result.

TEXT	LABEL
that monkeypox idea failed terribly	Negative
so on sept new york is under concurrent states of emergency for covid monkeypox and polio	Negative
states reporting the highest number of cases include california texas florida georgia illinois and new york	Positive
covid monkeypox we communicate to you the week of monitoring of elderly hasl set of the indicators	Neutral
he looks like he is in charge of deployment and distribution of monkeypox	Positive

TABLE 5. TextBlob annotation score.

Polarity	Count	Percentage
Positive	39,594	36.9%
Negative	14,788	13.8%
Neutral	52, 954	49.3%

Table 4 demonstrates the output samples from VADER technique implementation.

2) TEXTBLOB

TextBlob is another sentiment analyzer based on lexicon (Rule-based sentiment analyzer) [24] that we adopted in our study. We developed a python loop on all rows in our datasets, and the polarity and subjectivity were returned through the textblob() call. A polarity score is a floating number between 0 and 1, while its subjectivity lies between 0 and 1 [25]. In this study, we are interested in the polarity score, converted to a label, as shown in Equation 2.

$$\text{Label} = \begin{cases} Positive & if\ 0 < score \leq 1 \\ Negative & if\ -1 \leq score < 0 \\ Neutral & if\ score == 0 \end{cases} \quad (2)$$

As Table 5 depicts, 36.9% of the dataset was labeled positive, 13.8% as negative, and 49.3% as neural.

It was interesting to see a considerable variation between TextBlob and Vader. Positive polarity for TextBlob was almost 10% more than positive polarity in VADER. There had a 12% disparity between VADER and TextBlob for negative

TABLE 6. TextBlob annotation vs VADER annotation disparity.

S/NO	Text	VADER Label	TEXTBLOB Label
1.	monkeypox is spreading in lagos please be careful how you make physical contact with people	Positive	Negative
2.	is it me or did the media just suddenly and completely stop talking about monkeypox?	Negative	Positive
3.	bitch got bumps on her face that look like monkeypox	Negative	Neutral
4.	overdue for an mpx (monkeypox) update much progress has been made in us in the last few weeks with impressive gains	Positive	Positive
5.	health officials continue offering vaccines against the monkeypox virus recent statistics show over confirmed	Neutral	Neutral
6.	mississippi expands eligibility for monkeypox vaccine the vaccine a two-dose series is available at nine	Positive	Positive
7.	vaccinate everyone who has not received the initial vaccine and give the new booster to everyone we can do a circle	Neutral	Positive

polarity scores. The neutral polarity had slight differences of about 2%

Table 6 shows some disparities between the two annotation methods in our study. Our findings agree with previous works on the disparities between the two annotation techniques. Studies have shown that unlike TextBlob, VADER is more focused content [26], discerning polarity, and sentiments for emojis in social media [27].

In summary, by looking at the result of both techniques, it was evident that most people either had a positive or neutral opinion regarding the monkeypox outbreak; suggesting that people understood the outbreak.

F. WORD EMBEDDING AND VECTORIZATION

Word embedding or text vectorization is a technique in natural language processing that maps the words or phrases from sentences to the corresponding vector of real numbers used to find word prediction, similarities, and semantics [28]. Performing word embedding makes it easier to train and extract features in machine learning. Below are different methods of vectorization applied in our study:

1) COUNTERVECTORIZER

CounterVectorizer is a vectorization method that converts a data (text) into a vector of words and its corresponding frequencies [29]. This technique creates a dictionary of all possible and available words in the dataset where each word in the sentence is assigned a specific random number.

2) TF-IDF

For a more balanced study, another vectorization method called TF-IDF was used. This technique consists of Term frequency (TF) and Inverse document frequency (IDF). TF focuses on the raw word count in the document, while IDF focuses on how the frequency of a word is measured. Equation 3 depicts the TF formulation.

$$\text{TF(t, d)} = \frac{\text{Frequency of term (t) in the document(d)}}{\text{Total word in the document (d)}} \tag{3}$$

IDF's purpose is to calculate the informativeness of the word in a document. We need IDF because it helps minimize the weight of frequent terms and makes infrequent terms have a high impact. IDF can be computed using Equation 4.

$$IDF\ (t) = \log_2 (\frac{Total\ Documents(N)}{1 + Total\ Documents\ with\ term(df\,(t))}) \tag{4}$$

TF-IDF expression on Equation 5 is the aftermath of combining Equations 3 & 4 [30]

$$TF - IDF = tf\,.idf\ (t, d, N) = tf\ (t, f)\ .idf\,(t, N) \tag{5}$$

G. LEARNING ALGORITHMS

In this study, we designed, developed and evaluated various models using different machine learning algorithms. 80% of the datasets was used for training and 20% for validation. We measured the performance of each model based on accuracy, precision, recall, and F1 score. Default hyper-parameter values in sklearn were used for each of learning algorithms. Below is the discussion of the implemented algorithms.

1) LOGISTIC REGRESSION

Logistic regression can be defined as the supervised learning algorithm which predicts the probability of an event occurrence. Its probability is merely based on the selected independent variable vs the dependent variable. This kind of modeling outputs discrete outcome for the given input variable. Logistic regression is mathematically represented in equation 6.

$$P\,(Y = 1) = \frac{1}{1 + e^{-(b0+\sum b_i X_i)}} \tag{6}$$

In equation 6, Y is the discrete dependent variable (i.e., 0, 1, 2 . . .), and X is an independent variable with subscripts i.

The cost function is defined in equation 7.

$$Cost\ (h_\theta\ (x)\,, y) = \begin{cases} -\log(h_\theta\ (x)) & if\ \ y = 1 \\ -\log\,(1 - h_\theta\ (x)) & if\ \ y = 0 \end{cases} \tag{7}$$

Logistic regression uses binary cross entropy (see Equation 8), also known as log loss, as the loss function [31]

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i .\log(y_i) + (1-y_i)\,.log\,(1-y_i) \tag{8}$$

where:

N-total number of categories

Y-dependent variable.

2) Naïve BAYES

The Naïve Bayes was another algorithm we adopted for classification. It is a probabilistic classifier that uses conditional probability to determine the class likelihood of its input [32]. Equation 9 defines the calculation of probabilities for each class involved.

$$\frac{conditional\ Probality * Prior\ Probabilty}{Evidence} \tag{9}$$

Thomas Bayes (1701-1761) terms the output as the posterior probability.

Mathematically,

$$P\left({}^{y}\!/_{X}\right) = \frac{P\left({}^{X}\!/_{y}\right) P(y)}{P(X)} \tag{10}$$

where:

P(y): Prior Probability

$P(X/y)$: Likelihood probability

$P(y/X)$: Posterior Probability

$P(X)$: Marginal probability (Evidence).

Looking at Equation 10, it means that event y will occur given that event X occurred. In this case, y happens as the hypothesis and X as evidence. y event depends on event X occurrence. X comprises all independent features i.e $(x_1, x_2 \ldots\ldots\ldots x_n)$. There are multiple types of naïve Bayes; however, in our study, we applied multinomial naïve Bayes, a model which focuses on document processing classification problems [33]. This model assumes that each feature consists of multinomial distribution whereby each word count toward class prediction. Equation 11 defines the multinomial naïve bayes.

$$P\left({}^{c}\!/_{d}\right) = \frac{P(c)\prod_{i=1}^{n} P(w_i/c)^{f_i}}{P(d)} \tag{11}$$

where f is the frequency of a word (w) in document (d), P(c) is the prior probability that class (c) belongs to the document. P $(\mathbf{w_i}/\mathbf{c})$ is the conditional probability that the word occurs in the document belongs to the class.

3) SUPPORT VECTOR MACHINE

SVM is a robust model that sets boundaries between classes [34], sorting data into one of the available categories. SVM needs decision boundaries (separator lines) between classes called a hyperplane. There exist three hyperplanes, namely positive, negative, and optimal hyperplanes. Mathematically these hyperplanes are presented by Equation12-14:

$$\vec{w}.\vec{x} + b = 1 \quad \text{for Positive hyperplane} \tag{12}$$

$$\vec{w}.\vec{x} + \mathrm{b} = -1 \quad \text{for Negative hyperplane} \tag{13}$$

$$\vec{w}.\vec{x} + \mathrm{b} = 0 \quad \text{for optimal hyperplane} \tag{14}$$

w is the width of the margin, b is the bias, and x is the features.

The margin width needs to be maximized for the model to have the best optimal hyperplane. In cases where the problem is non-linear, the algorithm is good enough to solve it using the kernel, which mounts into higher dimensions, making them separable. Some kernels exist in SVM, such as polynomial, Gaussian, and Gaussian radial basis functions (RBF).

4) RANDOM FOREST

Random Forest is another model we utilized. It is an ensemble machine-learning classification algorithm. This algorithm develops numerous decision trees used for classification, which implies that class category is selected by most of the trees. This approach involves randomization and aggregation of tree prediction [35] into the final output.

Random forest requires at least three hyperparameters to be in place: node size, number of trees, and number of features sampled. It applies bootstrap aggregation, also known as the bagging ensemble technique, which creates a different subset of training adopted from sample training data. The result depends on the rate of preference [36].

a: K-NEAREST NEIGHBOR (KNN)

KNN is one of the most popular machine learning algorithms used not only as a classification algorithm but also in information retrieval, pattern recognition, and regression problems [37]. It is a non-parametric algorithm capable of generating a consistent result in a data sample. The algorithm first turns data into a vector with features extracted to help find similarities between two data points using a distance measurement.

Our study used a supervised KNN classifier to classify the polarized data. We classified data based on the polarity score. A tweet with a polarity score greater than zero (Tweet Polarity > 0) is classified as a positive. Polarity score less than zero (Tweet Polarity < 0) is classified as a negative. If the polarity score is equal to zero (Tweet Polarity $==$ 0), then we classified those as neutral [7]. The KNN algorithm used feature similarities to assign a data point based on how close it is to its neighbor. Algorithm 2 describes KNN in detail:

To calculate the distance between each data point in the KNN algorithm, we used the Euclidean distance, which is as calculated in Equation 1

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{15}$$

Euclidean Distance Function.

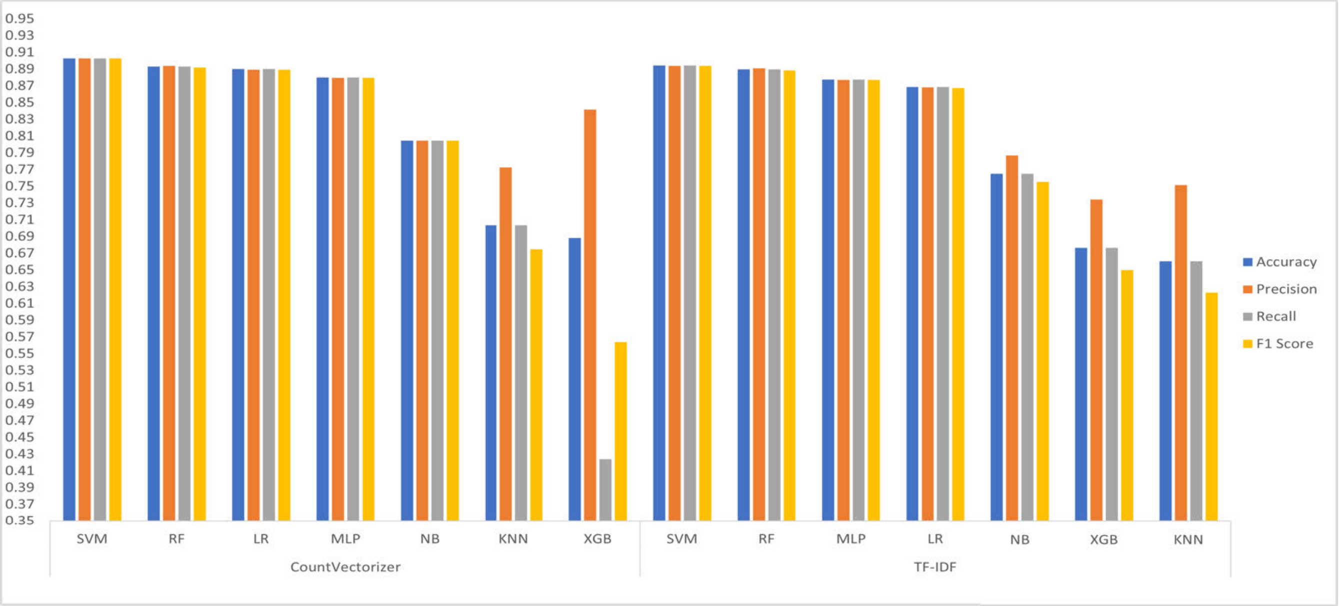

FIGURE 3. Machine Learning Algorithms Performances using Vader Labelling and Stemming tokenization with CountVectorizer or TF-IDF vectorizer. A chart illustrating the performance metrics for each machine learning algorithm in terms of accuracy, precision, recall, and F1 score.

```
Algorithm 2 KNN Algorithm
    Step 1: Load the preprocessed datase
    Step 2: Determine the parameter K
    Step 3: Calculate the distance between each data point
    Step 4: Sort data points according to the distance
                calculated
    Step 5: Select the top K row
    Step 6: Assign data points on the most frequent class
    Step 7: END
```

5) MLP

The Multilayer Perceptron model is one of the standard neural network models with a simple mathematical function used to learn complex features within a dataset. Generally categorized under the feedforward algorithm, where inputs and initial weights are combined in a weighted sum, subjected to the activation function [38].

In our MLP, we used gradient descent as the optimization function, i.e., for all iterations, a gradient mean-square error is computed until a specified convergence threshold is attained [39]. The mean squared error is calculated using the Equation 16:

$$\Delta_w(t) = -\varepsilon \frac{dE}{dw_{(t)}} + \propto \Delta_{w(t-1)} \tag{16}$$

6) XGBoost

eXtreme Gradient Boosting is a flexible gradient boosting decision tree available in machine-learning library. It provides cutting-edge results on many machine-learning problems [40]. XGBoost builds upon the concept of supervised machine learning, decision trees, ensemble learning, and Gradient boosting. It has an advantage over these machine learning methods due to its flexible nature and high speed, its ability to exploit parallel processing, support regularization, handle missing data, run cross-validation, and is essentially suitable for small and medium datasets. In our study, XGBoost was used from a scikit-learning package in Python. In our study, XGBoost was used along with a scikit-learning package in Python.

TABLE 7. Developed models.

S/NO	MODEL
1.	VADER+ Lemmatization+CountVectorizer
2.	VADER+ Lemmatization+TD-IDF
3.	VADER+Stemming+CountVectorizer
4.	VADER+ Stemming+TD-IDF
5.	TextBlob+ Lemmatization+CountVectorizer
6.	TextBlob + Lemmatization+TD-IDF
7.	TextBlob +Stemming+CountVectorizer
8.	TextBlob + Stemming+TD-IDF

IV. EXPERIMENT RESULTS AND DISCUSSION

This section presents the experimental steps used to evaluate the performance of the proposed models. In our study, we designed, developed, and evaluated 56 models based on the labeling, vectorization, and normalization methods. Table 7 shows the eight (8) ways we generated our models. We trained and evaluated each model on seven (7) machine learning algorithms, including Random Forest, Logistic regression, Support vector machine (SVM), multilayer

TABLE 8. Output of VADER labelling and Stemming tokenization with CountVectorizer or TF-IDF models.

Model		Accuracy	Precision	Recall	F1 Score
CountVectorizer	SVM	0.9029	0.902884	0.902879	0.902848
	Random Forest	0.893	0.894167	0.892957	0.891673
	Logistic Regression	0.89	0.88943	0.890022	0.88931
	MLP	0.8801	0.879807	0.880054	0.879785
	Naïve Bayes	0.8044	0.804465	0.804407	0.804425
	KNN	0.7033	0.772385	0.703326	0.674828
	XGBoost	0.6881	0.8416	0.42401	0.5639
TF-IDF	SVM	0.8943	0.894001	0.894261	0.894079
	Random Forest	0.8898	0.890843	0.889836	0.888511
	MLP	0.8776	0.877295	0.877632	0.877274
	Logistic Regression	0.8686	0.868222	0.868595	0.867412
	Naïve Bayes	0.7649	0.787029	0.764906	0.755195
	XGBoost	0.6766	0.734044	0.676588	0.649801
	KNN	0.6606	0.751419	0.660611	0.623142

TABLE 9. Output of vader Labelling and LEMMATIZATION tokenization with CountVectorizer or TF-IDF models.

Model		Accuracy	Precision	Recall	F1 Score
CountVectorizer	SVM	0.9295	0.92917	0.929523	0.929213
	MLP	0.9186	0.918336	0.918623	0.918436
	Logistic Regression	0.9125	0.912437	0.912521	0.911664
	Random Forest	0.902	0.903831	0.901994	0.900548
	Naïve Bayes	0.8266	0.827254	0.826626	0.826907
	KNN	0.6977	0.773116	0.697736	0.66761
	XGBoost	0.6824	0.750501	0.682411	0.654007
TF-IDF	SVM	0.9175	0.917074	0.917459	0.916771
	MLP	0.9054	0.905155	0.905394	0.904752
	Random Forest	0.891	0.894033	0.891047	0.889278
	Logistic Regression	0.8868	0.887996	0.886762	0.885045
	Naïve Bayes	0.7855	0.802457	0.785495	0.777564
	XGBoost	0.6774	0.748573	0.677427	0.648002
	KNN	0.6601	0.755215	0.660052	0.62135

perceptron (MLP), eXtreme Gradient Boosting (XGBoost), and Naïve Bayes. We recorded the results of each model based on their accuracy, precision, recall, and F1 score. Tables 8, 9, 10 and 11 show the results of our experiment tabular form. Figures 3, 4, 5 and 6 show the result in graphical form.

The results obtained are listed in the figure as follows:

- Table 8 shows the output for models developed from the combination of Stemming tokenization and VADER labeling with CountVectorizer or TF-IDF Vectorizer
- Table 9 shows the output for the models developed from combination of Lemmatization tokenization and VADER labeling with CountVectorizer or TF-IDF Vectorizer
- Table 10 shows the output of models developed from combination of Stemming tokenization and TextBlob labeling with CountVectorizer or TF-IDF Vectorizer
- Table 11 shows the output of models developed from the combination of Lemmatization tokenization and TextBlob labeling with CountVectorizer or TF-IDF Vectorizer
- Figures 3, 4 and 5 are the graphical representation of tables 8, 9, 10 and 11 respectively

From our experiments, we applied two normalization techniques, namely, Stemming and Lemmatization. We observed that lemmatization models showed better results than the stemmed models in all the cases. In the case of labeling, the TextBlob labeling model was better than the VADER model, and the CountVectorizer model outperformed the TF-IDF model. The model which applied SVM, lemmatization, CountVectorizer, and TextBlob annotation emerged as the best model, with an accuracy of about 0.9348. Random Forest, MLP, and logistic regression also performed well, with an accuracy ranging from 0.85 to 0.92. KNN and the

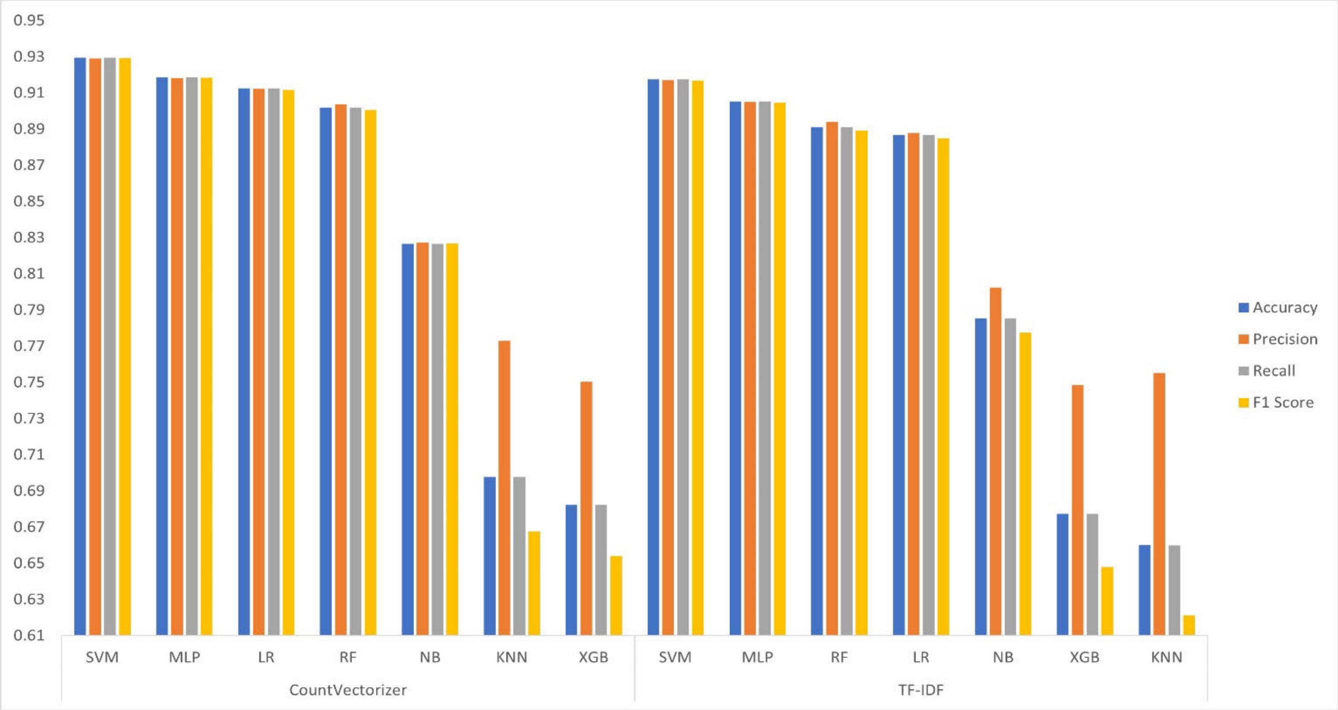

FIGURE 4. Machine Learning Algorithms Performances using Vader Labelling and Lemmatization with CountVectorizer or TF-IDF Vectorizer. A chart illustrating the performance metrics for each machine learning algorithm in terms of accuracy, precision, recall, and F1 score.

TABLE 10. Output of TextBlob Labelling and stemming tokenization with CountVectorizer or TF-IDF models.

Model		Accuracy	Precision	Recall	F1 Score
COUNTVECTORIZER	Random Forest	0.9062	0.907595	0.906186	0.903854
	SVM	0.905	0.904308	0.905021	0.904394
	Logistic Regression	0.8951	0.894331	0.895146	0.893569
	MLP	0.884	0.88345	0.883967	0.882966
	Naïve Bayes	0.8109	0.819083	0.810881	0.802484
	XGBoost	0.7346	0.779429	0.734628	0.707634
	KNN	0.7005	0.775376	0.700484	0.675092
TF-IDF	Random Forest	0.9005	0.902209	0.900456	0.897492
	SVM	0.8971	0.896378	0.897149	0.895903
	Logistic Regression	0.88	0.879512	0.880007	0.877217
	MLP	0.8799	0.880036	0.879914	0.879282
	Naïve Bayes	0.7795	0.80445	0.779486	0.738342
	XGBoost	0.733	0.779622	0.733045	0.705959
	KNN	0.6538	0.754721	0.65381	0.614214

XGBoost algorithms had the least performance with an accuracy that ranges between 0.6 to 0.

V. CONTRIBUTION

Previous studies have explored monkeypox sentiment analysis, our study highlights the following key contributions:

1) This study extracted over 500,000 monkeypox datasets from tweets between July 2022 and September 2022. Preprocessing and normalization of the collected data resulted in 107, 000 clean datasets. Preparing the datasets, we annotated and classified them into positive, negative, and neutral sentiments. These datasets are available to the public for research purposes1. According to [15], the authors generated over 255,000 monkeypox datasets from tweets available between May 2022 and June 2022. Another author also published a study in [17] using 61,379 datasets obtained

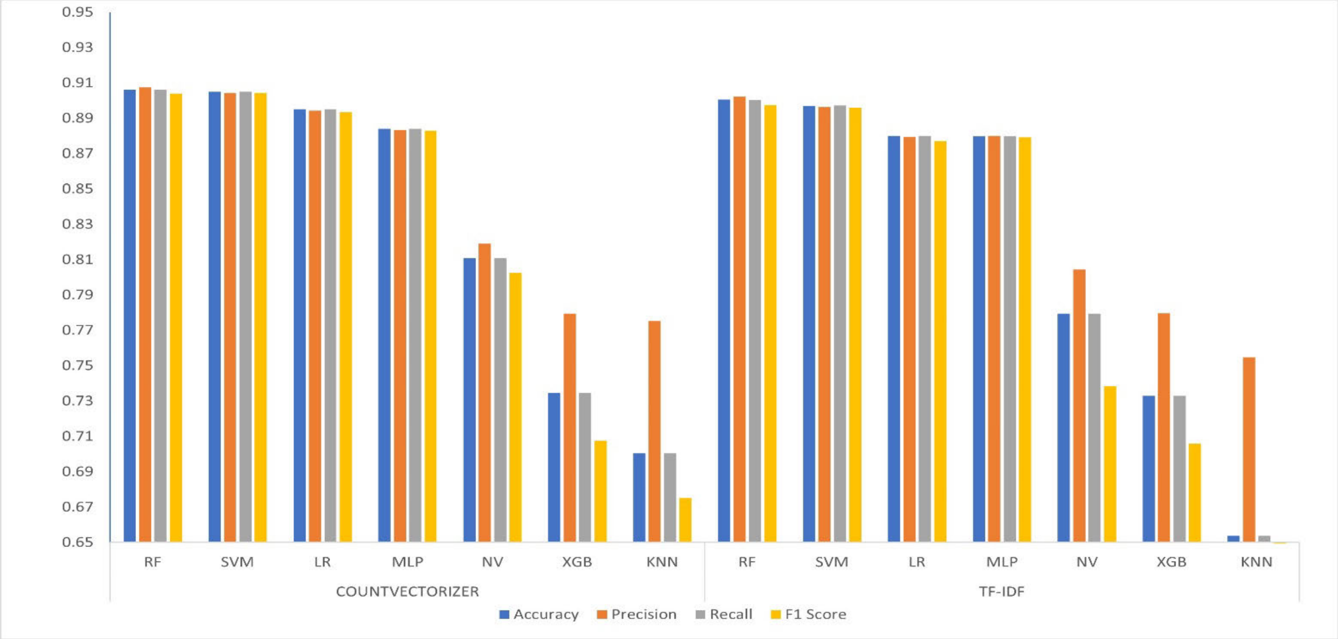

FIGURE 5. Machine Learning Algorithms Performances using TextBlob Labelling and Stemming tokenization with CountVectorizer or TF-IDF. A column chart illustrating the performance metrics for each machine learning algorithm in terms of accuracy, precision, recall, and F1 score.

TABLE 11. Output of TextBlob Labelling and lemmatization tokenization with CountVectorizer or TF-IDF models.

Model		Accuracy	Precision	Recall	F1 Score
COUNTVECTORIZER	SVM	0.9348	0.934379	0.934833	0.934261
	Random Forest	0.9251	0.926403	0.925098	0.923212
	Logistic Regression	0.9229	0.922801	0.922862	0.921573
	MLP	0.9137	0.912863	0.913685	0.913073
	Naïve Bayes	0.8362	0.845236	0.836175	0.829757
	XGBoost	0.7374	0.791622	0.737377	0.708219
	KNN	0.7003	0.782839	0.700252	0.67339
TF-IDF	SVM	0.9255	0.925349	0.92547	0.924255
	Random Forest	0.9184	0.920078	0.918437	0.916189
	MLP	0.9074	0.90767	0.907444	0.906647
	Logistic Regression	0.9026	0.903637	0.902553	0.899645
	Naïve Bayes	0.7954	0.8183	0.795416	0.756638
	KNN	0.6519	0.756565	0.651901	0.611043

from a public source. Data collected were also between May 2022 and June 2022. However, this study contains a much larger number of recent datasets than previous studies, since there is a possibility that the current monkeypox situation might have changed. Therefore, using the most recent datasets, we argue that our experimental results are more reliable and reproducible than previous studies.

2) The research presented in our study was based on a multilingual dataset, which makes it unique compared to other studies. Over 103 languages of tweets were extracted and analyzed. Table 1 shows the top five languages processed in the study. Previous research in related fields has not considered several languages. As a result, we were the first to perform sentiment analysis on monkeypox disease with multilingual datasets. We believe that our study provides a more universal and broader approach in understanding the sentiments of the public on the monkeypox virus.

3) We explored the tweeter dataset using Word Frequency and Word Cloud techniques. Our Word Frequency analysis showed that Monkeypox, Not, Vaccine, Cases, Health, Covid,

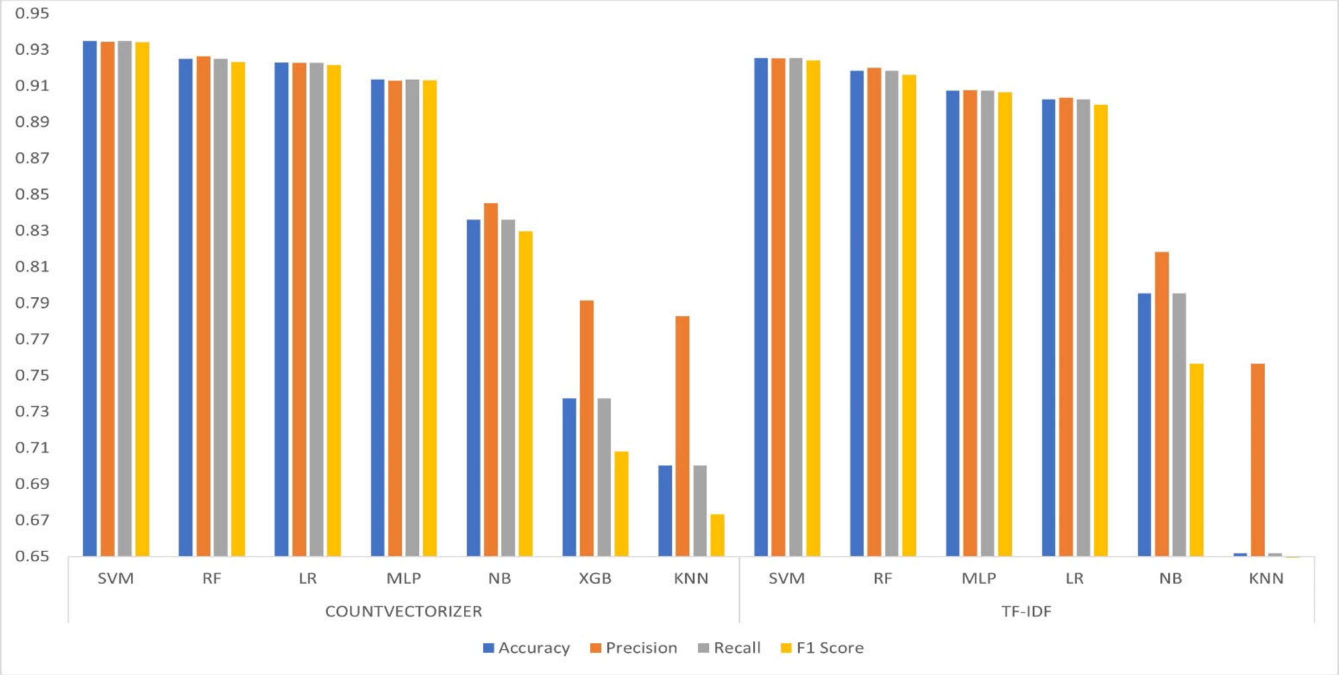

FIGURE 6. Machine Learning Algorithms Performances using TextBlob Labelling and Lemmatization with CountVectorizer or TF-IDF vectorizer. A chart illustrating the performance metrics for each machine learning algorithm in terms of accuracy, precision, recall, and F1 score.

New, People, First, and Sex are the most frequent words used in expressing public opinions on the monkeypox virus. Our Word Cloud exploration result agreed with findings from the Word Frequency analysis.

4) An evaluation of 56 classification models was conducted in this study. Stemming and lemmatization were used in the vocabulary normalization process. Vectorization was performed using the CountVectorizer and TF-IDF techniques. Our learning algorithms included K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Random Forest, Logistic Regression, Multilayer Perceptron (MLP), Naive Bayes, and XGBoost. Several factors were considered to evaluate performance, including Accuracy, F1 Score, Precision, and Recall. We found that the model combining TextBlob annotation, Lemmatization, CountVectorizer, and SVM was the most effective with an accuracy of 93%.

VI. CONCLUSION

The recent outbreak of Monkeypox has raised intense discussion on social media, especially Twitter, with different perspectives from users. Understanding the sentiment behind these expressions is critical. Using massive flow of data and the abundance of opinions, emotions, we seek to obtain important information on social media platforms. We extracted, labeled, and preprocessed collected datasets from tweets on Tweeter. We explored the dataset and built classifiers.

Before training and testing our models, we implemented text normalization and vectorization. 80% and 20% of the dataset were used for training and testing, respectively. Fifty-six (56) models in total were designed, developed and evaluated. The accuracy of these models ranges from 0.65 to 0.93. Models were generated from seven machine learning algorithms with a combination of vectorization, normalization, and annotation techniques. Our finding reflects the importance of sentiments in keeping track of public opinions. We believe that our analysis will help health authorities and individuals to take proactive measures in providing mitigating measures to reduce the spread of the disease.

VII. FUTURE WORK

For future work, additional methods and techniques will be implemented for word embedding (example: doc2Vec) and text labeling (example: Azure Machine Learning) to improve the model's performance. Moreover, we plan to use deep learning and transformer algorithms for a more accurate sentiment analysis and emotion prediction.

REFERENCES

[1] (Jul. 22, 2022). CDC. *MPOX in the U.S.* Centers for Disease Control Prevention. Accessed: Dec. 23, 2022. [Online]. Available: https://www.cdc.gov/poxvirus/monkeypox/about/index.html

[2] A. Mandavilli. (Jul. 23, 2022). *W.H.O. Declares Monkeypox Spread a Global Health Emergency*. The New York Times. Accessed: Dec. 23, 2022. [Online]. Available: https://www.nytimes.com/2022/07/23/health/monkeypox-pandemic-who.html

[3] (Aug. 4, 2022). A. S. P. for Affairs (ASPA). *Biden-Harris Administration Bolsters Monkeypox Response; HHS Secretary Becerra Declares Public Health Emergency*. Accessed: Dec. 23, 2022. [Online]. Available: https://www.hhs.gov/about/news/2022/08/04/biden-harris-administration-bolsters-monkeypox-response-hhs-secretary-becerra-declares-public-health-emergency.html

[4] J.-H. Yoo, "Once bitten, twice shy: Our attitude towards Monkeypox," *J. Korean Med. Sci.*, vol. 37, no. 22, p. e188, 2022.

[5] (Dec. 22, 2022). CDC. *2022 U.S. Map & Case Count*. Accessed: Dec. 23, 2022. [Online]. Available: https://www.cdc.gov/poxvirus/monkeypox/response/2022/us-map.html

[6] U. Naseem, I. Razzak, M. Khushi, P. W. Eklund, and J. Kim, "COVIDSenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 4, pp. 1003–1015, Aug. 2021.

[7] F. Shamrat, S. Chakraborty, M. M. Imran, J. N. Muna, M. M. Billah, P. Das, and O. M. Rahman, "Sentiment analysis on Twitter Tweets about COVID-19 vaccines using NLP and supervised KNN classification algorithm," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, pp. 463–470, 2021.

[8] H. Gupta, S. Pande, A. Khamparia, V. Bhagat, and N. Karale, "Twitter sentiment analysis using deep learning," in *Proc. IOP Conf. Mater. Sci. Eng.*, vol. 1022, no. 1, Jun. 2021, Art. no. 012114.

[9] K. Chakraborty, S. Bhatia, S. Bhattacharyya, J. Platos, R. Bag, and A. E. Hassanien, "Sentiment analysis of COVID-19 Tweets by deep learning classifiers—A study to show how popularity is affecting accuracy in social media," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106754.

[10] T. B. Shahi, C. Sitaula, and N. Paudel, "A hybrid feature extraction method for nepali COVID-19-related Tweets classification," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–11, Mar. 2022.

[11] C. Sitaula, A. Basnet, A. Mainali, and T. B. Shahi, "Deep learning-based methods for sentiment analysis on nepali COVID-19-related Tweets," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–11, Nov. 2021.

[12] C. Sitaula and T. B. Shahi, "Multi-channel CNN to classify nepali COVID-19 related Tweets using hybrid features," 2022, *arXiv:2203.10286*.

[13] P. Chinnasamy, V. Suresh, K. Ramprathap, B. J. A. Jebamani, K. S. Rao, and M. S. Kranthi, "COVID-19 vaccine sentiment analysis using public opinions on Twitter," *Mater. Today, Proc.*, vol. 64, pp. 448–451, Jan. 2022.

[14] M. Qorib, T. Oladunni, M. Denis, E. Ososanya, and P. Cotae, "COVID-19 vaccine hesitancy: Text mining, sentiment analysis and machine learning on COVID-19 vaccination Twitter dataset," *Expert Syst. Appl.*, vol. 212, Feb. 2023, Art. no. 118715.

[15] W. Chung, "eMood: Modeling emotion for social media analytics on Ebola disease outbreak," Tech. Rep., Dec. 2015.

[16] N. Thakur, "MonkeyPox2022tweets: A large-scale Twitter dataset on the 2022 monkeypox outbreak, findings from analysis of Tweets, and open research questions," *Infectious Disease Rep.*, vol. 14, no. 6, pp. 855–883, Nov. 2022, doi: 10.3390/idr14060087.

[17] C. Sitaula and T. B. Shahi, "Monkeypox virus detection using pre-trained deep learning-based approaches," *J. Med. Syst.*, vol. 46, no. 11, p. 78, Oct. 2022, doi: 10.1007/s10916-022-01868-2.

[18] T. B. Alakus and M. Baykara, "Comparison of Monkeypox and wart DNA sequences with deep learning model," *Appl. Sci.*, vol. 12, no. 20, p. 10216, Oct. 2022, doi: 10.3390/app122010216.

[19] Y. Liu, Z. Yue, and M. Anwar, "Monkeypox At-a-glance from Google trends and reddit," Tech. Rep.

[20] V. Rahmanian, K. Jahanbin, and M. Jokar, "Using Twitter and web news mining to predict the monkeypox outbreak," *Asian Pacific J. Tropical Med.*, vol. 15, no. 5, p. 236, 2022.

[21] K. K. Mohbey, G. Meena, S. Kumar, and K. Lokesh, "A CNN-LSTM-based hybrid deep learning approach to detect sentiment polarities on Monkeypox Tweets," 2022, *arXiv:2208.12019*.

[22] (Mar. 26, 2021). *Data Annotation & Labeling—What is it, & Why is it Important? | Shaip*. Accessed: Dec. 23, 2022. [Online]. Available: https://www.shaip.com/blog/the-a-to-z-of-data-annotation/

[23] C. Hutto and E. Gilbert, "VADER: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. Int. AAAI Conf. Web Social Media*, vol. 8, no. 1, 2014, pp. 216–225.

[24] J. P. Gujjar and H. P. Kumar, "Sentiment analysis: Textblob for decision making," *Int. J. Sci. Res. Eng. Trends*, vol. 7, no. 2, pp. 1097–1099, 2021.

[25] A. Bandi and A. Fellah, "Socio-analyzer: A sentiment analysis using social media data," in *Proc. 28th Int. Conf.*, vol. 64, 2019, pp. 61–67.

[26] A. Athar. (Jan. 6, 2021). *Textblob vs VADER For Sentiment Analysis in Python*. Analytics Vidhya. Accessed: Dec. 30, 2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/01/sentiment-analysis-vader-or-textblob/

[27] (Nov. 15, 2022). Amy @GrabNGoInfo. *TextBlob vs. VADER For Sentiment Analysis Using Python*. Accessed: Dec. 30, 2022. [Online]. Available: https://pub.towardsai.net/textblob-vs-vader-for-sentiment-analysis-using-python-76883d40f9ae

[28] P. Shah. (Nov. 6, 2020). *My Absolute Go-To for Sentiment Analysis—TextBlob*. Accessed: Dec. 23, 2022. [Online]. Available: https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524

[29] A. Pano, "A complete VADER-based sentiment analysis of bitcoin (BTC) Tweets during the era of COVID-19," *Big Data Cogn. Comput.*, vol. 4, no. 4, p. 33, 2020.

[30] H. C. Wu, R. W. P. Luk, K. F. Wong, and K. L. Kwok, "Interpreting TF-IDF term weights as making relevance decisions," *ACM Trans. Inf. Syst.*, vol. 26, no. 3, pp. 1–37, Jun. 2008.

[31] Y. Wang, Z. Zhou, S. Jin, D. Liu, and M. Lu, "Comparisons and selections of features and classifiers for short text classification," in *Proc. IOP Conf. Mater. Sci. Eng.*, vol. 261, no. 1, Oct. 2017, Art. no. 012018.

[32] D. Jurafsky and J. Martin, "Naive Bayes and sentiment classification," *Speech Lang. Process.*, pp. 74–91, 2017.

[33] J. Su, J. S. Shirab, and S. Matwin, "Large scale text classification using semisupervised multinomial naive Bayes," in *Proc. ICML*, 2011, pp. 1–8.

[34] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of support vector machine (SVM) learning in cancer genomics," *Cancer Genomics Proteomics*, vol. 15, no. 1, pp. 41–51, Jan./Feb. 2018.

[35] G. Biau and E. Scornet, *A Random Forest Guided Tour*. Springer, 2016.

[36] H. Zhang, J. Zimmerman, D. Nettleton, and D. J. Nordman, "Random forest prediction intervals," *Amer. Statistician*, vol. 74, no. 4, pp. 392–406, Oct. 2020, doi: 10.1080/00031305.2019.1585288.

[37] G. Chen, Y. Ding, and X. Shen, "Sweet KNN: An efficient KNN on GPU through reconciliation between redundancy removal and regularity," in *Proc. IEEE 33rd Int. Conf. Data Eng. (ICDE)*, Apr. 2017, pp. 621–632, doi: 10.1109/ICDE.2017.116.

[38] E. Mankolli, "Reducing the complexity of candidate selection using natural language processing," in *Proc. 29th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jun. 2022, pp. 1–4, doi: 10.1109/IWSSIP55020.2022.9854488.

[39] C. Bento. (Sep. 30, 2021). *Multilayer Perceptron Explained with a Real-Life Example and Python Code: Sentiment Analysis*. Accessed: Dec. 23, 2022. [Online]. Available: https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141

[40] *What is XGBoost?* NVIDIA Data Science Glossary. Accessed: Dec. 23, 2022. [Online]. Available: https://www.nvidia.com/en-us/glossary/data-science/xgboost/

STAPHORD BENGESI received the B.S. degree in computer science from the University of Dar es Salaam (UDSM), Dar es Salaam, Tanzania, in 2014, and the M.S. degree in computer science from Bowie State University, MD, USA, in 2021, where he is currently pursuing the doctoral degree in computer science.

From 2019 to 2021, he was a Research Graduate Assistant at the Computer Science Department, Bowie State University, where he has been an Adjunct Professor with the Computer Science Department, since 2021. His research interests include machine learning, deep learning, and cloud computing.

TIMOTHY OLADUNNI received the master's and doctoral degrees in computer science from Bowie State University, MD, USA, in 2013 and 2017, respectively.

He was an Assistant Professor at the Department of Computer Science and Information Technology, University of the District of Columbia, Washington DC, USA. He is currently an Assistant Professor of computer science with Morgan State University, MD, USA. He explores computer science fundamental concepts in developing sustainable, efficient, and innovative solutions to real world problem. He has a broad research experience in artificial intelligence with specific expertise in natural language processing, computer vision, data science, pattern recognition, and computational epidemiology.

RUTH OLUSEGUN received the B.S. degree in computer science and the M.S. degree in information security from the University of Ilorin, Nigeria, in 2011 and 2015, respectively, and the M.S. degree in computer science with a specialization in AI from Bowie State University, MD, USA, in 2022, where she is currently pursuing the D.S. degree in computer science.

She was an information security analyst at a Financial Institution, Nigeria, where she participated in several projects. Since 2018, she has been working as a Research Assistant at the Department of Computer Science, Bowie State University. She is currently a Research Scientist with AI Squared Inc., MD, USA. Her research interests include artificial intelligence with a special focus on machine learning, deep learning, natural language processing, and Blockchain technologies.

HALIMA AUDU received the bachelor's degree in electrical engineering from Ahmadu Bello University, Nigeria, in 2012, and the master's degree in computer science from Bowie State University, MD, USA, in 2018, where she is currently pursing the doctoral degree in computer science.

Since 2018, she has been an Adjunct Faculty Member with the Department of Computer Science, Bowie State University. Her research interests include cloud computing, artificial intelligence, and cybersecurity.