Classifying Public Complaints in Denpasar: a Comparative Study of CNN, RNN, LSTM, and Stacking Deep Learning Models

I Komang Dharmendra*[1], I Made Pasek Pradnyana Wijaya[2], I Made Agus Wirahadi Putra[3], Yohanes Priyo Atmojo[4], Luh Putu Safitri Pratiwi[5]

[1,2,4]Information System, Institute of Technology and Business STIKOM Bali, Indonesia

[3]Affiliation Informatics Management, Institute of Technology and Business STIKOM Bali, Indonesia

[5]Information Technology, Institute of Technology and Business STIKOM Bali, Indonesia

Email: [1]dharmendra@stikom-bali.ac.id

Abstract

The process of lodging complaints represents a complex behavioral construct, influenced by the interplay of emotional states, sociocultural factors, and situational contexts. It functions as a pivotal channel for citizens to express dissatisfaction regarding the quality of governmental services. This research aims to optimize public complaint management by leveraging deep learning-based text classification on citizen submissions collected from the Denpasar City Complaint Web Portal. The methodological approach integrates several neural network models, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) networks, further enhanced by a Stacking ensemble technique that amalgamates the strengths of each architecture. The dataset consists of 10,302 textual records, categorized into four semantic classes: Complaints, Suggestions, Inquiries, and Information. To improve the robustness and reliability of the classification, advanced preprocessing steps were implemented, including the application of the Synthetic Minority Over-sampling Technique (SMOTE) to alleviate class imbalance and the utilization of Term Frequency–Inverse Document Frequency (TF-IDF) for extracting the most informative textual features. Empirical results demonstrate that the Stacking ensemble model significantly outperforms individual baseline models, achieving an accuracy of 77.83%, with recall and F1-score values of 74.38%. These findings highlight the effectiveness of ensemble deep learning approaches in multiclass complaint classification, thereby supporting improvements in public service delivery and fostering greater governmental transparency. Ultimately, this study contributes to the field of automated text classification by illustrating the potential of advanced neural architectures to enhance citizen participation and institutional accountability.

Keywords : *Complaint Classification, Deep Learning, Ensemble Learning, Public Complaints, Text Mining*

1. INTRODUCTION

Complaint behavior represents a multifaceted psychological and social phenomenon that fulfills diverse roles across individual, interpersonal, and organizational settings. Its primary function pertains to the regulation and expression of emotional tension [1], problem-solving [2], seeking empathy [3], social bonding [4], and challenging unfairness or power structures [5]. The way complaints are expressed and received is shaped by cultural, relational, and psychological factors [6].

The expression of public dissatisfaction is instrumental in assessing and advancing the standards of services provided to the public [7]. With the rise of digital platforms, governments increasingly use electronic systems to collect, manage, and respond to citizen complaints, aiming for greater transparency, responsiveness, and service quality [8]. Effective complaint management is now seen as a key indicator of good governance and public trust [9]. When complaints are handled efficiently, it leads to higher citizen satisfaction [9], improved trust, and better service delivery. Well-designed

complaint systems and responsive processes are essential for enhancing public service quality and maintaining public trust [10], [11].

Managing public complaints is a critical aspect of public service delivery and governance. Effective complaint management systems not only improve service quality but also enhance accountability, citizen satisfaction, and organizational efficiency [9], [12]. Therefore, efficiently grouping and categorizing public complaints is paramount for improving the responsiveness and effectiveness of public service delivery [13]. Without proper classification, organizations struggle to accurately allocate resources, promptly address issues, and ultimately enhance overall service quality.

Grouping public complaints is a critical process for improving the efficiency and effectiveness of public service responses [14], [12]. By categorizing complaints, organizations can better allocate resources, address issues promptly, and enhance overall service quality [15], [16]. In this study, complaints submitted by the public were classified using several deep learning algorithms as well as combinations of these algorithms. The proposed framework is designed to empirically validate the effectiveness of deep learning methodologies in facilitating precise classification of public complaints.

Several studies that served as references for the development of this research include research from V. Vinayak et al. at 2023 regarding "Consumer Complaints Classification using Deep Learning & Word Embedding Models" [17]. By integrating advanced deep learning architectures—such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (Bi-LSTM), and one-dimensional Convolutional Neural Networks (1D-CNN)—together with sophisticated word embedding techniques including Word2Vec, FastText, BERT, and DistilBERT, consumer complaint data can be systematically classified into six distinct and well-structured category groups. Among these architectures, the DistilBERT and CNN models demonstrate superior performance in consumer complaint classification, attaining an exceptional F-score of 93%. Research from Sujata Khedkar et al. in 2020 entitled "Deep Learning and Ensemble Approach for Praise or Complaint Classification" [18]. The comparison between Machine Learning, Deep Learning, and Ensemble Learning methods for classifying praise or complaints leverages hybrid features based on the linguistic characteristics of extreme opinions. The proposed Ensemble classifier integrates SVC, Random Forest, Decision Tree, and MLP classifiers. Evaluations were conducted on hotel domain reviews, measuring performance through Accuracy, Precision, Recall, and F1-score. Both the proposed Ensemble classifier and Deep Learning models demonstrated superior classification accuracy for praise or complaint classification in online reviews when compared to existing methods. Then research from Shinta Devi Lukitasari et al. in 2020 entitled "Deep Learning-Based Complaint Classification for Indonesia Telecommunication Company's Call Center" [19]. Using RNN algorithms, it was successfully demonstrated that a deep learning framework can be employed to develop an automated complaint classification system. The training process of the classification model yielded promising results, with LSTM and GRU models reaching accuracies of up to 86.61% after 10 epochs. Research from M. A. Bashar et al. in 2020 entitled "Regularizing LSTM classifier by transfer learning for detecting misogynistic tweets with small training set" [20]. Despite restrictions imposed by a limited training corpus, a pretrained LSTM language model (LM) can function as a regularization mechanism for an LSTM classifier, facilitating the development of an accurate and generalizable classification model. Within a Bayesian theoretical framework, transfer learning (TL) may be interpreted as a principled approach to regularizing uncertainty, thereby improving model reliability in probabilistic reasoning contexts. When just a small training dataset is available, the study demonstrates that a language model pre-trained on datasets spanning general to task-specific domains can successfully regularize an LSTM classifier. This method allows the LSTM model to surpass the most advanced classification models, according to empirical study using two small Twitter datasets. Consumer complaints are presented as text and categorized into six different types using deep

learning models (LSTM, GRU,Bi-LSTM, and 1D CNN) and word embedding methods (Word2Vec, FastText, BERT, and DistilBERT).

For effective complaint classification, it is recommended to combine ensemble learning and deep learning algorithms [21], [22], capitalizing on the strengths of both approaches to improve robustness and accuracy [23], [24]. For instance, it has been demonstrated that combining ensemble classifiers like SVC, Random Forest, Decision Tree, and MLP with deep neural networks like CNN, Dense Neural Networks, and Multichannel CNN improves important performance metrics like accuracy, precision, recall, and F1-score when classifying reviews as either positive or negative [18]. Moreover, ensemble-based strategies, including Stacking, Augmenting, and Bagging, have consistently demonstrated their efficacy in augmenting the performance of deep learning models. Such enhancements have been especially prominent within high-precision domains such as medical image classification, where these methodologies have produced notable improvements in F1-score performance and classification reliability [25].

In response to the challenges previously outlined, this research employs a combination of Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) models, unified through a stacking ensemble strategy, to classify public complaints sourced from the Denpasar region. The central objective is to develop a deep learning–based classification system capable of identifying the polarity embedded within complaint sentences, thereby enabling a more nuanced understanding of public sentiment. These interpretive insights hold significant potential to inform legislative refinement and improve the responsiveness of public service institutions, offering valuable guidance for policymakers, researchers, and government entities alike.

This research also aims to assess and compare the effectiveness of various deep learning architectures in categorizing public complaint data by employing an ensemble learning strategy. The approach utilizes a stacked framework that combines Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) models into a unified predictive system. The primary contribution of this research lies in the comprehensive comparative assessment of these neural structures to determine the most efficient and robust model for the specified classification objective. The outcomes of this investigation are expected to serve as a foundational reference for future scholarly efforts aimed at advancing text classification methodologies, with particular emphasis on the Indonesian language context.

2. METHOD

2.1. Deep Learning

To support the efforts of government institutions, academic researchers, and practitioners in developing more responsive policies and services, this study aims to design a deep learning model architecture capable of detecting sentence-level polarity. By doing so, it is expected to generate more profound insights into the nature and sentiment of public complaints [26], [27], [28]. A paradigm shift has occurred within artificial intelligence, driven by the capacity of machines to perform functions historically demanding human cognitive abilities. Speech and picture recognition, natural language processing, and a variety of other challenging cognitive tasks are important application areas [29], [30].

Unlike conventional machine learning methods, deep learning represents a transformative advancement in text classification, providing significant improvements in both feature representation and predictive performance. One key strength of deep learning within this context is its capacity to forgo manual feature engineering, as it can automatically derive meaningful and hierarchical feature representations directly from unprocessed text data. This capability is realized through various neural architectures: Convolutional Neural Networks (CNNs) demonstrate exceptional proficiency in identifying local structural patterns, while Recurrent Neural Networks (RNNs) and Long Short-Term

Memory (LSTM) models effectively capture linguistic sequentiality by preserving temporal dependencies and extended contextual information. The inherent strength of deep learning frameworks in handling large-scale and complex datasets underpins their superior generalization ability and explains their prevailing success across numerous natural language processing tasks [31], [32], [33].

The performance of text classification systems can be further augmented through the strategic implementation of ensemble learning methods. Hybrid models that integrate CNNs and LSTMs, for example, take advantage of the strengths of both architectures: CNNs excel at extracting high-level features, while LSTMs capture long-term dependencies in text. Text classification has shown great success with this combination technique. Prior studies have demonstrated that hybrid model architectures tend to achieve superior performance compared to single-model approaches, offering improvements in both predictive accuracy and model robustness. For instance, the MCNN-LSTM model, which combines CNNs and LSTMs, has demonstrated superior performance in classifying multi-class text in imbalanced news datasets compared to traditional machine learning methods [34], [35]. Additionally, attention-based Bi-LSTM+CNN models, Several approaches have been introduced to enhance classification accuracy by directing the model's attention toward the most informative segments of the text [36]. These ensemble methods highlight the considerable promise of deep learning in improving text classification, offering solutions that are both more precise and resilient.

2.2. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a well-established class of deep learning techniques, renowned for their effectiveness across a range of machine learning applications, particularly in the domain of text classification [37]. Their architecture enables the effective extraction of local features and patterns through convolutional layers, which is particularly advantageous when processing sequential data such as textual inputs. Within the scope of text classification, CNNs are capable of learning and identifying important features, including n-gram structures, that contribute to accurate categorization of textual content.

Contemporary studies have introduced composite deep learning frameworks that fuse Convolutional Neural Networks (CNNs) with Recurrent Neural Networks (RNNs), thereby exploiting the complementary advantages inherent in each architecture. This integrative modeling strategy effectively combines CNNs' capacity for local feature abstraction with RNNs' aptitude for capturing temporal dependencies, resulting in significant performance gains in text classification tasks. [38]. Another approach involved optimizing CNNs using Genetic Algorithms to fine-tune parameter settings, which significantly improved classification accuracy for Arabic text [39]. Furthermore, innovative techniques such as term-based pooling and attention mechanisms have been integrated into CNN models to enhance their ability to focus on relevant words, leading to improved classification outcomes [40]. These improvements confirm CNNs' broad applicability and stability in text classification, proving reliable across diverse languages and datasets. Equation 1 is CNN theorem equation.

$$y^{i,j^k} = \sum m_0 \sum n_0 (x^{i+m,j+n} \cdot w^{m,n,k} + b^k) \quad (1)$$

Note:
- y^{i,j^k} = is the output feature map for the k-th filter at position (i,j)
- x = represents the input or feature map
- w = is the filter (or kernel)
- b^k = is the bias for the k-th filter
- m and n = are the dimensions of the filter.

2.3.　Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) represent a specialized class of neural network architectures engineered explicitly for processing sequential data, such as speech signals, textual inputs, and time-series information. In contrast to traditional feedforward networks, RNNs incorporate recurrent connections that form directed cycles, enabling the networks to retain contextual knowledge from preceding inputs. This ability to maintain temporal dependencies equips RNNs with a distinct advantage in domains where the order and contextual linkage among data points are critical. Specifically, in the realm of text classification, RNNs efficiently model sequences of words to uncover underlying syntactic structures and semantic associations. These sequential dependencies are essential for accurate comprehension of the nuanced meanings embedded within textual content [41]. For instance, a bi-directional RNN encodes each word by combining forward and backward hidden states, thus enhancing the representation of the text by considering both preceding and following contexts [42]. To address issues like vanishing gradients, contemporary neural architectures integrate specialized components such as Long Short-Term Memory (LSTM) units and Gated Recurrent Units (GRUs), which are designed to enhance gradient flow and preserve long-range contextual information. These enhancements significantly improve the model's capacity to capture and retain long-range dependencies in sequential data [38]. Hybrid neural configurations that integrate RNNs with architectures such as Convolutional Neural Networks (CNNs) or attention-based mechanisms have shown enhanced effectiveness in text classification tasks, as they capitalize on the complementary strengths of different modeling techniques [42], [43]. Attention mechanisms enhance classification performance by allowing the model to focus selectively on the most relevant portions of the input text. Consequently, RNN-based architectures and their derivatives have demonstrated strong effectiveness across various text classification tasks, including sentiment analysis and topic identification [44]. Equation 2 is RNNs theorem equation

$$h_t = f(h_{t-1} + x_t)y_t = g(h_t) \qquad (2)$$

Note:
- h_t　　　= the hidden state at time step t
- x_t　　　= the input at time step t
- f and g = activation functions

2.4.　Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) models, a refined form of recurrent neural networks (RNNs), are explicitly designed to capture and preserve long-distance dependencies within sequential data. This capability addresses a major limitation of standard RNNs, which often struggle to retain information over extended input sequences. The architectural innovation of Long Short-Term Memory (LSTM) networks lies in their integration of memory cells, which are designed to retain information over long sequences and thus resolve the vanishing gradient issue common to traditional RNNs. This structural advantage makes them particularly potent for applications in time series analysis and natural language processing (NLP). This enables LSTMs to learn and preserve important features from sequences, such as words in a sentence, which is essential for text classification tasks. In these tasks, LSTMs categorize unstructured text into predefined classes by learning the contextual relationships between words. For instance, several research have shown that merging LSTMs with Convolutional Neural Networks (CNNs) can enhance feature extraction and increase classification accuracy [35], [36]. Hybrid models, such as Bi-LSTMs with attention mechanisms, further enhance the classification process by selectively focusing on the most critical parts of the text[36]. In addition, performance enhancements have been reported when LSTM architectures are combined with optimization strategies such as genetic algorithms or augmented with graph-based attention mechanisms. Collectively, LSTMs remain a crucial component of contemporary

text classification frameworks, delivering robust capabilities for handling complex and large-volume textual datasets [45]. The ability of Long Short-Term Memory (LSTM) networks to address the vanishing gradient issue arises from the presence of four essential gating mechanisms within each memory cell. These gates act as control units, modulating the flow of information by determining what should be retained or discarded from the cell state during each sequential update.

Forget gate:

Within a Long Short-Term Memory (LSTM) unit, the forget gate serves as a fundamental mechanism that governs the selective retention or removal of information from the preceding cell state. This regulatory function enables LSTM architectures to efficiently manage long-term dependencies by systematically filtering out obsolete or non-essential data from their internal memory, thereby preserving information essential for sequential learning tasks. Equation 3 is Forget Gate LSTM.

$$ft = \sigma\big(Wf \cdot [h_{t-1}, x_t] + b_f\big) \qquad (3)$$

Note:
- ft = Output of the forget gate at time step t; decides what to forget
- σ = Sigmoid activation function, outputs values between 0 and 1
- Wf = Weight matrix for the forget gate
- $[h_{t-1}, x_t]$ = Concatenation of the previous hidden state and current input
- b_f = Bias term for the forget gate

Input gate:

At a given time step, the input gate governs the incorporation of new information into the cell state. By utilizing both the present input and the preceding hidden state, it refines the cell's internal memory structure in conjunction with a candidate memory, thereby facilitating the integration of relevant data into the network's long-term representation. Equation 4 is Input Gate LSTM.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (4)$$

Note:
- i_t = Output of the input gate (which parts of the new information to use)
- σ = Sigmoid activation function
- W_i = Weight matrices for the input gate
- $[h_{t-1}, x_t]$ = Concatenation of previous hidden state and current input
- b_i = Bias terms

Candidate Memory Cell:

The candidate memory cell, denoted as $\tilde{C}_t$, represents the new content that could be added to the LSTM's internal memory (cell state). It captures potential new information derived from the current input x_t and the previous hidden state h_{t-1}, but only parts of it are added. Equation 5 is Candidate Memory Cell LSTM.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \qquad (5)$$

Note:
- $\tilde{C}_t$ = Candidate cell state (new memory content)
- tanh = Hyperbolic tangent activation function (values between -1 and 1)

- W_C = Weight matrix for generating candidate memory
- $[h_{t-1}, x_t]$ = Concatenation of previous hidden state and current input
- b_C = Bias term

Memory Cell Update:

The memory cell update is the core step in an LSTM (Long Short-Term Memory) where the internal cell state C_t is updated based on what the LSTM remembers from the past (controlled by the forget gate), What new information to add (controlled by the input gate and candidate memory). This process allows the LSTM to retain long-term dependencies and dynamically update its memory over time. Equation 6 is Memory Cell Update LSTM.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \qquad (6)$$

Note:
- C_t = Updated cell state at time step t
- f_t = Forget gate output — what to forget
- C_{t-1} = Previous cell state (long-term memory)
- i_t = Input gate output — what new info to add
- $\tilde{C}_t$ = Candidate memory — potential new memory content

Output Gate:

The output gate controls what information from the cell state C_t should be sent as the hidden state h_t. This hidden state is also passed to the next time step and possibly used for downstream tasks (like classification, prediction, etc.). The output gate is the final decision-maker: it decides what the LSTM reveals from its internal memory to the outside world. Equation 7 is Output gate activation LSTM and Equation 8 is Hidden State LSTM.

Output Gate Activation:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad (7)$$

Hidden State (Output):
$$h_t = o_t \cdot \tanh(C_t) \qquad (8)$$
Note:
- o_t = Output gate vector (what to output)
- h_t = Hidden state at time t; the LSTM's output
- C_t = Current cell state (internal memory)
- $\tanh(C_t)$ = Scales memory values to range [-1, 1]
- $[h_{t-1}, x_t]$ = Concatenation of previous hidden state and current input
- W_o, b_o = Weight matrix and bias for the output gate
- σ = Sigmoid activation (outputs between 0 and 1

2.5. Stacked Model

Stacked generalization, or the stacked model, is an advanced ensemble learning approach that seeks to enhance prediction capabilities by amalgamating multiple distinct base models. In this architectural scheme, individual base models are arranged in a hierarchical configuration, typically consisting of one or more tiers. The resulting outputs are subsequently forwarded to a higher-level metamodel, which integrates these responses to generate the final predictive outcome [46]. By exploiting

the advantages of diverse algorithms, this technique diminishes the shortcomings of singular models and elevates aggregate accuracy. Typically, base models are trained via cross-validation, with their predictions then serving as features for the metamodel. This metamodel is subsequently trained on a distinct validation set to avert overfitting.

The effectiveness of stacked models has been demonstrated across various domains, including intrusion detection in wireless networks and crime prediction. For instance, in intrusion detection, stacked ensemble learning has shown superior performance compared to individual algorithm, achieving higher accuracy in classifying attacks. Similarly, in crime prediction, a stacked ensemble approach using SVM algorithms has outperformed other machine learning models, achieving a classification accuracy of 99.5% [47]. These examples highlight the robustness and versatility of stacked models in handling complex and diverse datasets, making them a valuable tool in the machine learning toolkit.

Ensemble method operates by having each subsequent predictor analyze the errors of its predecessors, aiming to generate more accurate forecasts. Figure 1 illustrates a stacked ensemble learning technique that combines several base learners by feeding the same training data into each and merging their predictions to train a meta-model, such as logistic regression. This technique enhances prediction quality by leveraging the diverse strengths of each constituent model. It arranges multiple comparatively weak learners into a coordinated progression, in which every subsequent model is optimized to address the misclassifications produced by the one before it. Consequently, the ensemble evolves into a robust predictive system that substantially increases overall accuracy and generalization capability.
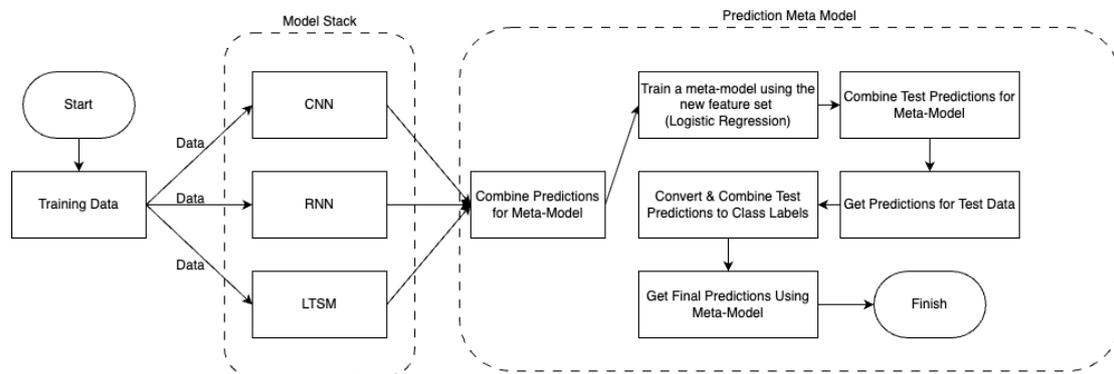

Figure 2. Stacked Model Flowchart

3. RESULT

3.1. Dataset

The dataset utilized in this research includes 10,302 public complaints submitted by citizens of Denpasar through the official Denpasar City Complaint Webpage (pengaduan.denpasarkota.go.id) [48], Figure 2 visually represents the interface of this webpage, serving as the primary source of our data. Data was systematically collected over a five-year period, spanning from February 2019 to February 2024. The complaints, originally unstructured text, were categorized into four distinct expression classes: 'Keluhan' (Complaint), 'Usul/Saran' (Suggestions), 'Pertanyaan' (Questions), and 'Informasi' (Information). As Table 1 indicates, there was a significant class imbalance within the collected data, with 'Keluhan' being the predominant category. To mitigate the skewed distribution among categories, the Synthetic Minority Over-sampling Technique (SMOTE) was employed to increase the presence of minority classes, thus promoting a more uniform data composition. Additionally, class-weight adjustments were introduced during model training to counterbalance the impact of class disparities, enabling improved generalization across diverse input instances. Subsequently, the dataset was

systematically partitioned into training and testing subsets following an 80:20 split, allocating approximately 80% of the records for model learning while reserving the remaining 20% to assess predictive performance on previously unseen instances.
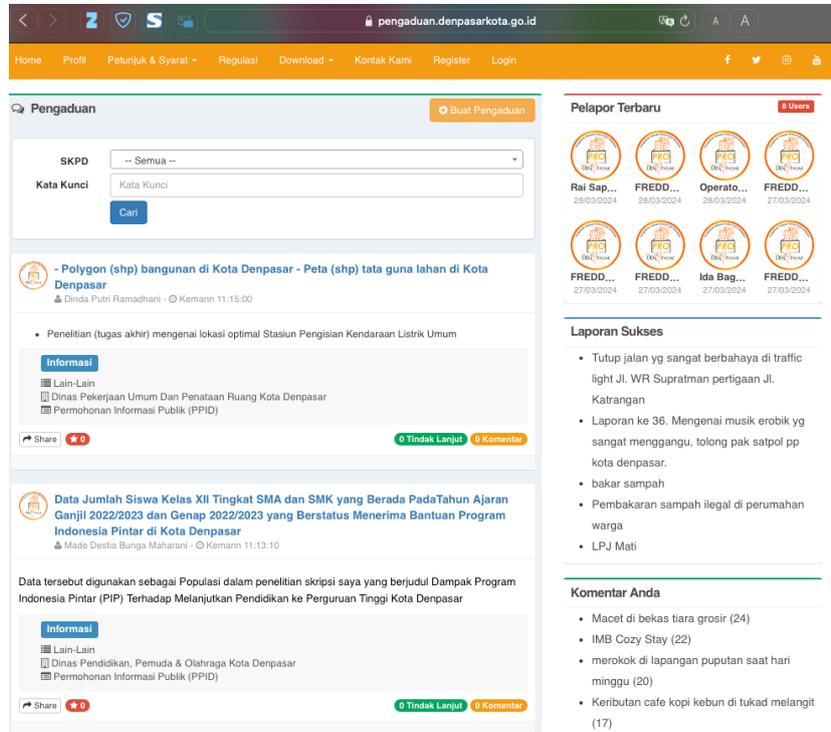

Figure 3. Denpasar City Report Webpage

There are four expression classes to be classified, namely Keluhan (Complaint), Usul/Saran (Suggestions), Pertanyaan (Questions), and Informasi (Information). Table 1 presents the distribution of these complaint classes, revealing a substantial imbalance. As shown in Table 1, 'Keluhan (Complaint)' is the most prevalent class with 8,256 data points, significantly outweighing 'Usul/Saran (Suggestions)' (794), 'Pertanyaan (Questions)' (700), and 'Informasi (Information)' (552). This substantial disparity in class sizes necessitated addressing the imbalanced dataset. To mitigate this, oversampling was performed using SMOTE to achieve a more balanced class distribution. Furthermore, to counteract potential biases introduced by oversampling and enhance the model's generalization capabilities, Class Weights were also employed during the training phase. This combined approach successfully mitigates class imbalance while maintaining the model's reliability and ensuring a fairer evaluative process across categories.

Table 1 Number of Data

Complain class	Number of Complaint Data
Keluhan (Complaint)	8256
Usul/Saran (Suggestions)	794
Pertanyaan (Questions)	700
Informasi (Information)	552

The dataset was partitioned into two subsets for experimental evaluation: a training set and a test set, following an 80:20 proportion. In this configuration, approximately 80% of the data is allocated for model learning, whereas the remaining 20% is designated to objectively assess the model's predictive performance.

3.2. Pre-processing

Before using any classification model, the dataset must be refined by data pre-processing, which is the next important step. This ensures that the data is properly prepared and optimized for classification methods. The overall flow of this research, including the pre-processing steps, is visually represented in Figure 3 (Research Design).

The pre-processing process begins by removing special characters, such as punctuation and symbols, to eliminate noise from the text. Additionally, all textual data were transformed to lowercase to ensure consistency and prevent the model from interpreting identical words as distinct solely based on letter casing. As an illustrative example, Table 2 displays a sample of the collected raw data. After these initial cleaning steps, a phrase like 'koman'kArXXXa- Harixxx 10:56:32' from the 'Description' field in Table 2 would be transformed into a cleaner, lowercase format, preparing it for further linguistic processing. These procedures contribute to producing a more standardized and streamlined dataset, thereby enhancing the overall effectiveness and reliability of the model's performance [49].

Table 2. Example of The Collected Data

Title	LPJIJ PDAM
Title URL	httos://pengaduan.denpasarkota.9.idt?page=DetaiI-Penzaduan&tan2uaZe=id&dQmain=&oenz_id=XXXXX
Image	https:f/pengaduan.denpasarkota.go.id/skins/renda/images flogo-prodenpasar.png
Description	koman'kArXXXa- Harixxx 10:56:32
boxbody	Hallo pemkot dps, untuk pengaduan lampu penerangan jalan mati kemana sy harus mengadu, mohon infonya terimakasih%l'. Lokasinya di jl tk badung XXIII renon denpasar Kondisinya mati kemudian hidup sekian detik mati lagi Sudati 3 butan lebih kondisinya seperti itu Semoga ce atditan ani
View_URL	https:l/pengaduan.denpasarkqta.go.idluypengaduan YeuxIW—WhatsA99%201mage%xxw-x-03-15%xx%xx.55.48%20AM.jpeg
Label	Informasi
Attachment ext	IPJ
Attachmentext1	Dinas Perhubungan Kota Denpasar
Attachmen ext2	Keluhan
badge	0
badge3	0 Tindak Lanjut
badge4	0 Komentar

Further, lemmatization is performed using the Sastrawi method [50], which is specifically tailored for the Indonesian language. After designing the model, the subsequent step is to train using the prepared dataset to achieve optimal performance. This training process focuses on optimizing the model's parameters and weights, thereby enhancing its accuracy and effectiveness in recognizing patterns and extracting meaningful features from the data. Through this process, the model learns to improve its predictions, allowing it to better classify or analyze new, unseen data based on the learned patterns. This technique reduces words to their root forms, ensuring that variations of a word are treated consistently. In addition, stop words, which are common words that add little informational value (e.g., "it", "the"), are eliminated to focus on the most relevant terms. In the case of Bahasa Indonesia, stop words such as "yang" and "di" are identified and removed. These pre-processing techniques collectively enhance the quality of the dataset, enabling more accurate and efficient modeling with classification algorithms.
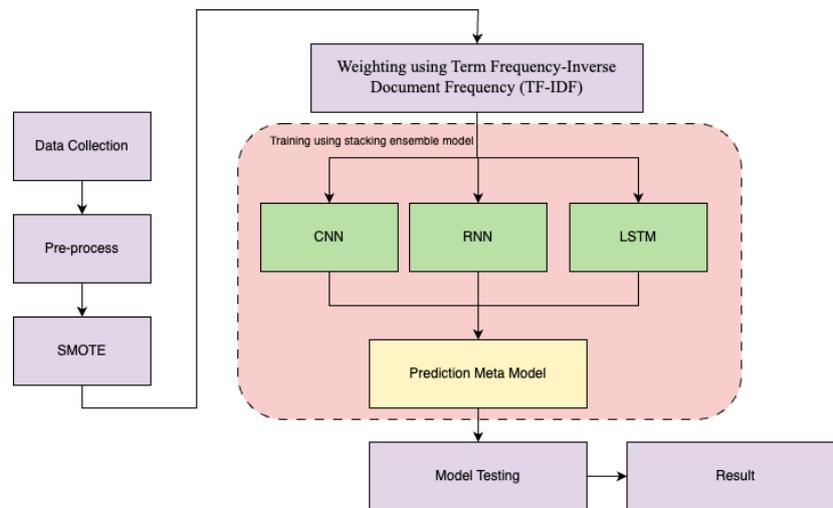
Figure 4. Research Design

After the pre-processing stage, the text data is transformed through Term Frequency–Inverse Document Frequency (TF-IDF) weighting. This method evaluates the significance of a term within a specific document relative to its occurrence across the entire collection of documents. A term receives a higher weight when it appears more frequently in a single document; however, its weight is reduced if the term is widespread throughout the corpus. This mechanism ensures that routinely occurring words do not exert an excessive impact on the final representation [51]. Highly beneficial in text classification, this technique was utilized in our research. Specifically, Term Frequency-Inverse Document Frequency (TF-IDF) weighting helped to ascertain the most indicative and important terms that delineate distinct complaint categories. This method allows for more effective text analysis by emphasizing important words in each document while reducing the influence of words that commonly appear across the entire corpus [52]. The transformed data, now represented by TF-IDF features, was then used as input for the various neural network models. Their configurations, carefully selected to optimize performance for text classification, are detailed in Table 3. This table outlines key parameters such as input length, vocabulary size, specific layers used in each architecture, and regularization techniques applied to prevent overfitting.

Table 3. Models parameters and configurations

Model	Input Length	Vocabulary Size	Layers	Regularization	Dropout	Epochs	Batch Size	Early Stopping
CNN	10	5000	Embedding, Conv1D, MaxPooling1D, GlobalMaxPooling1D, Dense, Dropout	L2 (0.01)	00.05	10	32	Yes (patience=3)
RNN	10	5000	Embedding, SimpleRNN, Dense	None	None	10	32	Yes (patience=3)
LSTM	10	5000	Embedding, LSTM, Dropout, Dense	L2 (0.01)	00.05	10	32	Yes (patience=3)

3.3.   Convolutional Neural Networks (CNN)

Figure 4 presents the training and validation outcomes achieved by the Convolutional Neural Network (CNN) model, which depict both loss and accuracy metrics over four epochs. The training loss demonstrates a consistent decline, indicating effective learning and optimization of the model parameters throughout the training process. In contrast, the validation loss exhibits a more fluctuating pattern, suggesting some variability in the model's ability to generalize to unseen data. As the model continues to refine its ability to classify the training data, a consistent increase in training accuracy is observed. Nevertheless, the validation accuracy remains comparatively modest, suggesting that although the model fits the training set effectively, it struggles to replicate that level of performance when evaluated on unseen validation data.
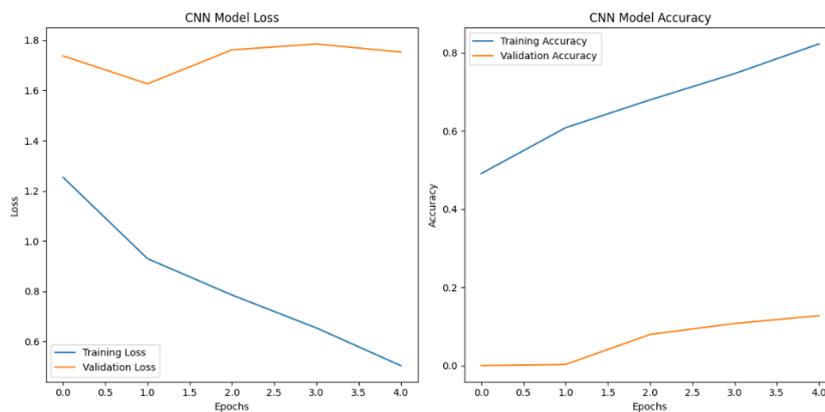

Figure 5. Training and Validation Loss and Accuracy for the CNN Model

This discrepancy between training and validation performance highlights the complexities involved in model training. The model's increasing ability to accurately categorize the training samples is reflected in the training accuracy, which is increasing steadily. The fluctuations in validation loss further emphasize the model's inconsistent performance on unseen data, which may be influenced by factors such as data distribution or the inherent difficulty of the classification task. Overall, these observations provide valuable insights into the model's learning dynamics and its current limitations in achieving robust performance across different datasets.

Quantitatively, based on the performance metrics presented in Table 4, the CNN model achieved an accuracy of 74.38%, Its precision, recall, and F1-Score values of 76.26%, 74.38%, and 72.55%, respectively. The high precision indicates that CNN is effective at minimizing false positives, making it strong in identifying non-relevant instances. However, the recall value of 74.38% suggests that the model still misses a notable portion of true positives. Despite this, the F1-Score, though slightly lower than precision, reflects a balanced performance between precision and recall. CNN thus provides reliable results, with a tendency towards classifying the complaints accurately, but with a minor gap in identifying all relevant cases.

3.4.   Recurrent Neural Networks (RNNs)

Figure 5 illustrates the training and validation performance of the Recurrent Neural Network (RNN) model, which illustrate both loss and accuracy metrics over seven epochs. the training loss exhibits a general downward trend, indicating that the model is effectively learning from the training data. However, the validation loss exhibits significant fluctuations, suggesting that the model's ability to generalize to unseen data is inconsistent. This variability in validation loss may indicate challenges in capturing the underlying patterns in the data.
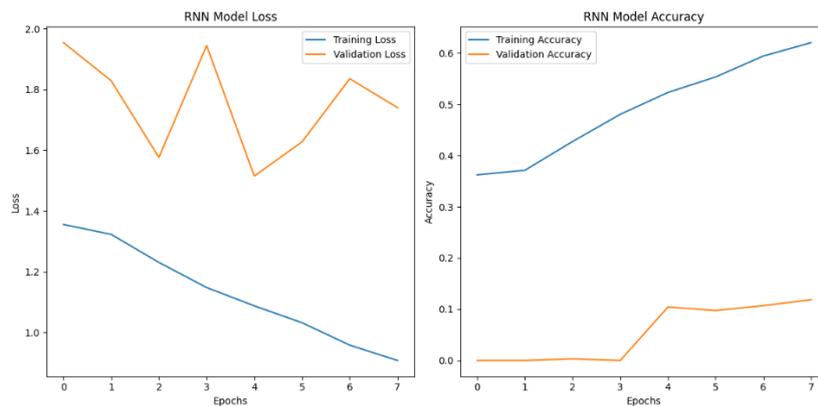
Figure 6. Training and Validation Accuracy and Loss for the RNN Model

Regarding accuracy, Figure 5 illustrates a steady rise in training accuracy, reflecting the model's growing proficiency in classifying the training dataset. In contrast, the validation accuracy remains relatively low and exhibits minimal improvement, suggesting that the model encounters difficulties in achieving comparable results on unseen data. This noticeable gap between training and validation performance suggests limitations in the model's ability to generalize effectively. Although the RNN demonstrates learning progress during training, it may lack the capacity to adequately handle the complexity inherent in the validation set.

From a quantitative standpoint, Table 4 indicates that the RNN model yielded the lowest performance across all evaluation metrics. It recorded an accuracy of 61.14%, a precision of 71.30%, a recall of 61.14%, and an F1-score of 65.27%. These figures suggest that the RNN was comparatively less effective in learning the underlying patterns present in the dataset. The modest accuracy and limited recall point to the model's difficulty in correctly identifying complaint instances, especially in capturing all relevant cases. Despite achieving a moderately satisfactory precision level, the notably low recall indicates that the model fails to identify a substantial portion of actual positive cases, ultimately diminishing its consistency and reliability across different class predictions. Overall, the RNN demonstrates considerably weaker performance compared to the other models assessed in this research.

3.5. Long Short-Term Memory (LSTM)

Figure 6 presents the training and validation performance of the Long Short-Term Memory (LSTM) model, showcasing both loss and accuracy metrics across seven training epochs. While the model demonstrates effective learning on the training set, the persistent complexity of the validation data appears to pose challenges for the model to generalize adequately. Notably, the validation loss displays noticeable fluctuations, indicating instability in the model's ability to adapt to previously unseen data. Such irregularities may reflect difficulties in consistently identifying the underlying structures within the dataset.

In terms of accuracy, the training accuracy shows a consistent upward trend, indicating the model's increasing capability in correctly classifying training instances. Nonetheless, the validation accuracy remains relatively low and exhibits only marginal gains, suggesting that the model encounters difficulties in replicating its training performance on unseen data. The observed variability in validation accuracy further underscores the model's unstable behavior during generalization. Collectively, these findings point to the inherent challenges in training the LSTM model and its current limitations in delivering reliable performance across diverse datasets.
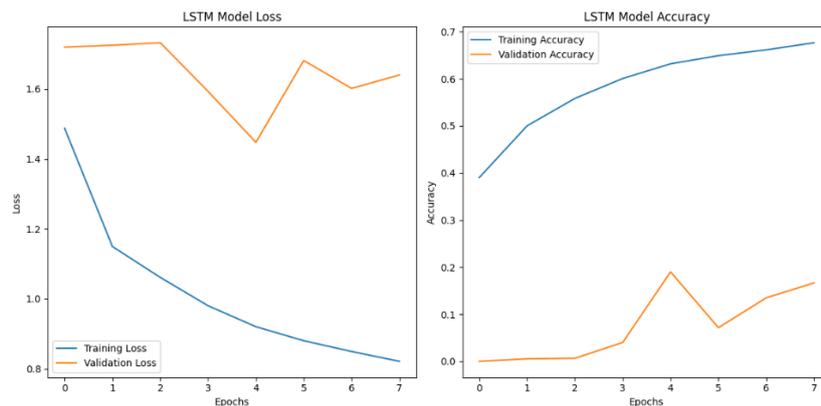
Figure 7. Training and Validation Loss and Accuracy for the LSTM Model

As reported in Table 4, the LSTM model achieved an accuracy of 69.92%, with a precision of 74.47%, a recall of 69.92%, and an F1-score of 71.48%. These outcomes reflect a relatively strong performance, placing the LSTM between CNN and RNN in terms of effectiveness. The precision value suggests that the model is proficient in minimizing false positives, while the recall indicates a solid capability in identifying true positive cases. The F1-score highlights a well-maintained balance between precision and recall, underscoring the model's overall reliability. Although LSTM effectively captures sequential dependencies and proves suitable for text classification tasks, its overall performance still falls short of that attained by CNN or the stacking ensemble approach.

3.6. Stacked Model

The stacking ensemble model is a sophisticated approach implemented in this study to enhance classification performance, as outlined in the "Stacking" phase of the Research Design shown in Figure 3. This model utilizes predictions from three base models: a Convolutional Neural Network (CNN), a Recurrent Neural Network (RNN), and a Long Short-Term Memory (LSTM) network.

The process begins by retrieving predictions for the training data from each base model and converting these into class labels using the 'np.argmax' function. These labels are then combined to form a new feature set, 'X_train_meta', which serves as input for a meta-model trained with Logistic Regression, utilizing the true labels of the training data as targets. After training, predictions are obtained for the test dataset, converted to class labels, and combined into 'X_test_meta'. The final predictions are generated by the meta-model, enhancing classification performance. The effectiveness of the stacked model is evaluated using the 'classification_report' function, providing metrics such as precision, recall, and F1-score for a comprehensive assessment of its predictive capabilities.

Table 4. Performance Metrics Comparison of CNN, RNN, LSTM, and Ensemble Learning Models

Model	Accuracy	Precision	Recall	F1-Score
CNN	74,38%	76,26%	74,38%	72,55%
RNN	61,14%	71,30%	61,14%	65,27%
LSTM	69,92%	74,47%	69,92%	71,48%
Stacking	77,83%	72,06%	77,83%	74,38%

As clearly shown in Table 4, the Stacking model exhibited the best overall performance across all metrics, It achieved a remarkable accuracy of 77.83%, precision of 72.06%, recall of 77.83%, and an F1-Score of 74.38%. The model's high accuracy indicates its strong overall ability to correctly classify instances, with a remarkable balance between identifying relevant and non-relevant complaints. The precision of 72.06% suggests that while the model is proficient at correctly classifying positive cases, it

also indicates a moderate occurrence of false positives, though it does not drastically affect the overall performance. This makes Stacking a reliable model for ensuring that the number of incorrectly classified non-relevant instances is kept to a reasonable level.

One of the key strengths of the Stacking model is its notably high recall of 77.83%, reflecting its strong ability to correctly identify the majority of relevant instances. The model exhibits effective learning behavior during training, as shown by the consistent decline in training loss, this outcome suggests that the parameter configuration has been effectively optimized. The achieved F1-score of 74.38% also indicates a well-maintained balance between precision and recall. Through the integration of several base learners, the stacking approach offsets the weaknesses of individual models, ultimately resulting in a more robust and adaptable classification system. This ensemble approach enhances both the detection of pertinent complaints and the overall classification accuracy, positioning the Stacking model as the most dependable and effective solution for the classification task at hand.

4. DISCUSSIONS

This research advances prior work in text classification by employing deep learning architectures, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), along with a stacking-based ensemble model, to analyze a distinct dataset of public complaints originating from Denpasar City. Our primary objective was to compare the performance of these algorithms in classifying complaints into 'Complaint', 'Information', 'Suggestion' and 'Question' categories, identifying the most effective model based on accuracy, precision, recall, and F1-Score. Based on table 4, particularly the superior performance of the Stacking model (Accuracy: 77.83%, F1-Score: 74.38%), corroborate the growing consensus in the field that ensemble methods often outperform individual models by leveraging their diverse strengths. While CNN demonstrated strong individual performance (Accuracy: 74.38%), this was still surpassed by the combined intelligence of the stacking approach. Conversely, the RNN model showed the lowest performance (Accuracy: 61.14%), while LSTM, with its improved memory capabilities, performed better (Accuracy: 69.92%) than RNN.

Despite these promising results, this research encountered several inherent challenges, particularly concerning the nature of the Indonesian language and the dataset itself. The complexity of Bahasa Indonesia, characterized by its rich morphology and informal usage in public complaints, poses significant hurdles for text processing. In contrast to widely resourced languages like English, Indonesian still has limited availability of extensive linguistic tools and large-scale pre-trained models, which may constrain the efficiency of tokenization and lemmatization processes. The use of the Sastrawi method, while effective, may not fully capture all nuances of colloquial or abbreviated Indonesian common in public discourse. Furthermore, the dataset, though oversampled with SMOTE, inherently stemmed from real-world, often informal text, which can contain typos, slang, and mixed language, introducing layers of noise that even robust pre-processing steps cannot entirely eliminate.

The outcomes of this study hold significance beyond the specific context of Denpasar City's complaint system, providing wider contributions to the fields of informatics and natural language processing (NLP). The effective utilization of deep learning approaches, especially the stacking ensemble framework, establishes a strong foundation for automating the categorization of unstructured text data in public service contexts. Such an approach can substantially enhance the operational efficiency of government agencies and related organizations by optimizing the processing of large volumes of citizen feedback, ultimately enabling faster response actions and more strategic allocation of resources. Practical implementations might include the use of similar models in emergency response systems to prioritize incoming messages based on textual content, in customer support centers to direct inquiries appropriately, or in social media surveillance to promptly detect emerging public concerns.

Furthermore, this study contributes to the growing body of knowledge on deep learning applications for low-resource languages such as Indonesian. The methodology employed, including the pre-processing pipeline and the comparative analysis of deep learning architectures, can serve as a blueprint for similar initiatives in other local languages or dialects across Indonesia and Southeast Asia. The understanding obtained regarding the relative advantages and limitations of CNN, RNN, and LSTM for this particular category of textual data can inform future decisions related to model selection and architectural development. Ultimately, this study highlights the transformative role of artificial intelligence in enhancing public sector services and promoting more effective citizen engagement through the deployment of intelligent text classification solutions.

5. CONCLUSION

This study contributes to the continued development of text classification by employing deep learning architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) models, and a stacking ensemble approach to analyze a distinctive dataset of public complaints originating from Denpasar City. The main objective was to assess and compare the performance of these models in classifying inputs into four categories: 'Complaint', 'Information', 'Suggestion', and 'Question'. Evaluation was based on standard metrics, including accuracy, precision, recall, and F1-score. The results, particularly the strong performance of the Stacking model (Accuracy: 77.83 percent, F1-score: 74.38 percent), support the growing body of evidence suggesting that ensemble-based methods often outperform individual models by combining their respective strengths. While the CNN model achieved a solid individual result (Accuracy: 74.38 percent), it was still surpassed by the integrated capabilities of the Stacking ensemble. In contrast, the RNN model delivered the weakest performance (Accuracy: 61.14 percent), whereas the LSTM model, benefiting from improved memory retention, performed moderately better (Accuracy: 69.92 percent).

Beyond its immediate application to Denpasar City's complaint system, this research contributes significantly to the broader field of text mining technology. The methodology and findings can be generalized to diverse domains requiring automated text categorization, such as customer service support ticket routing, sentiment analysis of social media feeds, spam detection, or document classification in legal and medical fields. This demonstrates the potential for efficient data processing and improved operational efficiency across various industries. The study additionally delivers important insights into the implementation of deep learning for low-resource languages such as Indonesian, serving as a reference for comparable initiatives targeting other regional languages or dialects throughout Southeast Asia.

For future investigations, it is suggested to incorporate more sophisticated transformer-based language models, including BERT, GPT, or IndoBERT which is specifically designed for Bahasa Indonesia, as these models have demonstrated state-of-the-art performance across numerous natural language processing benchmarks. Furthermore, investigating the efficacy of these models on multilingual datasets would be a valuable direction, expanding the scope of applicability beyond a single language. Future studies could also consider incorporating multi-modal data, such such as images or voice recordings accompanying complaints, to enrich the analytical framework and potentially enhance classification accuracy and reliability.

ACKNOWLEDGEMENT

REFERENCES

[1]   'Complaining during closed training: its functions and consequences', *Pol. Psychol. Bull.*, Jul. 2023, doi: 10.24425/ppb.2020.133772.

[2]   G. Leo and J. Pate, 'Complainer's Dilemma', *J. Public Econ. Theory*, vol. 27, no. 1, p. e70011, Feb. 2025, doi: 10.1111/jpet.70011.

[3]   M. Mauchand and M. D. Pell, 'Listen to my feelings! How prosody and accent drive the empathic relevance of complaining speech', *Neuropsychologia*, vol. 175, p. 108356, Oct. 2022, doi: 10.1016/j.neuropsychologia.2022.108356.

[4]   K.-E. Yoon, *Complaining As a Sociocultural Activity: Examining How and Why in Korean Interaction*. Lexington Books, 2021. doi: 10.5771/9781793604712.

[5]   G. Kruks-Wisner, 'Great Expectations, Great Grievances: The Politics of Citizens' Complaints in India', *Comp. Polit.*, vol. 54, no. 1, pp. 27–49, Oct. 2021, doi: 10.5129/001041521X16106633880914.

[6]   N. Meiners, 'Cultural Influences and Complaint Behavior of Older Consumers in International Markets: A Comparative Study', *Int. J. Appl. Res. Bus. Manag.*, vol. 5, no. 2, Jul. 2024, doi: 10.51137/wrp.ijarbm.2024.nmct.45636.

[7]   Umi Rahmawati, Septiana Wulandari, Merita Auli, and Tatang Sudrajat, 'Public Service Complaints Through Electronic Communication Media in The Era Of Digital Democracy in South Sumatera Province: Political and Public Policy Analysis', *Public Serv. Gov. J.*, vol. 6, no. 1, pp. 104–126, Jan. 2025, doi: 10.56444/psgj.v6i1.2434.

[8]   M. Haekal and M. B. Z. Tjenreng, 'Analisis Pengelolaan Pengaduan Masyarakat melalui Aplikasi JAKI dalam Rangka Pelayanan Publik di DKI Jakarta', *J. PKM Manaj. Bisnis*, vol. 5, no. 1, pp. 228–241, Jan. 2025, doi: 10.37481/pkmb.v5i1.1245.

[9]   F. A. Alenezy and M. Akhlaq, 'Fix-It: Design and Implementation of a Public Complaint Management System', in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, Jakarta, Indonesia: IEEE, Feb. 2023, pp. 858–862. doi: 10.1109/ICCoSITE57641.2023.10127715.

[10]   R. Ali, A. S. Abdul Latiff, and S. Abdul Wahab, 'Revitalizing Public Trust through Efficient Complaint Management: A Comprehensive Study on the Impact and Effectiveness of Public Complaint Management System of Malaysia', presented at the Proceedings of Society 5.0 Conference 2023, pp. 9--4. doi: 10.29007/zdn7.

[11]   W. A. Setyarini, 'Survei Kepuasan Masyarakat terhadap Pelayanan Pengaduan Masyarakat Lapor Hendi Tahun 2021', *J. Riptek*, vol. 16, no. 2, pp. 90–96, Dec. 2022, doi: 10.35475/riptek.v16i2.157.

[12]   R. Kumalasari, 'STUDY EFEKTIVITAS KEBIJAKAN INOVASI DOKADU (DOKUMENTASI PENGADUAN) DI DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA MAGELANG', *J. Jendela Inov. Drh.*, vol. 6, no. 1, pp. 88–109, Mar. 2023, doi: 10.56354/jendelainovasi.v6i1.138.

[13]   F. A. Lestari, L. Efrizoni, E. Ali, and R. Rahmiati, 'Sistem Klasifikasi Pengaduan Masyarakat Pada BPJS Ketenagakerjaan Menggunakan Algoritma Naïve Bayes Berbasis Mobile', *Build. Inform. Technol. Sci. BITS*, vol. 4, no. 1, Jun. 2022, doi: 10.47065/bits.v4i1.1685.

[14]   J. Van Dael, T. W. Reader, A. Gillespie, A. L. Neves, A. Darzi, and E. K. Mayer, 'Learning from complaints in healthcare: a realist review of academic literature, policy evidence and front-line insights', *BMJ Qual. Saf.*, vol. 29, no. 8, pp. 684–695, Aug. 2020, doi: 10.1136/bmjqs-2019-009704.

[15]   R. Bewinda, Andi Fitri Rahmadany, and Mansyur Achmad, 'Community Complaint Services Through Electronic Complaint Channels in Increasing the Effectiveness of Government Affairs Compulsory Basic Services', *J. Humanit. Soc. Sci. Stud.*, vol. 6, no. 9, pp. 89–97, Sep. 2024, doi: 10.32996/jhsss.2024.6.9.11.

[16] D. Hertati, 'Improving the Quality of Public Services through Innovation of Online Complaints Based on the Wargaku Application', *J. Soc.*, vol. 4, no. 3, pp. 97–102, Oct. 2023, doi: 10.37899/journal-la-sociale.v4i3.831.

[17] V. Vinayak and J. C., 'Consumer Complaints Classification using Deep Learning & Word Embedding Models', in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Delhi, India: IEEE, Jul. 2023, pp. 1–5. doi: 10.1109/ICCCNT56998.2023.10307286.

[18] S. Khedkar and S. Shinde, 'Deep Learning and Ensemble Approach for Praise or Complaint Classification', *Procedia Comput. Sci.*, vol. 167, pp. 449–458, 2020, doi: 10.1016/j.procs.2020.03.254.

[19] S. Lukitasari and F. Hidayat, 'Deep Learning-Based Complaint Classification for Indonesia Telecommunication Company's Call Center', in *Proceedings of the Proceedings of the 7th Mathematics, Science, and Computer Science Education International Seminar, MSCEIS 2019, 12 October 2019, Bandung, West Java, Indonesia*, Bandung, Indonesia: EAI, 2020. doi: 10.4108/eai.12-10-2019.2296518.

[20] M. A. Bashar, R. Nayak, and N. Suzor, 'Regularising LSTM classifier by transfer learning for detecting misogynistic tweets with small training set', *Knowl. Inf. Syst.*, vol. 62, no. 10, pp. 4029–4054, Oct. 2020, doi: 10.1007/s10115-020-01481-0.

[21] A. Mohammed and R. Kora, 'A comprehensive review on ensemble deep learning: Opportunities and challenges', *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 2, pp. 757–774, Feb. 2023, doi: 10.1016/j.jksuci.2023.01.014.

[22] K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, 'Sentiment Analysis With Ensemble Hybrid Deep Learning Model', *IEEE Access*, vol. 10, pp. 103694–103704, 2022, doi: 10.1109/ACCESS.2022.3210182.

[23] S. Abimannan, E.-S. M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla, and D. Satheesh, 'Ensemble Multifeatured Deep Learning Models and Applications: A Survey', *IEEE Access*, vol. 11, pp. 107194–107217, 2023, doi: 10.1109/ACCESS.2023.3320042.

[24] A. Mohammed and R. Kora, 'An effective ensemble deep learning framework for text classification', *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8825–8837, Nov. 2022, doi: 10.1016/j.jksuci.2021.11.001.

[25] D. Muller, I. Soto-Rey, and F. Kramer, 'An Analysis on Ensemble Learning Optimized Medical Image Classification With Deep Convolutional Neural Networks', *IEEE Access*, vol. 10, pp. 66467–66480, 2022, doi: 10.1109/ACCESS.2022.3182399.

[26] R. K. Mishra, G. Y. S. Reddy, and H. Pathak, 'The Understanding of Deep Learning: A Comprehensive Review', *Math. Probl. Eng.*, vol. 2021, pp. 1–15, Apr. 2021, doi: 10.1155/2021/5548884.

[27] C. Janiesch, P. Zschech, and K. Heinrich, 'Machine learning and deep learning', *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[28] J. Egger, A. Pepe, C. Gsaxner, Y. Jin, J. Li, and R. Kern, 'Deep learning—a first meta-survey of selected reviews across scientific disciplines, their commonalities, challenges and research impact', *PeerJ Comput. Sci.*, vol. 7, p. e773, Nov. 2021, doi: 10.7717/peerj-cs.773.

[29] H. J. S and Mrs. M. Raju, 'A Study on Deep Learning', *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 10, no. 11, pp. 961–964, Nov. 2022, doi: 10.22214/ijraset.2022.47486.

[30] D. Berrar and W. Dubitzky, 'Deep learning in bioinformatics and biomedicine', *Brief. Bioinform.*, vol. 22, no. 2, pp. 1513–1514, Mar. 2021, doi: 10.1093/bib/bbab087.

[31] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, 'A comparative review on deep learning models for text classification', *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, p. 325, Jul. 2020, doi: 10.11591/ijeecs.v19.i1.pp325-335.

[32] Y. Zhou, 'A Review of Text Classification Based on Deep Learning', in *Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis*, Marseille France: ACM, Apr. 2020, pp. 132–136. doi: 10.1145/3397056.3397082.

[33] H. Wu, Y. Liu, and J. Wang, 'Review of Text Classification Methods on Deep Learning', *Comput. Mater. Contin.*, vol. 63, no. 3, pp. 1309–1321, 2020, doi: 10.32604/cmc.2020.010172.

[34] H. Zhou, 'Research of Text Classification Based on TF-IDF and CNN-LSTM', *J. Phys. Conf. Ser.*, vol. 2171, no. 1, p. 012021, Jan. 2022, doi: 10.1088/1742-6596/2171/1/012021.

[35] K. M. Hasib *et al.*, 'MCNN-LSTM: Combining CNN and LSTM to Classify Multi-Class Text in Imbalanced News Data', *IEEE Access*, vol. 11, pp. 93048–93063, 2023, doi: 10.1109/ACCESS.2023.3309697.

[36] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, 'Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism', *Appl. Sci.*, vol. 10, no. 17, p. 5841, Aug. 2020, doi: 10.3390/app10175841.

[37] L. Khan, A. Amjad, K. M. Afaq, and H.-T. Chang, 'Deep Sentiment Analysis Using CNN-LSTM Architecture of English and Roman Urdu Text Shared in Social Media', *Appl. Sci.*, vol. 12, no. 5, Art. no. 5, Jan. 2022, doi: 10.3390/app12052694.

[38] P. Sunagar and A. Kanavalli, 'A Hybrid RNN based Deep Learning Approach for Text Classification', *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130636.

[39] M. Yousef and A. ALali, 'Analysis and Evaluation of Two Feature Selection Algorithms in Improving the Performance of the Sentiment Analysis Model of Arabic Tweets', *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 6, 2022, doi: 10.14569/IJACSA.2022.0130683.

[40] H. Zhao, Z. Liu, X. Yao, and Q. Yang, 'A machine learning-based sentiment analysis of online product reviews with a novel term weighting and feature selection approach', *Inf. Process. Manag.*, vol. 58, no. 5, p. 102656, Sep. 2021, doi: 10.1016/j.ipm.2021.102656.

[41] Y. Karaman, F. Akdeniz, B. K. Savaş, and Y. Becerikli, 'A Comparative Analysis of SVM, LSTM and CNN-RNN Models for the BBC News Classification', in *Innovations in Smart Cities Applications Volume 6*, M. Ben Ahmed, A. A. Boudhir, D. Santos, R. Dionisio, and N. Benaya, Eds., Cham: Springer International Publishing, 2023, pp. 473–483. doi: 10.1007/978-3-031-26852-6_44.

[42] S. Lyu and J. Liu, 'Convolutional Recurrent Neural Networks for Text Classification':, *J. Database Manag.*, vol. 32, no. 4, pp. 65–82, Oct. 2021, doi: 10.4018/JDM.2021100105.

[43] D. Kim, 'Text Classification Based on Neural Network Fusion', *Teh. Glas.*, vol. 17, no. 3, pp. 359–366, Jul. 2023, doi: 10.31803/tg-20221228154330.

[44] E. A. Nismi Mol and M. B. Santosh Kumar, 'Study on Impact of RNN, CNN and HAN in Text Classification', in *2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*, Cochin, India: IEEE, Jul. 2020, pp. 94–102. doi: 10.1109/ACCTHPA49271.2020.9213231.

[45] H. Wang and F. Li, 'A text classification method based on LSTM and graph attention network', *Connect. Sci.*, vol. 34, no. 1, pp. 2466–2480, Dec. 2022, doi: 10.1080/09540091.2022.2128047.

[46] A. Chatzimparmpas, R. M. Martins, K. Kucher, and A. Kerren, 'StackGenVis: Alignment of Data, Algorithms, and Models for Stacking Ensemble Learning Using Performance Metrics', *IEEE Trans. Vis. Comput. Graph.*, vol. 27, no. 2, pp. 1547–1557, Feb. 2021, doi: 10.1109/TVCG.2020.3030352.

[47] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, 'An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach', *IEEE Access*, vol. 9, pp. 67488–67500, 2021, doi: 10.1109/ACCESS.2021.3075140.

[48] 'Pro Denpasar'. Accessed: Apr. 17, 2023. [Online]. Available: https://pengaduan.denpasarkota.go.id/

[49] I. K. Dharmendra, I. G. N. A. Kusuma, I. A. M. C. Dewi, and Edwar, 'IMPLEMENTASI TEXT MINING UNTUK KLASIFIKASI OPINI ALUMNI PADA PERGURUAN TINGGI', *J. Teknol. Inf. Dan Komput.*, vol. 9, no. 3, Art. no. 3, Jul. 2023, Accessed: Aug. 18, 2023. [Online]. Available: https://jurnal.undhirabali.ac.id/index.php/jutik/article/view/2504

[50] 'Stemming Bahasa Indonesia · sastrawi/sastrawi Wiki · GitHub'. Accessed: Jun. 26, 2019. [Online]. Available: https://github.com/sastrawi/sastrawi/wiki/Stemming-Bahasa-Indonesia

[51] M. Das, S. K., and P. J. A. Alphonse, 'A Comparative Study on TF-IDF feature Weighting Method and its Analysis using Unstructured Dataset', 2023, *arXiv*. doi: 10.48550/ARXIV.2308.04037.

[52]   Z. Jiang, B. Gao, Y. He, Y. Han, P. Doyle, and Q. Zhu, 'Text Classification Using Novel Term Weighting Scheme-Based Improved TF-IDF for Internet Media Reports', *Math. Probl. Eng.*, vol. 2021, pp. 1–30, Mar. 2021, doi: 10.1155/2021/6619088.