

Sentiment Analysis of ChatGPT Using the KNN Algorithm and K-Fold Cross-Validation Optimization of the K Value

Azliza Yacob^{1*}, Nurzal Effiyana Ghazali², Faez M. Hassan³

¹Department of Computer Science, University College TATI, Terengganu 24000 (Malaysia)

²Centre for Engineering Education, Universiti Teknologi Malaysia, Johor 81310, (Malaysia)

³Department of Physics, College of Education, Mustansiriyah University, Baghdad (Iraq)



* Corresponding author: azliza@uctati.edu.my

Received 6 September 2025 | Accepted 25 October 2025 | Early Access 30 November 2025

ABSTRACT

ChatGPT is a cutting-edge artificial intelligence language model powered by advanced machine learning technologies, such as GPT-4.0. Its remarkable ability to generate human-like text and engage in interactive conversations has captured widespread attention, particularly on social media. As a result, public sentiment toward ChatGPT has become a significant topic, necessitating detailed sentiment analysis to comprehend the broader societal reactions to this technology. This study focuses on optimizing the K value in sentiment analysis applied to Twitter data about ChatGPT. Selecting the appropriate K value is crucial, as improper values can result in overfitting or underfitting the model. The research methodology includes several stages: data collection, pre-processing, feature extraction, k-fold cross-validation for K optimization, implementing the K-Nearest Neighbors (KNN) algorithm, and evaluating results. The analysis determined that the optimal K value for this sentiment analysis is K=9. Using this value, the KNN algorithm achieved an accuracy of 88%, indicating robust performance in classifying sentiment effectively. These findings highlight the potential of this approach to provide meaningful insights into public perception and sentiment regarding ChatGPT on social media platforms. This result not only underscores the technical effectiveness of KNN for sentiment analysis but also demonstrates the practical application of machine learning in understanding societal trends in the context of emerging AI technologies.

KEYWORDS

Sentiment Analysis, ChatGPT, K-Fold Cross Validation, KNN, K Optimal.

I. INTRODUCTION

GENERATIVE Pretrained Transformer (GPT) is an advanced language model that combines natural language processing (NLP) with machine learning techniques to produce high-quality text that mimics human writing across various languages and contexts [1],[2]. One of the most popular implementations of GPT, ChatGPT, has significantly impacted how humans interact with technology. These impacts include positive aspects, such as enhanced communication efficiency, and negative aspects, such as concerns over human job replacement by technology [3],[4]. The primary strength of ChatGPT lies in its ability to generate text of remarkable quality, but this phenomenon has also sparked debates in society, particularly on social media platforms like Twitter [5],[6],[7].

Twitter, with over 368 million active users in 2023, serves as a major platform for analysing public opinions about ChatGPT [8]. Tweets discussing ChatGPT reflect a range of sentiments positive, negative, and neutral that can be analysed

using sentiment analysis techniques [9]. Sentiment analysis is crucial for understanding public responses to this new technology, offering insights that can guide its development to better align with needs and expectations [10],[11].

This research employs sentiment analysis using the K-Nearest Neighbour (KNN) algorithm enhanced with the k-fold cross-validation method to determine the optimal value of k [12]. KNN is chosen for its adaptability, simplicity in classification processes, and proven ability to yield high evaluation metrics [13]. The k value in KNN significantly affects classification accuracy, as a low value can lead to underfitting, while a high value may cause overfitting [14]. To address this, the k-fold cross-validation method is applied to iteratively partition the dataset and improve accuracy [15],[16].

Previous research has highlighted the importance of selecting the optimal k value. For example, research by [17] emphasized the influence of the k value on the accuracy of sentiment analysis using KNN on Indonesian text. Their research concluded that improper selection of k could reduce

Please cite this article as: Yacob, A., Effiyana Ghazali, N., & M. Hassan, F. (2025). Sentiment Analysis of ChatGPT Using the KNN Algorithm and K-Fold Cross-Validation Optimization of the K Value. International Journal of Informatics and Computing, 1(2), 49-55.

model performance. Additionally, Agustina et al. [18] presented that data validation techniques affect sentiment analysis accuracy, revealing differences in effectiveness between hold-out and k-fold cross-validation methods.

Building on these previous findings, this study aims to evaluate the effectiveness of k-fold cross-validation in enhancing the accuracy of sentiment analysis on ChatGPT using the KNN algorithm. The research focuses on how variations in the k value during k-fold cross-validation

influence accuracy in the context of sentiment analysis on Twitter data. Consequently, this study seeks to contribute to the development of more effective sentiment analysis methods for emerging technologies like ChatGPT.

II. METHODOLOGY

Figure 1 below illustrates the schematic of the method employed for sentiment analysis in this study.

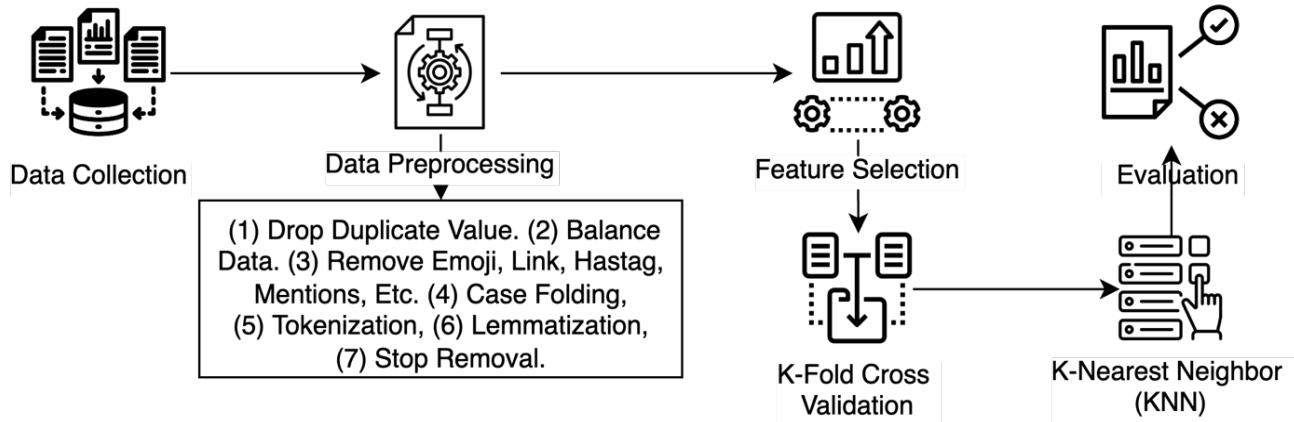


Fig. 1. Methodology

A. Data Collection

The data collection in this study was carried out by searching for the appropriate dataset for Chat GPT sentiment analysis from various dataset provider platforms. The dataset search process was conducted by considering various factors, such as dataset size, dataset quality, and dataset availability. The data collection process involved evaluating datasets from platforms known for providing high-quality information, including independently collected datasets as well as those provided by leading data providers. Accuracy and diversity of the datasets were the main criteria in the selection process, aiming to create a representative and relevant data foundation.

B. Data Preprocessing

This stage is a crucial step in data analysis. It involves transforming raw data into a more structured and analyzable format. The primary goal of data preprocessing is to remove noise and inconsistencies from the data, ensuring that subsequent analysis is more accurate and reliable. In this research, several text preprocessing techniques are employed, including value removal, data balancing, case folding, emoji removal, numerical data removal, tokenization, and stop word removal.

C. Feature Extraction

Feature extraction is the process of transforming textual data into numerical representations that can be used by classification models. Word embedding is one of the most effective feature extraction methods. Word embedding converts words into numerical vectors that represent their meaning and context. These vectors can capture both semantic and syntactic relationships between words. Word embeddings enhance the accuracy of machine learning models by capturing these relationships, which are essential for

understanding the meaning of text and classifying it correctly.

D. K-fold Cross-validation

This technique used to evaluate and test the performance of a machine learning model, helping to compare and select the most suitable model [19]. In this process, the optimal value of K is determined using the K-fold cross-validation method. First, the data is divided into several equal folds, for example, five folds, where the data is split into five equal parts. Then, the range of K values is set, for instance, from 1 to 10. The model is evaluated by applying cross-validation, and in each iteration, the model is assessed using appropriate performance metrics. This evaluation is performed for each K value within the specified range.

E. K-Nearest Neighbor (KNN)

This machine learning algorithm that works by calculating the distance between a test point and the data points in the training set. KNN is a classification algorithm that uses distance metrics to classify new data. The value of K is an integer specified by the user. Once the K nearest neighbors is found, the new data is classified into the most common class among these neighbors. In the context of sentiment analysis, K-Nearest Neighbor (KNN) is a relevant and effective algorithm. The implementation of KNN for sentiment analysis involves determining the K nearest neighbors of new sentiment data based on distance metrics.

F. Evaluation

At this stage, the trained model will be evaluated using a confusion matrix. The confusion matrix is an essential tool for assessing the performance of a machine learning model. By using this metric, we can understand how well the model is able to correctly classify instances and identify uncertainties in its predictions.

III. RESULT AND ANALYSIS

The following section presents the implementation results of the proposed Discrete Wavelet Transform (DWT)-based watermarking scheme. The algorithm operates by decomposing pixel intensity values of the host image into frequency-domain components, enabling watermark embedding within selected sub-bands. This transformation facilitates controlled modification of image coefficients while preserving perceptual quality.

In this study, a level 1 two-dimensional DWT decomposition is employed, producing four sub-bands: Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH). Among these, the LH and HL sub-bands are primarily utilized for PSNR-based imperceptibility analysis, as they capture significant edge and texture information while maintaining a balanced trade-off between robustness and visual fidelity [10]. Embedding within these mid-frequency components reduces visible distortion compared to spatial-domain approaches and enhances resistance to minor signal processing operations.

Prior to conducting robustness evaluation through controlled attack simulations, it is essential to outline the watermark embedding workflow implemented in the proposed system. This workflow includes image preprocessing, DWT decomposition, coefficient modification for watermark insertion, inverse DWT reconstruction, and quantitative quality assessment using MSE and PSNR metrics. Understanding this structured insertion mechanism provides a clear foundation for interpreting the robustness and performance results discussed in the subsequent sections.

TABLE I. Sample Dataset

No	Tweets	Labels
1.	ChatGPT: Enhancing Language Models for Conversations...	Neutral
2.	Engage in a conversation with ChatGPT,...	Good
3.	ChatGPT: Enhancing Language Models for Conversations.	Good
4.	Just two minutes ago, @OpenAI unveiled their latest release.	Bad

The dataset in Table I represents a labeled collection of tweets categorized into three sentiments: Neutral, Good, and Bad. While it showcases diversity in tweet content, such as promotional and announcement-like messages, it suffers from critical issues like redundancy, with one tweet labeled inconsistently as both Neutral and Good, and an overall small sample size of four entries, which limits its utility for training robust models. The sentiment distribution, with two Good, one Neutral, and one Bad, suggests a potential class imbalance. To improve the dataset, expanding the sample size, resolving label inconsistencies, and ensuring equal representation of sentiments are essential steps to enhance its quality and effectiveness for sentiment analysis tasks.

A. Data Preprocessing

Subsequent to the collection of data, a series of processes are employed for the purpose of data cleansing. The objective of these processes is the elimination of noise and inconsistencies from the data. These steps are carried out to ensure that the data is ready for use in ChatGPT sentiment

analysis. The data preprocessing steps include:

1. Drop Null and Duplicate Values

Null data is defined as data that is devoid of value or unknown in nature. In contrast, duplicate data refers to identical data that manifests itself multiple times within the dataset. It is imperative to meticulously cleanse both types of data prior to analysis, as their presence can compromise the accuracy and reliability of the ensuing analysis results.

As illustrated in Figure 2, the dataset has been subjected to a process of elimination to remove both null values and duplicates. The initial dataset comprised 219,294 rows. However, subsequent to the removal of duplicate entries, the number of rows was reduced to 217,623. A total of 1,671 rows were eliminated due to the presence of duplicate data, thus ensuring a more accurate and streamlined dataset.

```

Duplicated values: 0
<class 'pandas.core.frame.DataFrame'>
Int64Index: 217623 entries, 0 to 219293
Data columns (total 2 columns):
#   Column  Non-Null Count  Dtype
---  -
0   tweets  217623 non-null  object
1   labels  217623 non-null  object
dtypes: object(2)
memory usage: 5.0+ MB
None

```

Fig. 2. Dataset Not Null and No Duplicated Values

2. Balance Data

In the context of machine learning, class imbalance refers to a situation where the number of samples in one class of a dataset significantly exceeds the number of samples in another class. This imbalance has the potential to adversely impact model performance, as the model may become biased toward the majority class. One approach to address class imbalance is resampling, which involves the reduction of the number of samples in the majority class (the class with more samples) to balance the dataset. This is referred to as undersampling. The objective is to minimize the dominance of the majority class, thereby enabling the model to more effectively learn from both classes. As illustrated in Table 2, the undersampling technique was employed, where the number of samples in the "Bad" label was reduced to 70,000.

TABLE II. Balance Data Results

Label	Dataset Before Balance	Dataset After Balance
Bad	106695	70000
Good	55754	55754
Neutral	55174	55174

3. Removing emojis, links, mentions, hashtags, punctuation, symbols and irrelevant characters

As illustrated in Table 3, the preprocessing stage entailed the elimination of emojis, links, mentions, hashtags, punctuation, symbols, and other extraneous characters. This procedure was implemented to enhance the integrity and relevance of the data. The cleanup was executed through the utilization of the remove function in Python, thereby ensuring that the dataset was rendered more concise and more suitable for subsequent analysis. By eradicating noise and non-informative elements, the process facilitated an enhancement in the emphasis on meaningful text features.

TABLE III. Results Removing Emoji, Hashtag Links, Punch Marks

Original Tweets	Tweets After Remove
OpenAI just dropped their new ChatGPT! Get ready to chat with AI, it's live now: 🙌 https://t.co/VyPGPNw988	OpenAI just dropped their new ChatGPT Get ready to chat with AI its live now

4. Case Folding

This stage, the process of converting all uppercase letters into lowercase, is a crucial technique for standardizing text format, enhancing data analysis accuracy, streamlining the analytical process, and improving overall efficiency. This technique ensures uniformity in text, thereby eliminating potential inconsistencies that could arise from variations in letter case. Consequently, it reduces noise in the dataset. Table 4 illustrates the results of case folding, which was performed using the Python lower () function. This process ensures that all text is consistently in lowercase, making it more suitable for subsequent analysis, improving model performance, and reducing bias in text processing.

TABLE IV. Results of Case Folding

Tweets Before Case Folding	Tweets After Case Folding
OpenAI just dropped their new ChatGPT Get ready to chat with AI its live now	openai just dropped their new chatgpt get ready to chat with ai its live now

5. Tokenizing

This process of breaking text into smaller units, such as words or tokens. In this study, tokenization is performed using two functions from the Python library: *word_tokenize* and *is_alpha()*. The *word_tokenize* function is used to split the text into words based on spaces and punctuation marks, while the *is_alpha()* function checks whether a token consists only of alphabetic characters, filtering out tokens that contain numbers or non-alphabetic symbols. The result of this tokenization process is presented in Table 5, which shows the words extracted after tokenization and filtering.

TABLE V. Results of Tokenizing

Tweets Before Tokenizing	Tweets After Tokenizing
openai just dropped their new chatgpt get ready to chat with ai its live now	'openai', 'just', 'dropped', 'their', 'new', 'chatgpt', 'get', 'ready', 'to', 'chat', 'with', 'ai', 'its', 'live', 'now'

6. Lemmatization

In this research, lemmatization is defined as the process of converting a word to its base or root form, with the

lemmatize() function of the *WordNetLemmatizer* class being implemented for this purpose. This function utilizes the WordNet dictionary to determine and return the base form of a word, thereby standardizing words to their canonical forms and ensuring consistency in textual analysis. The reduction of words to their lemma significantly enhances the quality and consistency of the dataset, facilitating more accurate analysis and interpretation, particularly in tasks such as sentiment analysis, topic modeling, information retrieval, and natural language understanding across varied contexts.

TABLE VI. Results of Lemmatization

Before Lemmatization	After Lemmatization
'as', 'of', 'minutes', 'ago', 'released', 'their', 'new', 'chatgpt', 'and', 'you', 'can', 'use', 'it', 'right', 'now'	as of 2 minutes ago, released their new chatgpt and you can use it right now

7. Stopword Removing

This stage represents common words that frequently appear in a language but hold minimal independent meaning. In the English language, common stopwords include "the", "a", "an", "in", "of", and "to". These words frequently provide little to no useful information in text analysis and can hinder performance in tasks such as natural language processing, machine learning, and search engine optimization. To address this challenge, stopwords are typically removed using techniques such as list comprehensions, which filter out words present in a predefined *stop_words* list, leaving only meaningful terms for further analysis. This process ensures that the text is cleaner, more concise, and more relevant for computational or linguistic tasks.

TABLE VII. Results of Stopword Removing

Before Stopword Removing	After Stopword Removing
a of 2 minute ago, released their new chatgpt and you can use it right now	minute ago, released new chatgpt use right

8. Convert Text to Numerical

At this stage, the label column is converted from text to numeric format using LabelEncoder. This process maps each unique label to an integer index, enabling more efficient data representation for further analysis. The encoded labels are then added as a new column to the DataFrame. For example, the label *bad* is converted to the numeric value 0, as shown in Table 8, which illustrates the results of converting text to numerical format. This approach simplifies data processing for machine learning models.

TABLE VIII. Results of Covert Text to Numerical

No	Index	Tweets	Cleaned_tweets	Numeric_tweets	Labels	Encoded labels
0	4	As of 2 minutes ago, @OpenAI released their ne...	minute ago, released new chatgpt use right	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...]	Bad	0
1	6	As of 2 minutes ago, @OpenAI released their ne...	minute ago, released new chatgpt use right	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...]	Bad	0
2	10	GOD DAMN IT @OpenAI STOP ANNOUNCING THINGS I A...	god damn stop announcing thing busy	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...]	Bad	0

B. Feature Extraction

The feature extraction method employed in this study is

word embedding, specifically the GloVe technique. GloVe, an integral component of word embedding, employs a statistical

model to learn vector representations of words. It utilizes a co-occurrence matrix to calculate the frequency of words appearing together in the same context, thereby capturing the semantic relationships between them. This approach offers a robust methodology for representing words as numerical vectors while preserving their contextual meaning.

C. K-Fold Cross Validation

In this research, the training process employs the k-fold cross-validation technique, dividing the dataset into 10 equal subsets or folds. Each fold is used as a test set exactly once, while the remaining nine folds serve as the training set. This rotation ensures that every subset of the data is utilized for both training and testing, reducing the risk of bias and variance in performance evaluation. The process is repeated 10 times, one for each fold, providing a comprehensive assessment of the model’s behavior across all subsets. The results of the k-fold cross-validation are aggregated to calculate the cross-validation accuracy, which represents the average performance of the model on all test folds. This metric offers a robust estimate of how well the model is likely to perform on unseen data, effectively gauging its generalization ability. By systematically testing on all parts of the data, k-fold cross-validation minimizes overfitting and provides a more reliable evaluation compared to a single train-test split, ensuring the model’s reliability in real-world applications.

The figure 3 illustrates the performance of the KNN model based on the number of neighbors (k), using cross-validation accuracy as the metric. Accuracy is low for smaller k values (k = 1, 2), indicating overfitting as the model becomes overly sensitive to the training data. Accuracy improves significantly from k = 3 to k = 5 and stabilizes around 0.91 for k ≥ 6. The optimal value of k, identified through this analysis, is k = 9, which attains an accuracy of approximately 0.915, exhibiting a balanced distribution of low bias and low variance. This observation underscores the significance of selecting an optimal k value to circumvent the pitfalls of overfitting at smaller values and underfitting at larger values, thereby ensuring more reliable and consistent predictions.

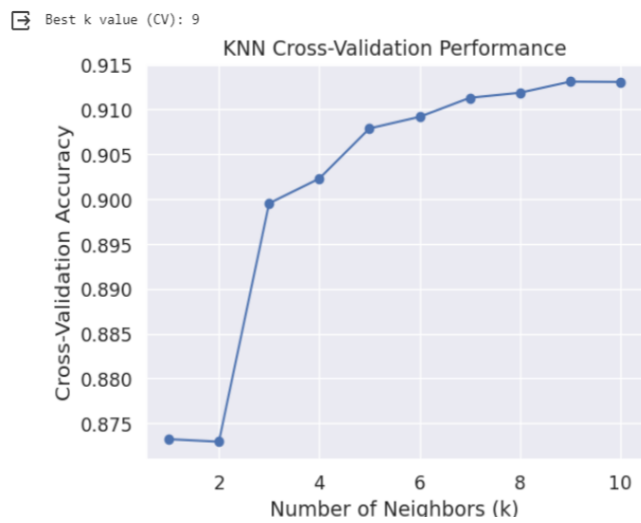


Fig. 3. Results of K-Fold Validation

D. K-Nearest Neighbors (KNN)

The results in figure 4 presented in the classification report demonstrate the efficacy of the KNN model for this dataset. The high precision, recall, and F1-scores across all labels indicate the model's capacity to accurately and consistently identify each category. The high overall accuracy further underscores the model's reliability and robustness in handling the dataset. The balanced performance across all evaluation metrics highlights the model's ability to minimize both false positives and false negatives. When combined with an optimal choice of k, as determined through cross-validation, this analysis underscores that the KNN model is not only suitable but also highly efficient for this classification task.

The classification report highlights that the KNN model performs well overall, with an accuracy of 88% and strong precision, recall, and F1-scores, particularly for class 0, which benefits from its large representation (support = 11,375). Performance for class 1 is reliable, with an F1-score of 0.89, but class 2 shows weaker performance (F1-score = 0.80), indicating challenges such as class imbalance or overlapping features. The macro averages (0.87 for all metrics) show balanced performance across classes, while the weighted averages (0.88–0.89) reflect the dominance of larger classes. To improve performance, especially for class 2, strategies like addressing class imbalance (e.g., oversampling or class weighting), refining features, experimenting with alternative distance metrics, and fine-tuning hyperparameters could be beneficial.

Classification Report (Best K-NN):				
	precision	recall	f1-score	support
0	0.94	0.92	0.93	11375
1	0.88	0.89	0.89	6609
2	0.80	0.81	0.80	6420
accuracy			0.88	24404
macro avg	0.87	0.87	0.87	24404
weighted avg	0.89	0.88	0.88	24404

Fig. 4. Accuracy of KNN Model

E. Evaluation Model

The confusion matrix in figure 5 provides a detailed breakdown of the model’s performance for sentiment classification, highlighting the number of correct (main diagonal) and incorrect (off-diagonal) predictions across all classes. With an impressive overall accuracy of 94.6%, the model demonstrates a strong ability to accurately predict sentiment in the majority of cases. However, examining the off-diagonal elements is crucial to identify specific misclassification patterns, such as which classes are most frequently confused. This could indicate potential challenges like overlapping features or class imbalance. To gain a deeper understanding of the model’s reliability, it is essential to evaluate class-specific metrics like precision, recall, and F1-score, which would provide a clearer picture of its effectiveness across different sentiment categories. Addressing any observed misclassifications through techniques such as feature refinement, rebalancing the dataset, or fine-tuning hyperparameters could further enhance the model’s performance.

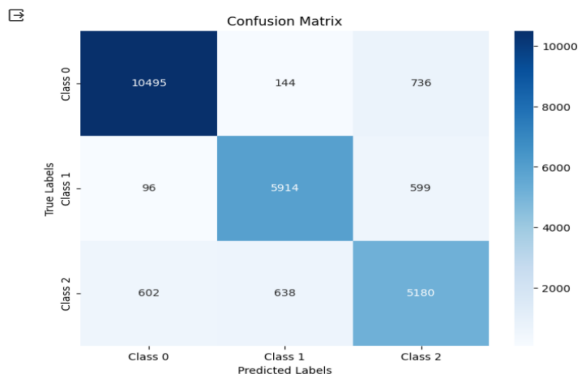


Fig. 5. Model Evaluation with Confusion Matrix

The confusion matrix demonstrates that the model performs strongly overall, achieving high true positive counts for Class 0 (10,495) and Class 1 (5,914), but struggles with Class 2, which has 5,180 correct classifications alongside notable misclassifications into Class 0 (602) and Class 1 (638). This pattern suggests potential overlap in features between Class 2 and the other classes or an imbalance in the dataset affecting its classification. With an impressive overall accuracy of 94.6%, the model effectively predicts most cases, but the misclassification rates for Class 2 highlight areas for improvement. Addressing these issues may involve refining features to better separate classes, mitigating class imbalance through resampling techniques, or leveraging alternative algorithms like ensemble methods. Evaluating class-specific precision, recall, and F1-scores would provide deeper insights, guiding efforts to enhance the model's overall robustness and reduce errors, particularly for the more challenging Class 2.

IV. CONCLUSION

The research on sentiment analysis of tweets discussing ChatGPT on Twitter, using the KNN algorithm with k-fold cross-validation, identified an optimal value of $k = 9$ for the analysis. With this configuration, the KNN algorithm achieved an impressive accuracy of 88%, showcasing its effectiveness in sentiment classification tasks with a relatively high level of reliability. These findings underscore the suitability of KNN for analyzing social media sentiments, particularly for technology-related topics like ChatGPT. To further enhance the model's performance and robustness, future research should consider leveraging a larger and more diverse dataset while also exploring advanced feature engineering techniques, alternative distance metrics, and other machine learning algorithms for comparison. Expanding the dataset would enable the model to capture a broader range of sentiment variations, improve its ability to generalize to new data, and provide deeper, more nuanced insights into public sentiment surrounding ChatGPT on social media platforms across different languages and regions.

REFERENCES

- [1] P. P. Ray, 'ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope', *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023. DOI: 10.1016/j.iotcps.2023.04.003
- [2] R. Santhosh, M. Abinaya, V. Anusuya, and D. Gowthami, 'ChatGPT: Opportunities, Features and Future Prospects', in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2023, pp. 1614–1622. DOI: 10.1109/ICOEI56765.2023.10125747
- [3] K. Hötte, M. Somers, and A. Theodorakopoulos, 'Technology and jobs: A systematic literature review', *Technological Forecasting and Social Change*, vol. 194, p. 122750, Sep. 2023. DOI: 10.1016/j.techfore.2023.122750
- [4] L. Babashahi *et al.*, 'AI in the Workplace: A Systematic Review of Skill Transformation in the Industry', *Administrative Sciences*, vol. 14, no. 6, p. 127, Jun. 2024. DOI: 10.3390/admsci14060127
- [5] Y. K. Dwivedi *et al.*, 'Opinion Paper: "So what if ChatGPT wrote it?" Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy', *International Journal of Information Management*, vol. 71, p. 102642, Aug. 2023. DOI: 10.1016/j.ijinfomgt.2023.102642
- [6] P. P. Ray, 'ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope', *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023. DOI: 10.1016/j.iotcps.2023.04.003
- [7] U. A. Bukar, M. S. Sayeed, S. F. A. Razak, S. Yogarayan, O. A. Amodu, and R. A. Raja Mahmood, 'Text Analysis on Early Reactions to ChatGPT as a Tool for Academic Progress or Exploitation', *SN Computer Science*, vol. 5, no. 4, p. 366, Mar. 2024. DOI: 10.1007/s42979-024-02714-7
- [8] Y. Xing, J. Z. Zhang, G. Teng, and X. Zhou, 'Voices in the digital storm: Unraveling online polarization with ChatGPT', *Technology in Society*, vol. 77, p. 102534, Jun. 2024. DOI: 10.1016/j.techsoc.2024.102534
- [9] S. J. Pipin, F. M. Sinaga, S. Winardi, and M. N. Hakim, 'Sentiment Analysis Classification of ChatGPT on Twitter Big Data in Indonesia Using Fast R-CNN', *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 7, no. 4, p. 2137, Oct. 2023. DOI: 10.30865/mib.v7i4.6816
- [10] M. Bordoloi and S. K. Biswas, 'Sentiment analysis: A survey on design framework, applications and future scopes', *Artificial Intelligence Review*, vol. 56, no. 11, pp. 12505–12560, Nov. 2023. DOI: 10.1007/s10462-023-10442-2
- [11] M. Wankhade, A. C. S. Rao, and C. Kulkarni, 'A survey on sentiment analysis methods, applications, and challenges', *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, Oct. 2022. DOI: 10.1007/s10462-022-10144-1
- [12] Z. R. Tembusai, H. Mawengkang, and M. Zarlis, 'K-Nearest Neighbor with K-Fold Cross Validation and Analytic Hierarchy Process on Data Classification', *International Journal of Advances in Data and Information Systems*, vol. 2, no. 1, Jan. 2021. DOI: 10.25008/ijadis.v2i1.1204
- [13] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, 'Efficient kNN Classification With Different Numbers of Nearest Neighbors', *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, May 2018. DOI: 10.1109/TNNLS.2017.2673241
- [14] S. C. Gupta and N. Goel, 'Performance enhancement of diabetes prediction by finding optimum K for KNN classifier with feature selection method', in *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 980–986. DOI: 10.1109/ICSSIT48917.2020.9214129

- [15] H. Rhomadhona and J. Permadi, 'Klasifikasi Berita Kriminal Menggunakan Naïve Bayes Classifier (NBC) dengan Pengujian K-Fold Cross Validation', *Jurnal Sains dan Informatika*, vol. 5, no. 2, pp. 108–117, Dec. 2019. DOI: 10.34128/jsi.v5i2.177
- [16] Dr. D. Mabuni and Dr. S. A. Babu, 'High Accurate and a Variant of k-fold Cross Validation Technique for Predicting the Decision Tree Classifier Accuracy', *International Journal of Innovative Technology and Exploring Engineering*, vol. 10, no. 3, pp. 105–110, Jan. 2021. DOI: 10.35940/ijitee.C8403.0110321
- [17] A. M. Pravina, K. N. Sani, H. D. Harfianto, T. A. Pratama, A. Fahrina, and Y. Ruldeviyani, 'Sentiment Analysis Of Delivery Service Opinions On Twitter Documents Using K-Nearest Neighbor Arsyah, et., al [Sentiment Analysis Of Delivery Service Opinions On Twitter Documents Using K-Nearest Neighbor]', *Hal; Jl. Salemba Raya*, vol. 9, no. 2, pp. 996–1012, 2022 [Online]. Available: <http://jurnal.mdp.ac.id>
- [18] N. Agustina, D. H. Citra, W. Purnama, C. Nisa, and A. R. Kurnia, 'Implementasi Algoritma Naive Bayes untuk Analisis Sentimen Ulasan Shopee pada Google Play Store', *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 2, no. 1, pp. 47–54, Apr. 2022. DOI: 10.57152/malcom.v2i1.195



Azliza Yacob

She is currently a senior lecturer and a researcher at University College TATI (UCTATI). She received her Bachelor of Science (Computer) and Master of Science (Information Technology – Manufacturing) at University Teknologi Malaysia (UTM). She also has completed her PhD in Computer Science, from University Malaysia Terengganu. Her research interest includes e-learning, quality control, decision support system, computer industry, cyber security and artificial intelligence. She has published numerous research paper including national and international journal papers and proceeding. She also has awarded with best paper award and many more awards in various innovations competitions and research grant.



Nurzal Effiyana Ghazali

She is a Senior Lecturer at Universiti Teknologi Malaysia (UTM) and Director of the Centre for Engineering Education (CEE). A scholar in engineering and education, she holds a Ph.D. from UTM and has authored over 30 publications, securing more than 20 research grants in areas such as IoT, mobility management, and engineering education. She is a certified Design Thinking instructor and has collaborated internationally with institutions in Japan, Indonesia, and Brunei. Dr. Nurzal has received numerous awards for her contributions, including the IEEE Humanitarian Technologies Board Award, UTM Excellence Service Award, and the Sakura Science Plan award, reflecting her dedication to advancing engineering education and innovative teaching practices globally.



Faez M. Hassan

He is currently an associated professor at physics department, college of education, Mustansiriyah University in Baghdad, Iraq. He received his Doctor of Philosophy in physics / Digital Image processing from Universiti Sains Malaysia, Malaysia. His research interests in digital image processing and Remote Sensing, in addition land cover / use Mapping using Unmanned Aerial Vehicles (UAV) Imaging.