Pertanika J. Sci. & Technol. 33 (S3): 97 - 121 (2025)

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Public Sentiment on Awareness of Climate Change Based on Support Vector Machine

Norlina Mohd Sabri[1]*, Izzatul Syahirah Ismail[2], Nik Marsyahariani Nik Daud[3] and Nor Azila Awang Abu Bakar[3]

[1]Research and Industrial Linkages Unit, Universiti Teknologi MARA Cawangan Terengganu, Kampus Kuala Terengganu, 21080 Kuala Terengganu, Terengganu, Malaysia
[2]Yayasan Hasanah Level 2, Blok A, Dataran PHB Saujana Resort, Seksyen U2, 40150 Shah Alam, Selangor, Malaysia
[3]College of Computing, Informatics and Mathematics, Universiti Teknologi MARA Cawangan Terengganu, Kampus Kuala Terengganu, 21080 Kuala Terengganu, Terengganu, Malaysia

ABSTRACT

Climate change has threatened human society and natural ecosystems, yet public opinion surveys have found that public awareness and concern are very deficient. If society is unaware of climate change, activities such as open burning, deforestation, and releasing excessive carbon dioxide gases would not be reduced. There are several methods to detect public opinion on climate change, and one of the convenient and efficient methods is conducting sentiment analysis on Twitter. This study uses machine learning techniques to collect and analyze public opinion on climate change from Twitter. Due to the increasing occurrences of natural disasters worldwide, understanding public awareness of climate change is crucial. The objective of the study is to analyze public sentiment on the awareness of climate change based on the Support Vector Machine (SVM) algorithm. The methodology for the study consists of several phases: data collection, pre-processing, labeling, feature extraction and classifier evaluation. The evaluation results indicated that SVM achieved a high accuracy of 91% with an 80:20 data split. The SVM classifier model has also produced high precision, F1-score, and recall results. The government could use the study results and non-governmental organizations (NGOs) to help them spread awareness on climate change issues. Future work will improve the classifier by analyzing non-English tweets and using SentiWordNet to handle word ambiguity in the messages.

Keywords: Awareness, climate change, public sentiment, Support Vector Machine

ARTICLE INFO

Article history:
Received: 22 August 2024
Accepted: 24 February 2025
Published: 24 April 2025

DOI: https://doi.org/10.47836/pjst.33.S3.07

E-mail addresses:
norli097@uitm.edu.my (Norlina Mohd Sabri)
izzatul.syahirah@hasanah.org.my (Izzatul Syahirah Ismail)
nikma944@uitm.edu.my (Nik Marsyahariani Nik Daud)
azila268@uitm.edu.my (Nor Azila Awang Abu Bakar)
*Corresponding author

e-ISSN: 2231-8526

INTRODUCTION

Climate change is one of the important issues today and has often been associated with global warming. This association makes sense since one of the effects of climate change is global warming. According to an encyclopedic entry from National Geographic, climate change refers to long-term shifts in global temperatures and atmospheric characteristics, including typical weather patterns in a specific region (National Geographic Society, 2019). Today, the climate is evolving, with temperatures rising worldwide. Climate change can bring a lot of harm and difficulties to all the living things on this earth if this issue is always ignored. For example, the farmers will face difficulties maintaining and growing crops because of the expected temperature and rainfall levels. Other than that, climate change can cause a rise in sea level and damage the land due to increased flooding and erosion. Therefore, climate change has become a major concern because the impact of climate change is big and irreversible. Climate change is undeniably one of the threats to humans, animals, and the environment today.

Even though the effects of climate change are very obvious today, many people are still ignorant due to many factors, such as economic factors. People all over the world are still lacking awareness and concern for climate change. There is a need to know the public perception periodically on this matter to educate and warn the world. Humans must be aware of this issue and be prepared to prevent climate change. Knowing the public perception of this issue can help boost the preparation to prevent climate change and bring awareness to humans. One of the most convenient and efficient methods to detect public opinion on climate change is sentiment analysis on social media. In general, sentiment analysis focuses on systematic identification, extraction, quantification, and research of subjective data. It also belongs to the broader research area of social media content analysis. Sentiment analysis requires using a range of tools, principally, computational linguistics (Alkhatib et al., 2020; Yogi et al., 2024). Sentiment analysis helps track emotional trends over time and analyze information shared on social networks (Ramanathan et al., 2024; Singleton et al., 2019). In this context, sentiment analysis is used to analyze public opinion on climate change and determine if social media users are falling behind or are aware of the conversation about climate change.

Social media, such as Facebook, Twitter, and Tumblr, have become the trends and the most admired communication medium used on the internet. Each day, millions of texts appear on these micro-blogging sites. Social media serves as a valuable source of information, enabling users to share their opinions on various topics and engage in discussions on current issues without restrictions (Joseph et al., 2024; Loureiro & Alló, 2020). This study has chosen Twitter as one of the most popular microblogging sites, with more than 330 million monthly users worldwide. Twitter has increasingly become a top choice to raise awareness on a variety of topics (Otero et al., 2021; Ram et al., 2024). Analyzing the real-

time monitoring of public opinion about climate change on social media can be utilized for decision-making in problematic situations. Social media can be a valuable source of information to the debate on current issues and allow individuals from different cultures, backgrounds, and preferences to share their opinions and concerns with no restrictions (Alkhatib et al., 2020; Singh et al., 2024). Based on this motivation, the objective of the study is to analyze the public sentiment on the awareness of climate change over Twitter based on the Support Vector Machine (SVM) algorithm. The SVM classifier should be able to classify the tweets into positive (aware) or negative (not aware) sentiments. Machine learning-based sentiment analyses have proven reliable, and the techniques can produce good performance (Singh et al., 2024; Sudhir & Suresh, 2021). Previous studies have classified climate change sentiments tweets from the public using SVM and other algorithms such as Random Forest, Logistic Regression, Naive Bayes, Decision Tree, Neural Network, CNN and also RNN (Anhsori & Shidik, 2024; Anoop et al., 2024; Baguio et al., 2023; Ray & Kumar, 2023; Thenmozhi et al., 2024; Varshney et al., 2022; Wang et al., 2020). In this study, the Support Vector Machine (SVM) algorithm has been chosen due to its exceptional capability in solving various problems related to regressions and classifications (Kumar, 2020; Reddy et al., 2022). Analyzing the public's opinion can help to spread climate change awareness, and sentiment analysis is one of the methods that is convenient and efficient in detecting public views on climate change. Based on similar works that had been conducted on climate change sentiment analysis, many approaches have been proposed. Even with the emergence of new algorithms such as CNN and RNN, algorithms such as SVM are still being applied for text analysis, as seen in the latest works (Ahnsori & Shidik, 2024; Thenmozhi et al.,2024). Thus, for this work, SVM is chosen to solidify the arguments that SVM is still relevant for text analysis.

MATERIALS AND METHODS

The study's methodology consists of data collection, pre-processing, sentiment annotation, feature extraction, classifier training and evaluation phases. This study has proposed the SVM algorithm to solve the climate change sentiment classification problem. The core principle of SVM is to identify linear separators within the search space, which separate the different classes. Figure 1 shows the phases of the methodology for this study. The following discussion explains the research methodology phases.

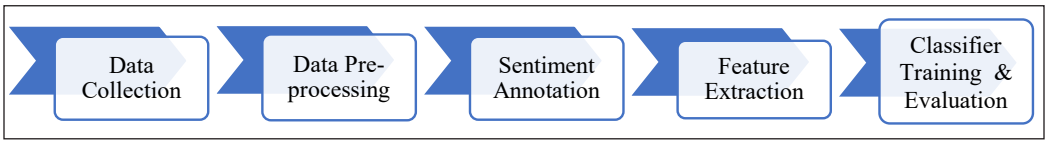

Figure 1. Research methodology phases

Data Collection

In this project, the Twitter API was accessed using Tweepy to scrape data from Twitter. Tweepy is an open-source Python package that facilitates interaction with the Twitter API. It provides a collection of classes and methods representing Twitter's models and API endpoints. Additionally, Tweepy manages various implementation aspects, including data encoding and decoding, HTTP requests, and OAuth authentication. In this project, Tweepy scraped 9244 rows of data using keywords, which were #ActonClimate, #ClimateEmergency, #ClimateReality, #ClimateCrisis, #ClimateChaos and #ClimateAction. Then, the data were stored in a CSV file to proceed to the next phase. This study scraped the Twitter data from March until August 2022. This was the period after the Covid-19 restriction orders were slowly lifted all over the world.

Data Pre-processing

Data pre-processing involves transforming raw data into a structured format, enabling the extraction of valuable information for training the model. It helps to get rid of unhelpful parts of the data. In machine learning processes, data pre-processing is important to ensure that large datasets are formatted so that learning algorithms can interpret and process their data content. Several steps have been taken to process the raw data before continuing with the data labeling. Text Cleaning, Tokenization, Normalization, Lemmatization, and Remove Duplicates are steps.

Text Cleaning

The first step in the text cleaning was lowercasing all the tweets to maintain the flow consistency during the NLP tasks and text mining (Singhal, 2020). Twitter tweets could be labeled or unlabeled, and these tweets are noisy. In order to remove the noise, the tweets must be cleaned first. Twitter usually contains informal sentences, URLs, and emojis, as most Twitter users use spoken language when posting a tweet (Otero et al., 2021)Cleaning is required to help analyze the dataset. Figure 2 shows the code fragment to lowercase all the tweets, including the output.

The second step was to remove any URL links and HTML reference characters in the tweets. Then, the next task was to remove placeholders such as 'LINK,' 'VIDEO' and any filler text that temporarily held any URL and HTML link for typesetting and layout. The following step was to remove Twitter handles and non-letter characters such as punctuation, question marks, symbols such as hashtag (#), slash (/) and other unwanted entities such as â, € and more. Next, stop words were removed to eliminate low-value information from tweets and emphasize essential content. This process helps reduce the dataset size, leading to shorter training times by minimizing the number of tokens involved. Afterward, the

Lowercasing all the tweets

```
1 df_clean =  pd.DataFrame()
2 df_clean['tweet_text'] = tweets.tweet_text.str.lower()
3 df_clean.head()
```

	tweet_text
0	rt @jinnieeee_274: @strwberissa here are the little things but has a big help to save the earth...\n#lettheearthbreath \n#climatecrisis httpsâ€¦
1	rt @trsrupdates: â€œeveryday is earth day, letâ€™s restore our earth!â€□ - @treasuremembers \n\n#scienceprotest\n#lettheearthbreath\n#climatechangeawâ€¦
2	rt @walwrites_: instead of using google, you can use ecosiaã…¡a search app just like google, but the difference is for every search you do inâ€¦
3	rt @sassiesahi: [#letttheearthbreath]\n\nhi, everyone ! you can donate your plastic wastes here. there are places where you can drop-off. yoâ€¦
4	rt @gyuubie: how to save mother earth\nâ€¢save water\nâ€¢save electricity\nâ€¢plant some trees\nâ€¢unplug some uppliances\nâ€¢delete emails\nâ€¢reduce the usâ€¦

Figure 2. The code fragment to lowercase the tweets and the output

cleaned data were ready to proceed to the next steps: tweet normalization, tokenization and lemmatization.

Tokenization

Tokenization separates a piece of text into smaller units called tokens to build blocks of Natural Language (Pai, 2022). The following tokens are then used to prepare vocabulary for the count vectorizer and boost the purpose of the SVM model. Figure 3 shows the code snippet for tokenization using the TweetTokenizer function from the nltk.tokenize library. The figure also shows the tokenized words column in the output.

Tokenization

```
1 from nltk.tokenize import TweetTokenizer
2 from nltk.stem import WordNetLemmatizer
3 tknzr = TweetTokenizer()
4 df_clean['tokenized'] = df_clean['remove_short'].apply(tknzr.tokenize)
5 df_clean.head()
```

	tweet_text	remove_short	tokenized
0	here are the little things but has big help sa...	little things big help save earth lettheearthb...	[little, things, big, help, save, earth, letth...
1	everyday earth day let restore our earth scie...	everyday earth day let restore earth sciencepr...	[everyday, earth, day, let, restore, earth, sc...
2	instead using google you can use ecosia search...	instead using google use ecosia search app lik...	[instead, using, google, use, ecosia, search, ...
3	lettheearthbreath hi everyone you can donate ...	lettheearthbreath hi everyone donate plastic w...	[lettheearthbreath, hi, everyone, donate, plas...
4	how save mother earth save water save electric...	save mother earth save water save electricity ...	[save, mother, earth, save, water, save, elect...

Figure 3. Code snippet for tokenization and the output

Normalization

Data normalization aims to improve the cohesion of entry types and organize the data to make it consistent across all contents. Since tweets often contain non-standard words such

as acronyms and misspelled words, normalization is helpful in reducing the number of unique tokens in the data. Figure 4 shows the output after the data is normalized.

	tweet_text	remove_short	tokenized	normalized_tweet
0	here are the little things but has big help save the earth lettheearthbreath climatecrisis https	little things big help save earth lettheearthbreath climatecrisis https	[little, things, big, help, save, earth, lettheearthbreath, climatecrisis, https]	[big, help, save, earth, lettheearthbreath, climatecrisis, http]
1	everyday earth day let restore our earth scienceprotest lettheearthbreath climatechangeaw	everyday earth day let restore earth scienceprotest lettheearthbreath climatechangeaw	[everyday, earth, day, let, restore, earth, scienceprotest, lettheearthbreath, climatechangeaw]	[day, let, restore, earth, scienceprotest, lettheearthbreath, climatechangeaw]
2	instead using google you can use ecosia search app just like google but the difference for every search you	instead using google use ecosia search app like google difference every search	[instead, using, google, use, ecosia, search, app, like, google, difference, every, search]	[google, use, ecosia, search, app, like, google, difference, every, search]
3	lettheearthbreath hi everyone you can donate your plastic wastes here there are places where you can drop off	lettheearthbreath hi everyone donate plastic wastes places drop	[lettheearthbreath, hi, everyone, donate, plastic, wastes, places, drop]	[everyone, donate, plastic, waste, place, drop]
4	how save mother earth save water save electricity plant some trees unplug some uppliances delete emails reduce the	save mother earth save water save electricity plant trees unplug uppliances delete emails reduce	[save, mother, earth, save, water, save, electricity, plant, trees, unplug, uppliances, delete, emails, reduce]	[earth, save, water, save, electricity, plant, tree, unplug, uppliances, delete, email, reduce]

Figure 4. The output after the data is normalized

Lemmatization

Lemmatization is a technique used in NLP to switch words to their base root form. The process removes inflectional suffixes and prefixes to return a word to its dictionary form (Wali, 2022). According to Singhal (2022), lemmatization is better than stemming. Sometimes, stemming loses the actual meaning of the words even though lemmatization and stemming have the same goal: returning the base or dictionary form of a word. Therefore, this study adopted the lemmatization over stemming technique.

Remove Duplicates

The last step in data pre-processing was to remove duplicates. Duplicate tweets can disrupt the division of train, validation, and test sets, potentially causing biased performance in the Support Vector Machine model (Chorev, 2021). After removing duplicates, 4037 rows of tweets were left. Figure 5 shows the tweets after the data pre-processing phase was completed.

Sentiment Annotation

Sentiment annotation, or data labeling, involves identifying raw data and assigning meaningful and informative labels to provide context for the machine learning model to learn effectively. Labeled data is also used to train the NLP models to make predictions or understand the text. This study used a Python library called Textblob to label the data. Firstly, the sentiment polarity function was called to return the polarity of the TextBlob. Figure 6 shows the code snippet for labeling using the sentiment polarity function and the output.

	tweet_text	remove_short	normalized_tweet	grams	tokenized	pos_tags	wordnet_pos	lemmatized
0	here are the little things but has big help save the earth lettheearthbreath climatecrisis https	little things big help save earth lettheearthbreath climatecrisis https	[strwberissa, little, thing, big, help, save, earth, lettheearthbreath, climatecrisis, http]	[strwberissa little, little thing, thing big, big help, help save, save earth, earth lettheearthbreath, lettheearthbreath climatecrisis, climatecrisis http, strwberissa little thing, little thing big, thing big help, big help save, help save earth, save earth lettheearthbreath, earth lettheearthbreath climatecrisis, lettheearthbreath climatecrisis http]	[little, things, big, help, save, earth, lettheearthbreath, climatecrisis, https]	[(little, JJ), (things, NNS), (big, JJ), (help, NN), (save, VB), (earth, JJ), (lettheearthbreath, JJ), (climatecrisis, NN), (https, NN)]	[(little, a), (things, n), (big, a), (help, n), (save, v), (earth, a), (lettheearthbreath, a), (climatecrisis, n), (https, n)]	[little, thing, big, help, save, earth, lettheearthbreath, climatecrisis, http]
1	everyday earth day let restore our earth scienceprotest lettheearthbreath climatechangeaw	everyday earth day let restore earth scienceprotest lettheearthbreath climatechangeaw	[everyday, earth, day, let, restore, earth, treasuremembers, scienceprotest, lettheearthbreath, climatechangeaw]	[everyday earth, earth day, day let, let restore, restore earth, earth treasuremembers, treasuremembers scienceprotest, scienceprotest lettheearthbreath, lettheearthbreath climatechangeaw, everyday earth day, earth day let, day let restore, let restore earth, restore earth treasuremembers, earth treasuremembers scienceprotest, treasuremembers scienceprotest lettheearthbreath, scienceprotest lettheearthbreath climatechangeaw]	[everyday, earth, day, let, restore, earth, scienceprotest, lettheearthbreath, climatechangeaw]	[(everyday, JJ), (earth, DT), (day, NN), (let, VBD), (restore, VB), (earth, JJ), (scienceprotest, JJS), (lettheearthbreath, NN), (climatechangeaw, NN)]	[(everyday, a), (earth, n), (day, n), (let, v), (restore, v), (earth, a), (scienceprotest, a), (lettheearthbreath, n), (climatechangeaw, n)]	[everyday, earth, day, let, restore, earth, scienceprotest, lettheearthbreath, climatechangeaw]
2	instead using google you can use ecosia search app just like google but the difference for every search you	instead using google use ecosia search app like google difference every search	[instead, using, google, use, ecosia, search, app, like, google, difference, every, search]	[instead using, using google, google use, use ecosia, ecosia search, search app, app like, like google, google difference, difference every, every search, instead using google, using google use, google use ecosia, use ecosia search, ecosia search app, search app like, app like google, like google difference, google difference every, difference every search]	[instead, using, google, use, ecosia, search, app, like, google, difference, every, search]	[(instead, RB), (using, VBG), (google, NN), (use, NN), (ecosia, JJ), (search, NN), (app, NN), (like, IN), (google, NN), (difference, NN), (every, DT), (search, NN)]	[(instead, r), (using, v), (google, n), (use, n), (ecosia, a), (search, n), (app, n), (like, n), (google, n), (difference, n), (every, n), (search, n)]	[instead, use, google, use, ecosia, search, app, like, google, difference, every, search]

Figure 5. The results after data pre-processing

```
df_clean.lemmatized= df_clean.lemmatized.astype(str)
df_clean['label'] = ''
for i,x in df_clean.lemmatized.iteritems():
    label = TextBlob(x)
    df_clean['label'][i] = label.sentiment.polarity
    print("Index: ", i ,  "label" , label.sentiment.polarity)
```

```
Index:  0 label 0.0
Index:  1 label 0.0
Index:  2 label 0.0
Index:  3 label -0.2
Index:  4 label 0.0
Index:  5 label -0.075
Index:  6 label 0.2
Index:  7 label 0.0
Index:  8 label 0.5
Index:  9 label 0.0
Index:  10 label 0.0
```

Figure 6. Code snippet for labeling data and the output

In Figure 6, the ‘label’ contains a float from -1 to 1 corresponding to each text. This label column will be transformed into a categorical column using the code in Figure 7.

```
df_clean.lemmatized= df_clean.lemmatized.astype(str)
def polarity_to_label(x):
    if(x >= -1 and x < 0):
        return 'neg'
    if(x == 0):
        return 'neutral'
    if(x > 0 and x <= 1):
        return 'pos'
df_clean.label = df_clean.label.apply(polarity_to_label)
```

Figure 7. Code snippet to categorize the label floats

If the polarity is between -1 and 0, it is labeled as negative; if the polarity ranges between 0 and 1, it is positive. Otherwise, it is neutral. Figure 8 below shows the counts of each data set based on their classes.

```
df_clean.label.value_counts()
```

```
neutral    2167
pos        1273
neg         598
Name: label, dtype: int64
```

Figure 8. Counts of each data based on their classes

However, this study does not require data that was labeled as neutral. Only positive and negative data were used in this classification problem. Therefore, the neutral-labeled data were removed, leaving only 1870 rows of data to proceed to the next step.

Figure 9 shows the labeled data with an additional column 'sentiment' that changes the negative label to -1 and the positive label to 1. Value 1 indicates that the public is aware of climate change, while value -1 indicates that the public is unaware of climate change.

	tweet_text	remove_short	tokenized	normalized_tweet	pos_tags	wordnet_pos	lemmatized	label	sentiment
3	lettheearthbreath hi everyone you can donate your plastic wastes here there are places where you can drop off	lettheearthbreath hi everyone donate plastic wastes places drop	[lettheearthbreath, hi, everyone, donate, plastic, wastes, places, drop]	[everyone, donate, plastic, waste, place, drop]	[(everyone, NN), (donate, NN), (plastic, NN), (waste, NN), (place, NN), (drop, NN)]	[(everyone, n), (donate, n), (plastic, n), (waste, n), (place, n), (drop, n)]	['everyone', 'donate', 'plastic', 'waste', 'place', 'drop']	neg	-1
5	ways prevent climate change all can something about small action can help lot you can	ways prevent climate change something small action help lot	[ways, prevent, climate, change, something, small, action, help, lot]	[climate, change, something, small, action, help, lot]	[(climate, NN), (change, NN), (something, NN), (small, JJ), (action, NN), (help, NN), (lot, NN)]	[(climate, n), (change, n), (something, n), (small, a), (action, n), (help, n), (lot, n)]	['climate', 'change', 'something', 'small', 'action', 'help', 'lot']	neg	-1
6	shit getting real lettheearthbreath scientistprotest	shit getting real lettheearthbreath scientistprotest	[shit, getting, real, lettheearthbreath, scientistprotest]	[real, lettheearthbreath, scientistprotest]	[(real, JJ), (lettheearthbreath, NN), (scientistprotest, NN)]	[(real, a), (lettheearthbreath, n), (scientistprotest, n)]	['real', 'lettheearthbreath', 'scientistprotest']	pos	1
8	look the hashtag lettheearthbreath tops twitter trending list the philippines today april fil	look hashtag lettheearthbreath tops twitter trending list philippines today april fil	[look, hashtag, lettheearthbreath, tops, twitter, trending, list, philippines, today, april, fil]	[lettheearthbreath, top, twitter, trending, list, philippine, today, april, fil]	[(lettheearthbreath, NN), (top, JJ), (twitter, NN), (trending, VBG), (list, NN), (philippine, NN), (today, NN), (april, VBP), (fil, NN)]	[(lettheearthbreath, n), (top, a), (twitter, n), (trending, v), (list, n), (philippine, n), (today, n), (april, v), (fil, n)]	['lettheearthbreath', 'top', 'twitter', 'trend', 'list', 'philippine', 'today', 'april', 'fil']	pos	1

Figure 9. Labeled data

Feature Extraction

Feature extraction converts raw data into numerical features while retaining the essential information from the original dataset. The process can be accomplished manually or automatically using either method, such as Term Frequency-Inverse Document Frequency (TF-IDF) or Bag of Words (BOW). This study has implemented the TF-IDF method using sklearn. TF-IDF is used to calculate the weight of each word. In this technique, words with higher TF-IDF weights are regarded as more representative and kept, while those with lower weights will be discarded. The Term Frequency (TF) of a particular term (t) is calculated as the number of times a term occurs in a document and is divided by the total number of words. Inverse Document Frequency (IDF) is used to calculate the importance of a term because some terms occur frequently but are not important such as "is," "are," "also," "the," and many more (Ahuja et al., 2019). In other words, TF measures how frequently a word occurs in the text, while IDF decreases the weight of terms that occur very frequently and increases the weight of terms that rarely occur instead (Shofiya & Abidi, 2021). The output is shown in Figure 10 to get a clearer glimpse of the IDF values. The word 'climate' is expected to have the lowest IDF values since this word appears in every document in the tweets collection. The lower the IDF value of a word, the less unique the word.

	idf_weights
climate	2.844403
climateaction	3.009310
climateemergency	3.319827
climatecrisis	3.738972
climatechange	3.781172

Figure 10. IDF values for the most frequent words

Support Vector Machine Implementation

Support Vector Machine (SVM) is a supervised machine learning algorithm capable of addressing regression and classification problems (Arora, 2020). The SVM algorithm has been particularly prominent in text classification performance in recent years. In this study, after the data has been processed and divided, the data is ready to be processed by the SVM classifier. SVM is an algorithm that constructs a line or hyperplane to divide data into distinct classes. The objective of SVM is to establish an optimal decision boundary that separates an n-dimensional space into distinct classes, enabling accurate classification of new data in the future. The first step is defining the kernel matrix. A kernel in SVM is a function that simplifies complex computations, efficiently solving classification problems. The SVM kernel in this project can be defined by the K matrix shown in Equation 1.

$$K_{jk} = k(\vec{x}_j, \vec{x}_k) = \vec{x}_j \cdot \vec{x}_k \quad [1]$$

The kernel matrix defining the transformation is symmetric. Figure 11 shows the K matrix construction to be used in the SVM classifier of this project, and Figure 12 shows the K values obtained. The structure of the kernel matrix is an array.

```
k_value = np.array(X_train @ X_train.T + np.identity(len(y_train))*1e-12)
pd.set_option('display.max_columns', None)
k_value
```

Figure 11. Code snippet for finding K matrix

```
array([[ 1.00000000e+00,  0.00000000e+00,  0.00000000e+00, ...,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00,  1.00000000e+00,  0.00000000e+00, ...,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00,  0.00000000e+00,  1.00000000e+00, ...,
         0.00000000e+00,  0.00000000e+00,  0.00000000e+00],
       ...,
       [ 7.47161478e-02,  0.00000000e+00,  0.00000000e+00, ...,
         1.24410253e-06,  0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00,  6.81259001e-02,  0.00000000e+00, ...,
         1.06596945e-12,  1.09194231e-06,  0.00000000e+00],
       [ 0.00000000e+00,  3.85339450e-02,  0.00000000e+00, ...,
        -3.70580822e-14, -5.02160834e-13,  1.14222655e-06]])
```

Figure 12. K-values obtained

Then, the next step is to set up and minimize the dual function given the constraints using cvxpy tools. The tools allow the user to express a convex optimization problem in a readable form, convert it into a format that can be used to call a solver and translate the result into a readable form. This step is crucial before recreating the hyperplane of the SVM classifier. The code snippet is shown in Figure 13 below.

```
alpha = cp.Variable(shape=y_train.shape) # Create optimization variables.

beta = cp.multiply(alpha, y_train) # to simplify notation

K = cp.Parameter(shape=k_value.shape, PSD=True, value=k_value)

# objective function
obj = .5 * cp.quad_form(beta, K) - np.ones(alpha.shape).T @ alpha

# constraints
const = [np.array(y_train.T) @ alpha == 0,
        -alpha <= np.zeros(alpha.shape),
        alpha <= 10*np.ones(shape=alpha.shape)]
prob = cp.Problem(cp.Minimize(obj), const)
result = prob.solve()
```

Figure 13. Code snippet to minimize the dual function

The next step is to recreate the hyperplane. The code to recreate the hyperplane is shown in Figure 14.

```
w = np.multiply(y_train, alpha.value).T @ X_train
```

```
S = (alpha.value > 1e-4).flatten()
b = y_train[S] - X_train[S] @ w
b = b[0]
b = np.mean(b)
```

Figure 14. Code snippet to recreate the hyperplane

Equation 2 shows how the plane's parameter is modified to fulfill the SVM plane's formula.

$$\vec{w} \cdot \vec{x} - b = 0 \qquad [2]$$

The above formula is also equivalent to $\vec{w} \cdot \vec{x} = b$, which is the equation of the separating hyperplane (Siong, 2019). w is the normal direction of the plane, and b is a form of threshold. If $w \cdot x$ is calculated to be bigger than b, it belongs to a class. If not, then it belongs to another class. Ultimately, after the hyperplane is created, the SVM classifier is built by deploying the obtained hyperplane. Figure 15 shows the code snippet to build the SVM classifier.

```
def classify(x):
    result = w @ x + b
    return np.sign(result)
```

```
correct = 0
incorrect = 0
predictions = []
for i in X_test:

    my_svm = classify(i)

    predictions.append(my_svm)

predictions = np.array(predictions)
```

Figure 15. Code to build the SVM classifier

Classifier Training and Performance Evaluation

Classifier training is an important phase in which the SVM classifier learns to distinguish between positive and negative sentiments. Before the training, the dataset has to be split

into the training and testing ratios based on the hold-out method. The classifier is then evaluated using the appropriate performance measurements. The performance evaluation plays a significant role in accuracy measurement through a sentiment analysis word level (Mohamed & El-din, 2017). The performance of the SVM classifier can be evaluated based on accuracy, precision, recall, F1-Score, and receiver operating characteristic (ROC). One of the common techniques used to evaluate the performance of sentiment analysis is the Confusion Matrix.

Confusion Matrix

The confusion matrix is a tool used to evaluate the performance of machine learning classification models, accommodating two or more output classes. It is particularly valuable for assessing recall, precision, and accuracy. This matrix is commonly employed to illustrate how well a classification model performs on a test dataset with known actual values (Markham, 2020). A confusion matrix is also used to summarize the prediction results of a classification problem (Brownlee, 2020). Figure 16 shows the confusion matrix. The rows represent the instances in an actual class, while the columns represent the instances in a predicted class.

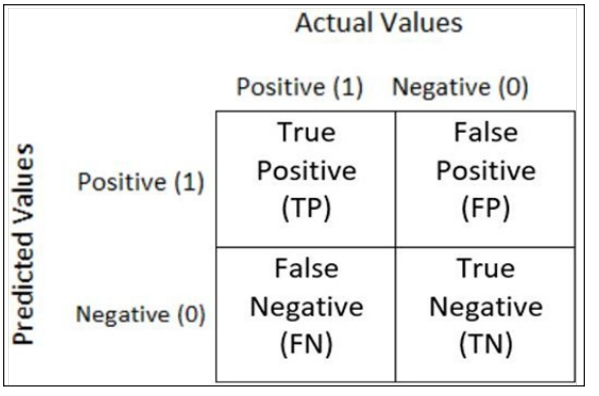

Figure 16. Confusion matrix

The basic terms in the confusion matrix are true positive (TP), true negative (TN), false positive (FP), and false negatives (FN). These basic terms are used to calculate the rates often computed from a confusion matrix. The first-rate that can be computed from a confusion matrix is accuracy. Accuracy measures how often a sentiment rating is correct. The accuracy can be calculated using Equation 3.

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

Next, the second rate is precision, which measures the exactness of a classifier. Higher precision means fewer false positives. Otherwise, lower precision means more false positives. Equation 4 shows how the precision of the project can be obtained.

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

The confusion matrix can also compute the recall rate and F-measure. Recall rates measure the completeness or sensitivity of a classifier. A higher recall means fewer false negatives, while a lower recall means more false negatives. The F1 Score estimates the accuracy of a test by considering Precision and Recall. The Recall and F1 Score can be calculated using Equations 5 and 6.

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

$$F1\ Measure = \frac{2\ (Precision * Recall)}{(Precision + Recall)} \quad [6]$$

ROC Curve

The ROC curve is a metric used to evaluate classification models across different threshold settings, indicating the model's ability to distinguish between classes (Narkhede, 2021). It can also be described as a graphical representation that assesses the diagnostic performance of binary classifiers (Chan, 2020). The ROC curve is constructed by plotting the True Positive Rate (TPR) on the y-axis against the False Positive Rate (FPR) on the x-axis. To summarize the performance of a classifier, the common approach is to calculate the area under the ROC curve (AUC). An excellent model has a value of AUC near 1. It means it has a good measure of separability (Narkhede, 2021).

RESULTS AND DISCUSSION

This section provides the results of the SVM classifier model performance evaluation. It covers the Confusion Matrix evaluation, the hold-out method, and the AUC calculation results. The elaboration of results also includes data exploration, which has been done through word cloud generation.

Confusion Matrix Results

One suitable tool for evaluating behavior and understanding the effectiveness of a categorical classifier is the confusion matrix. The confusion matrix is a 2-dimensional

array comparing predicted labels to the true label. For binary classification, the categories are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Figure 17 shows this study's confusion matrix results based on the 80:20 split of data training and testing.

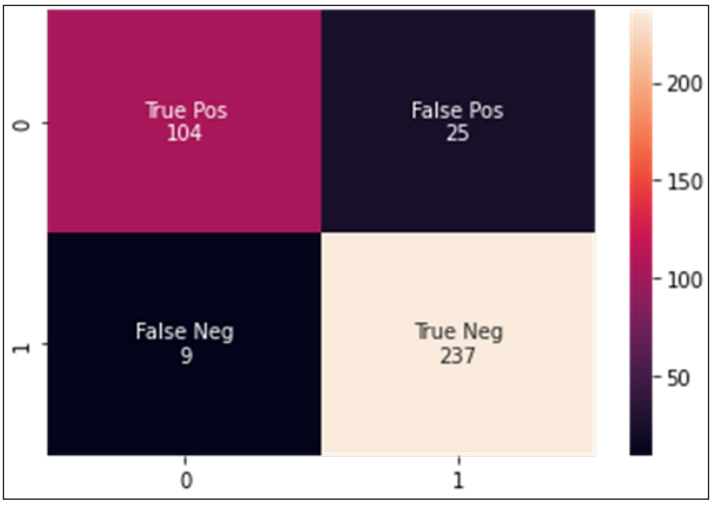

Figure 17. Confusion matrix of the classifier

Based on Figure 17, the diagonal from the top left to the bottom right contains the correctly predicted observations. Based on the confusion matrix above, the total predictions are 375. TP shows that there are 104 correctly predicted positive instances, while FP shows that there are 25 incorrectly predicted positive instances by the model. There are nine instances where the model incorrectly predicted the negative class (FN). On the contrary, the classifier model has correctly predicted 237 negative instances (TN). The total number of correct predictions is 341 (TP+TN), while the incorrect predictions are 34 (FN+FP). The confusion matrix shows that the SVM model performs well overall, with high accuracy and recall. Although the confusion matrix gives a detailed view of the total number of predictions, understanding how good the model is at classifying awareness of climate change samples is complicated. Therefore, the classification report is built to show the metrics that quantify the model's performance.

Classification Report

A classification report is a performance evaluation metric that shows the SVM classifier model's precision, recall, F1 Score, and support. In this project, the classification report also represents the confusion matrix. Table 1 shows all the metrics in the classification report to better understand the result obtained.

Table 1
Metrics in classification report (Kharwal, 2021)

Metrics	Definition
Precision	Describes how well the model can predict the labels correctly.
Recall	Describes how the model can retrieve all of the labels correctly.
F1 Score	The weighted average of precision and recall.
Support	The number of actual occurrences of the class in the dataset.

A function from the sklearn.metrics library called classification_report is used to generate the classification report. Figure 18 shows the code fragment to view this project's machine learning model's classification report, and Figure 19 shows the output.

```
1 from sklearn.metrics import accuracy_score, f1_score, precision_score, recall_score,
2 classification_report, confusion_matrix
3 print(classification_report(y_test,predictions))
```

Figure 18. Code to display the classification report on the classifier

	precision	recall	f1-score	support
-1	0.92	0.81	0.86	129
1	0.90	0.96	0.93	246
accuracy			0.91	375
macro avg	0.91	0.88	0.90	375
weighted avg	0.91	0.91	0.91	375

Figure 19. Classification report on the classifier

Figure 19 displays the classification report, which summarizes the performance of the SVM classification model across two classes, positive (class 1) and negative (class -1), based on an 80:20 data split. For the positive class (class 1), the precision shows that 90% of predictions were correct, and the recall shows that 96% of actual instances were correctly classified. The value of 0.93 of the F1-score shows a very strong performance for the positive class (class 1). As for the negative class (class -1), the precision shows that 92% of predictions were correct, and the recall shows that 81% of actual instances were correctly classified. The value of 0.86 of the F1-score shows the harmonic mean of precision and recall, indicating strong but slightly lower recall for the negative class (class -1).

The overall metrics show that the model correctly classified 91% of all instances. The macro average results show the values of 0.91, 0.88 and 0.90 for the precision, recall and F1-score, respectively. The macro average shows the unweighted average of the metrics for

both classes. It treats both classes equally, regardless of their support. As for the weighted average, the results show the values 0.91 for the precision, recall, and the F1-score. This weighted average considers each class's support (number of samples), making it more representative of the actual class distribution.

Based on the classification report, the model performs very well overall, with an accuracy of 91%. The positive class (class 1) has slightly lower precision but higher recall, indicating that the model effectively captures most positive cases (96%). The negative class (class -1) has higher precision but lower recall, meaning some negative instances are not classified as positives. The F1 score shows a good balance between precision and recall, particularly for the positive class (class 1). The classification report reflects strong performance for both classes. However, in the future, the model could be further improved for the negative class (class -1) to increase recall (reduce false negatives). This might involve balancing the dataset, fine-tuning the model, or adjusting classification thresholds.

Hold-out Method

The hold-out method is a straightforward and widely used technique for assessing the performance of machine learning models. It involves dividing the dataset into distinct subsets for training and testing, allowing evaluation of the model's ability to generalize new, unseen data. In classification problems, different training and testing data sizes usually can affect the classifier's performance. Therefore, it is necessary to compare the ratio of data splits through the hold-out method. Table 2 shows the accuracy, precision, F1-score and recall rates when the data are divided into 60:40, 70:30 and 80:20 ratios.

Table 2
Comparison between values of evaluation metrics with different ratios of data splitting

Evaluation Metrics	**Training and Testing Ratio**		
	60:40	**70:30**	**80:20**
Accuracy	86%	87%	91%
Precision	88%	88%	91%
F1-Score	83%	85%	90%
Recall	81%	84%	88%

Table 2 shows a slight difference in the evaluation metrics values between the ratios. However, when the data are divided into 80% training and 20% testing, the results could exceed 90%. From the table, it can be seen that the higher the amount of testing data, the better the results will be. Thus, the best data split for this project is 80% for data training and 20% for data testing. The accuracy and precision obtained are 91%, which shows how well the model could classify the sentiments correctly. The high F1-score of 90% represents

the harmonic means between the precision and recall results. The accuracy result in this study is also higher than the result of the SVM classification (82.9%) in one of the similar works (Ruz et al., 2020). The accuracy of SVM in this study has shown that the algorithm could generate good and reliable performance in this classification problem.

ROC Curve

The receiver operating characteristic (ROC) curve represents a classification model's performance across various thresholds. It plots two key metrics: true and false positive rates. Figure 20 shows the ROC curve results for the best model with a split data ratio of 80:20.

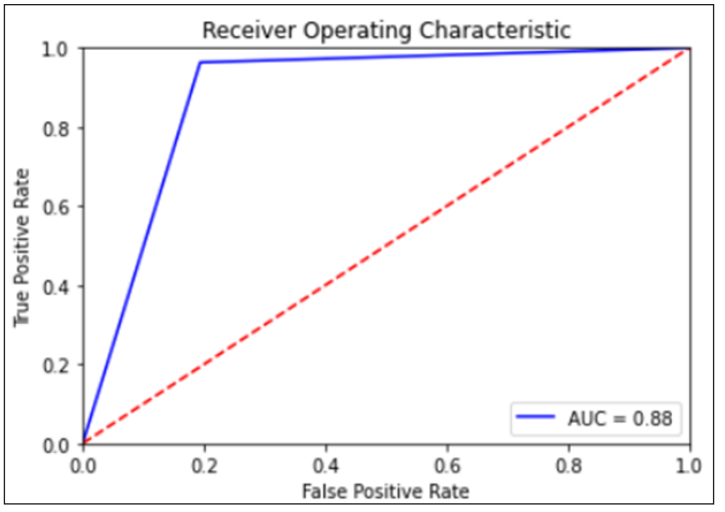

Figure 20. ROC Curve for the best model

In a ROC curve, a higher X-axis value indicates a higher number of FP than TN. Meanwhile, a higher Y-axis value shows a higher number of TP than FN. The area under the ROC curve (AUC) provides the information needed to compute the points in an ROC curve. Based on Figure 20, the ROC curve rises steeply towards the top-left corner of the graph, indicating a high True Positive Rate (TPR-sensitivity) with a low False Positive Rate (FPR). The steep rise in the curve suggests that the model achieves a high TPR while keeping the FPR low. The SVM model has an 88% probability of correctly differentiating between randomly selected positive and negative instances. The model performs much better than random guessing (AUC = 0.5). The value 0.88 indicates that the SVM model performs very well distinguishing between positive and negative classes. The AUC for this SVM classifier with a data split of 80:20 has shown good performance with a 0.88 value. Bhandari (2022) states that a higher AUC indicates better model performance in distinguishing between the two classes.

The ROC curves for the other two models based on the data split of 60:40 and 70:30 are shown in Figure 21. Based on the figure, the green curve represents the data split of 70:30, while the red curve represents the 60:40 data split. The green curve shows that the classifier model performs slightly better than the other model, meaning it has a higher true positive rate for a given false positive rate. The **green curve is slightly higher than the red curve**, indicating better classification performance with more training data (70% training vs 60% training). In this research, increasing the training data ratio has **improved classification performance**. This can be proven by the performance of the best model (AUC 0.88), which has more training data with an 80:20 data split. Based on Figure 21, the **AUC values** confirm that the 70:30 model (AUC 0.788) has a slightly better overall performance than the 60:40 model (AUC 0.730).

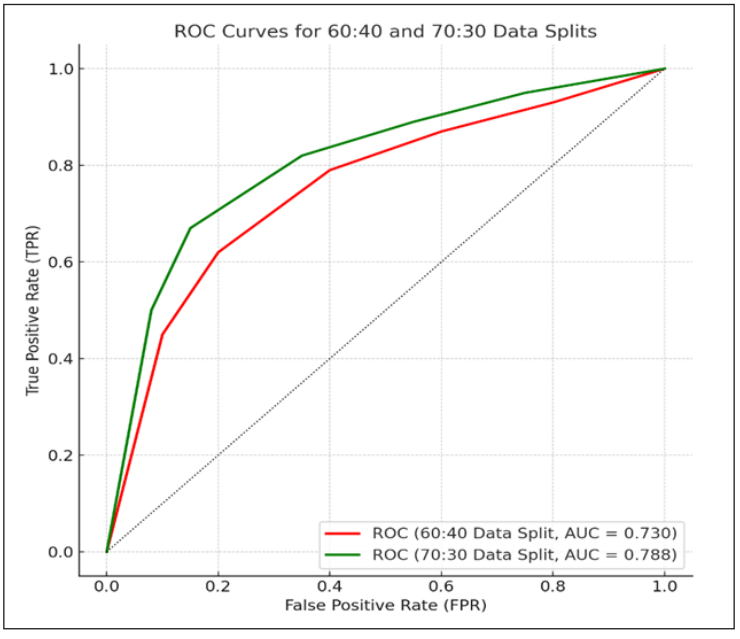

Figure 21. ROC Curve for the other two models

Word Cloud

A word cloud is an electronic image showing the collection of words in different sizes or series of texts. The words' size differs according to how often they are found in the text (Boost Labs, 2020). A word cloud is an excellent option for quickly gaining insight into the most used words and helps to interpret the text visually. The word cloud is generated for this project to give additional information about the dataset. Figure 22 shows the word cloud for the positively labeled tweets. The figure highlights the word ‘climate action’ since this word appeared in most of the text. In Figure 22, words such as ‘solutions,’ ’save’ and

'change' show that the positively labeled tweets are aware of climate change. Based on Figure 8, more positive tweets have been scrapped than negative ones. This shows that people were actually aware of climate change on Twitter.

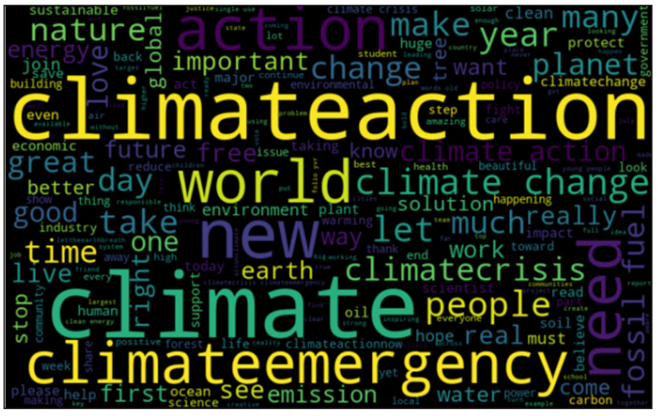

Figure 22. Word cloud for positively labeled tweets

Comparison Between Similar Works

This section compares the proposed project's accuracy values with similar climate change projects that utilized various methods and algorithms. Table 3 shows the accuracy score for each system.

Table 3
Comparison of accuracy between similar work

References	Algorithm/Methods	Accuracy
Anoop et al. (2024)	Support Vector Machine	88.66%
Thenmozhi et al. (2024)	Support Vector Machine	76%
Anhsori & Shidik (2024)	Support Vector Machine	75.42%
Proposed system	Support Vector Machine	91%
Ruz et al. (2020)	**Dataset 1**	
	Random Forest	72.5%
	Naïve Bayes	74.2%
	Tree-augmented Naïve Bayes	72.1%
	Support Vector Machine	81.2%
	Dataset 2	
	Random Forest	85.8%
	Naïve Bayes	78.1%
	Tree-augmented Naïve Bayes	80.8%
	Support Vector Machine	82.9%

Based on Table 3, the research carried out by Anoop et al. (2024) and Ruz et al. (2020) has generated good performances with an accuracy of more than 80% for the SVM classifier. As for research done by Thenmozhi et al. (2024) and Anhsori and Shidik (2024), the SVM classifier has also generated more than 70% acceptable accuracies. This similar research has analyzed the public sentiments on climate change based on Twitter data. In the research by Ruz et al. (2020), the dataset has been divided into two datasets. In dataset 1, the SVM algorithm gives the best accuracy score compared to other algorithms that have been implemented. Meanwhile, in dataset 2, the SVM algorithm still scores a high accuracy of more than 80%. Also shown in the table, the proposed system achieved a score of 91%, which is higher accuracy than other comparable works carried out using different algorithms. In light of this, it can be shown that the suggested project is more effective and superior in categorizing tweets about climate change awareness.

Discussion

The evaluation parameters used to evaluate the performance of this project's SVM classifier model are the accuracy score, confusion matrix, classification report, and ROC curve. This section elaborates on the evaluation results and discusses the possible outcomes if the training and test sizes differ. The first evaluation parameter used in this project is accuracy. Accuracy is defined as the ratio of correct predictions to the total dataset size. In this study, the model achieved an accurate score of 91%. As Barkved (2022) stated, accuracy above 70% indicates an optimal model performance. High accuracy indicates how well the model correctly predicts all labels. However, as noted by Khalid (2021), higher accuracy does not necessarily mean exceptional model performance, as accuracy considers only the overall correct predictions without accounting for the performance of individual labels. Accordingly, a confusion matrix, classification report, and ROC curve were generated to assess the machine learning model's performance quantitatively. In accordance with the results obtained in this section, the SVM classifier model in this project is excellent and reliable in classifying the sentiment on awareness of climate change among the public.

Contribution of Study

It is important to make the public realize the consequences if no action is taken to prevent climate change. This study applies data mining on Twitter to find the sentiment on awareness of climate change for the targeted users for this project, which are authorities such as governments and non-governmental organizations (NGOs), to help them tackle the climate change issue. The realization of creating a society that is very aware is vital to ensure they practice the initiatives to sustain the environment. Therefore, this study highlights how the targeted users can use Twitter for beneficial use, especially in finding a small quantity of awareness to address climate change issues. Besides that, this study also contributes to

knowing the trending words that Twitter users use to crusade about climate change. With this study, society can also launch an awareness campaign and educate the people who tweet ignorant comments about climate change.

As for the machine learning algorithm implemented in this study, SVM showed good performance with a high accuracy of 91%. For several reasons, researchers in sentiment analysis are still implementing SVMs compared to deep learning models. First, SVM is computationally efficient, requiring less computational power and training time compared to deep learning models, which is particularly advantageous when resources are limited. Second, SVM performs well with smaller datasets, while deep learning typically requires large-scale data to avoid overfitting and achieve high accuracy (Mustapha et al., 2024). Third, SVM is easier to interpret and tune compared to the complex architectures of deep learning models, making them suitable for studies focusing on simplicity and transparency (Thenmozhi et al., 2024). Additionally, in some sentiment analysis tasks with well-structured features, SVM can achieve comparable performance to deep learning, providing a robust baseline for comparison (Maada et al., 2022). This makes SVM a valuable tool for research and applications where simplicity, speed, or data scarcity are concerns.

CONCLUSION

This study successfully analyzed public sentiments on the awareness of climate change based on the Support Vector Machine (SVM) algorithm. The SVM classifier could classify the Twitter data into positive (aware) and negative (not aware) polarities with a good accuracy of 91%. This shows that the SVM classifier is capable and reliable in solving this sentiment classification problem. The exploratory data analysis found that more people were aware of climate change situations from the tweets. The significance of the study is that the government could use the study's results and non-governmental organizations (NGOs) to help them increase public awareness and tackle the climate change issue. In Malaysia, the Ministry of Natural Resources, Environment and Climate Change is responsible for handling environmental problems. Environmental NGOs are vital in raising climate change awareness within communities and supporting governments in formulating and implementing adaptive measures. The public needs to be alerted to act before it is too late to save themselves from climate consequences. The increasing impacts of climate change have made humanity and nature suffer from extreme weather, worst storms, unusually heavy rain, more flooding, melting glaciers, and rising sea levels. Preventing climate change is beneficial to humans, wildlife, and nature. As for the SVM classifier, the recommendation for future work is to make an option for a language other than English or to create a multilingual classifier model. Currently, the classifier is able to process tweets in English only. Another suggestion is to use word ambiguity approaches such as SentiWordNet to define polarity based on the sentence context, due to the incorrectly classified tweets if the

users use sarcasm or double-meaning words. Future works also include comparing SVM with other well-known classification algorithms for sentiment analysis, namely the Naive Bayes, Random Forest, and deep learning algorithms such as CNN and RNN.

ACKNOWLEDGEMENT

Special thanks to Universiti Teknologi MARA Cawangan Terengganu for its unwavering support in advancing research at the university. The authors also thank everyone involved, directly or indirectly, for their insightful and constructive feedback.

REFERENCES

Ahuja, R., Chug, A., Kohli, S., Gupta, S., & Ahuja, P. (2019). The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, *152*, 341-348. https://doi.org/10.1016/j.procs.2019.05.008

Alkhatib, M., Barachi, M. El, Samuel Mathew, S., & Oroumchian, F. (2020). Using artificial intelligence to monitor the evolution of opinion leaders' sentiments: Case study on global warming. In *2020 5th International Conference on Smart and Sustainable Technologies (SpliTech)* (pp. 1-6). IEEE. https://doi.org/10.23919/SpliTech49282.2020.9243726

Anhsori, K., & Shidik, G. F. (2024). Comparison performance of SVM, Naïve Bayes and XGBoost Classifier on climate change issue. In *2024 International Seminar on Application for Technology of Information and Communication (iSemantic)* (pp. 1-6). IEEE. https://doi.org/10.1109/iSemantic63362.2024.10762214

Anoop, V. S., Krishnan, T. K. A., Daud, A., Banjar, A., & Bukhari, A. (2024). Climate change sentiment analysis using domain specific bidirectional encoder representations from transformers. *IEEE Access, 12*, 114912-114922. https://doi.org/10.1109/access.2024.3441310

Arora, S. (2020, February 4). SVM: Difference between Linear and Non-Linear Models. *AITUDE.* https://www.aitude.com/svm-difference-between-linear-and-non-linear-models/

Baguio, J. D. S., Lu, B. A., & Peña, C. F. (2023). Text classification of climate change tweets using artificial neural networks, fasttext word embeddings, and latent dirichlet allocation. In *2023 International Conference in Advances in Power, Signal, and Information Technology (APSIT)* (pp. 688-692). IEEE. https://doi.org/10.1109/APSIT58554.2023.10201782

Barkved, K. (2022, March 9). How to know if your machine learning model has good performance. *Obviously AI.* https://www.obviously.ai/post/machine-learning-model-performance

Bhandari, A. (2022, June 14). AUC-ROC curve in machine learning clearly explained. *Analytics Vidhya.* https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

Boost Labs. (2020, November 3). What are word clouds? The value of simple visualizations. *Boost Labs.* https://boostlabs.com/blog/what-are-word-clouds-value-simple-visualizations/#:%7E:text=Word%20clouds%20(also%20known%20as,words%20depicted%20in%20different%20sizes

Brownlee, J. (2020, August 15). What is a confusion matrix in machine learning? *Machine Learning Mastery.* https://machinelearningmastery.com/confusion-matrix-machine-learning/

Chan, C. (2020, December 9). What is a ROC curve and how to interpret it. *Displayr.* https://www.displayr.com/what-is-a-roc-curve-how-tointerpret-it/

Chorev, S. (2021, August 9). What is data cleaning: A practical guide. *Deepchecks.* https://deepchecks.com/what-is-data-cleaning/

Joseph, V., Lora, C. P., & Narmadha, T. (2024). Exploring the application of natural language processing for social media sentiment analysis. In *2024 3rd International Conference for Innovation in Technology (INOCON)* (pp. 1-6). IEEE. https://doi.org/10.1109/INOCON60754.2024.10511841

Khalid, I. A. (2021, December 14). Greater accuracy does not mean greater machine learning model performance. *Medium.* https://towardsdatascience.com/greater-accuracy-does-not-mean-greater-machine-learning-model-performance-771222345e61

Kharwal, A. (2021, July 7). Classification report in machine learning. *Thecleverprogrammer.com.* Article 1774. https://thecleverprogrammer.com/2021/07/07/classification-report-in-machine-learning/

Kumar, S. (2020, February 4). SVM: Difference between Linear and Non-Linear Models. *AITUDE*. https://www.aitude.com/svm-difference-between-linear-and-non-linear-models/

Loureiro, M. L., & Alló, M. (2020). Sensing climate change and energy issues: Sentiment and emotion analysis with social media in the U.K. and Spain. *Energy Policy, 143,* 111490. https://doi.org/10.1016/j.enpol.2020.111490

Maada, L., Al Fararni, K., Aghoutane, B., Fattah, M., & Farhaoui, Y. (2022). A comparative study of sentiment analysis machine learning approaches. In *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)* (pp. 1-5). IEEE. https://doi.org/10.1109/IRASET52964.2022.9738346

Markham, K. (2020, February 3). Simple guide to confusion matrix terminology. *Data School.* https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Mohamed, D. M. E. D., & El-din, M. H. N. (2017). Performance analysis for sentiment techniques evaluation perspectives. In *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017* (pp. 448-457). Springer International Publishing. https://doi.org/10.1007/978-3-319-64861-3_42

Mustapha, W. N. A. W., Sabri, N. M., Abu Bakar, N. A. A., Nik Daud, N. M., & Azizan, A. (2024). Detection of harassment toward women in Twitter during pandemic based on machine learning. *International Journal of Advanced Computer Science and Applications*, *15*(3), 1035-1043. https://doi.org/10.14569/IJACSA.2024.01503103

Narkhede, S. (2021, June 15). Understanding AUC - ROC curve - Towards data science. *Medium.* https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

National Geographic Society. (2019, March 27). *Climate Change*. https://www.nationalgeographic.org/encyclopedia/climate-change/

Otero, P., Gago, J., & Quintas, P. (2021). Twitter data analysis to assess the interest of citizens on the impact of marine plastic pollution. *Marine Pollution Bulletin, 170.* https://doi.org/10.1016/j.marpolbul.2021.112620

Pai, A. (2022, June 21). What is Tokenization in NLP? Here's all you need to know. *Analytics Vidhya.* https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

Ram, N. R., Gautum, S., Jadeja, A., Joisar, H., & Rathore, N. (2024). Social media sentiment analysis using Twitter dataset. In *2024 1st International Conference on Cognitive, Green and Ubiquitous Computing (IC-CGU)* (pp. 1-5). IEEE. https://doi.org/10.1109/IC-CGU58078.2024.10530694

Ramanathan, V., Al Hajri, H., & Ruth, A. (2024). Conceptual level semantic sentiment analysis using Twitter data. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)* (pp. 1-8). IEEE. https://doi.org/10.1109/ADICS58448.2024.10533498

Ray, S., & Kumar, A. M. S. (2023). Prediction and analysis of sentiments of reddit users towards the climate change crisis. In *2023 International Conference on Networking and Communications (ICNWC)* (pp. 1-7). IEEE. https://doi.org/10.1109/ICNWC57852.2023.10127496.

Reddy, M. B. K., Vani, B., & Babu, C. N. K. (2022). A comparative analysis for the detection of hit rate of popular music videos in social network using logistic regression over support vector machine algorithm. In *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 544-549). IEEE. https://doi.org/10.1109/ICIEM54221.2022.9853055

Ruz, G. A., Henríquez, P. A., & Mascareño, A. (2020). Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers. *Future Generation Computer Systems*, *106*, 92-104. https://doi.org/10.1016/j.future.2020.01.005

Shofiya, C., & Abidi, S. (2021). Sentiment analysis on COVID-19-related social distancing in Canada using twitter data. *International Journal of Environmental Research and Public Health*, *18*(11), 5993. https://doi.org/10.3390/ijerph18115993

Singh, G., Singh, J., & Tripathy, B. (2024). Significance of sentiment analysis approaches using Machine Learning (ML) techniques. In *2024 2nd International Conference on Computer, Communication and Control (IC4)* (pp. 1-6). IEEE. https://doi.org/10.1109/IC457434.2024.10486310

Singhal, G. (2020, October 5). Importance of text pre-processing. *Pluralsight.* https://www.pluralsight.com/guides/importance-of-text-pre-processing

Singleton, S., Kumar, S. A. P., & Li, Z. (2019). Twitter analytics-based assessment: Are the United States coastal regions prepared for climate change. *In Proceedings of the International Symposium on Technology and Society (ISTAS)* (pp. 150-155). IEEE. https://doi.org/10.1109/ISTAS.2018.8638266

Siong, T. G. (2019, December 13). What are w and b parameters in SVM? *Cross Validated.* https://stats.stackexchange.com/users/128610/siong-thye-goh

Sudhir, P., & Suresh, V. D. (2021). Comparative study of various approaches, applications and classifiers for sentiment analysis. *Global Transitions Proceedings, 2*(36), 205-211. https://doi.org/10.1016/j.gltp.2021.08.004

Thenmozhi, M., Shubigsha, G., Sindhuja, G., & Dhinakar, V. (2024). Sentiment analysis on climate change using Twitter data. *In 2024 2nd International Conference on Networking and Communications (ICNWC)* (pp. 1-6). IEEE. https://doi.org/10.1109/icnwc60771.2024.10537404

Varshney, H., Shishodiya, P., & Sriramulu, S. (2022). Classifying tweets based on climate change. In *2022 3rd International Conference on Intelligent Engineering and Management (ICIEM)* (pp. 956-960). IEEE. https://doi.org/10.1109/ICIEM54221.2022.9853094.

Wali, K. (2022, May 4). Explained: Stemming vs lemmatization in NLP. *Analytics India Magazine.* https://analyticsindiamag.com/explained-stemming-vs-lemmatization-in-nlp

Wang, J., Obradovich, N., & Zheng, S. (2020). A 43-million-person investigation into weather and expressed sentiment in a changing climate. *One Earth*, *2*(6), 568-577. https://doi.org/10.1016/j.oneear.2020.05.016

Yogi, K. S., Gowda, V. D., Sindhu, D., Soni, H., Mukherjee, S., & Madhu, G. C. (2024). Enhancing accuracy in social media sentiment analysis through comparative studies using machine learning techniques. In *2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS)* (Vol. 1, pp. 1-6). IEEE. https://doi.org/10.1109/ICKECS61492.2024.10616441