

REFERENCES

- [1] H. Prasetyo and A. S. Fitrani, "Sentiment Analysis Before Presidential Election 2024 Using Naïve Bayes Classifier Based On Public Opinion In Twitter," *Jurnal RESTI*, vol. 8, no. 3, pp. 568–574, 2024. [Online]. Available: <https://doi.org/10.29207/resti.v8i3.6460>
- [2] O. Oueslati, M. B. Hajhmida, H. Ounelli, and E. Cambria, "Sentiment Analysis of Influential Messages for Political Election Forecasting," in *Proc. of the Int. Conf. on Computer Systems and Applications (AICCSA)*, IEEE, 2019. [Online]. Available: <https://www.researchgate.net/publication/337325392>
- [3] A. Nugroho, "US Presidential Election 2020 Prediction Based on Twitter Data Using Lexicon-Based Sentiment Analysis," *Indonesian Journal of Computing and Cybernetics Systems*, vol. 15, no. 2, pp. 131–142, 2021. [Online]. Available: <https://doi.org/10.22146/ijccs.66244>
- [4] P. Chauhan, N. Sharma, and G. Sikka, "The Emergence of Social Media Data and Sentiment Analysis in Election Prediction," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, pp. 10041–10065, 2021. [Online]. Available: <https://doi.org/10.1007/s12652-020-02423-y>
- [5] R. Martins, J. J. Almeida, P. Henriques, and P. Novais, "Predicting an Election's Outcome Using Sentiment Analysis," in *Proc. 2020 Int. Conf. on Intelligent Systems (IS)*, IEEE, 2020, pp. 1–8. [Online]. Available: <https://doi.org/10.1109/IS.2020.00020>
- [6] W. Budiharto and M. Meiliana, "Prediction and Analysis of Indonesia Presidential Election from Twitter Using Sentiment Analysis," *J. Big Data*, vol. 5, no. 1, p. 51, 2018. [Online]. Available: <https://doi.org/10.1186/s40537-018-0164-1>
- [7] O. Oyeboade and R. Orji, "Social Media and Sentiment Analysis: The Nigeria Presidential Election 2019," in *Proc. 10th Int. Conf. on Information Technology, Electronics and Mobile Communication (IEMCON)*, IEEE, 2019, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/IEMCON.2019.8936139>
- [8] R. Prodhani, A. U. Islam, and L. Das, "Election Prediction Using Twitter Sentiment Analysis," *J. of Global Research in Computer Science*, vol. 13, no. 3, Jun. 2022. [Online]. Available: <https://doi.org/10.4172/2229-371X.13.3.006>
- [9] A. F. Hidayatullah, S. Cahyaningtyas, and A. M. Hakim, "Sentiment Analysis on Twitter Using Neural Network: Indonesian Presidential Election 2019 Dataset," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1077, 012001, 2021. [Online]. Available: <https://doi.org/10.1088/1757-899X/1077/1/012001>
- [10] A. Yavari, H. Hassanpour, B. Rahimpour Cami, and M. Mahdavi, "Event Prediction in Social Network Through Twitter Messages Analysis," *Social Network Analysis and Mining*, vol. 12, no. 1, p. 78, Jul. 2022. [Online]. Available: <https://doi.org/10.1007/s13278-022-00911-x>

APPENDIX

```
# Install necessary libraries
import pandas as pd
import numpy as np
import re
import string
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Embedding,
Dropout
from keras.callbacks import EarlyStopping
import nltk
from textblob import TextBlob

from imblearn.over_sampling import SMOTE
from tensorflow.keras.regularizers import l2
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import to_categorical
from sklearn.utils.class_weight import compute_class_weight
from tensorflow.keras.layers import BatchNormalization # Import
BatchNormalization
from keras.callbacks import ReduceLROnPlateau
from google.colab import files

# Ambil data dan label
X = df['cleaned_tweet'].values
y = df['sentiment'].values

# Stratified split 65/17.5/17.5
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.35,
stratify=y, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp,
test_size=0.5, stratify=y_temp, random_state=42)

# Stratified split 55/22.5/22.5
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.55,
stratify=y, random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp,
test_size=0.5, stratify=y_temp, random_state=42)
```