Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

Comparison of Convolutional Neural Network and Support Vector Machine for Student Question Classification in ChatGPT-based Learning Tools

Brilliant Jordan, Kemas Muslim Lhaksmana*

School of Computing, Telkom University, Bandung, Indonesia
Email: [1]jordanidq@student.telkomuniversity.ac.id, [2,*]kemasmuslim@telkomuniversity.ac.id
Correspondence Author Email: kemasmuslim@telkomuniversity.ac.id
Submitted: **01/07/2025**; Accepted: **01/09/2025**; Published**: 02/09/2025**

Abstract−Artificial Intelligence (AI) has revolutionized educational tools by enabling systems that proactively understand and respond to student needs. ChatGPT, a widely used generative model for education in Indonesia. However, it struggles to classify student questions accurately due to ambiguous phrasing, overlapping sentence structures, and difficulty recognizing intent, which limits its effectiveness as a learning assistant. This study compares the performance of Convolutional Neural Networks (CNN), which extract locally important features from word sequences with Support Vector Machines (SVM) in classifying student questions known for handling high-dimensional data and efficiently finding the optimal hyperplane for text classification. A dataset of 2,797 Indonesian ChatGPT interactions (71% clear vs. 29% unclear) was preprocessed through case folding, stop-word removal, stemming, and tokenisation, followed by data augmentation based on synonyms, which was applied to the minority class to balance the dataset. The models were tuned through grid or random search with prediction testing of the best model using 5-fold cross-validation comparisons across three data splits (70:30, 80:20, and 90:10). Results showed that CNN achieved balanced accuracy, precision, recall, and F1-score of 0.90 on the 90:10 split, outperforming SVM, which plateaued at 0.85 accuracy and dropped to 0.76 in F1-score. The embedded filters of the CNN found generality from lexical variation through the process of augmentation, while the TF-IDF sparse vectors in the SVM failed to maintain this level of semantics. These findings underscore that CNN is more adaptive to diverse data and better suited for integration into ChatGPT-based educational tools, particularly in supporting reliable classification and personalised AI feedback in student learning contexts.

Keywords: CNN; SVM; ChatGPT; Text Classification; Data Augmentation

1. INTRODUCTION

Artificial Intelligence (AI) has revolutionized educational tools by making it possible to create systems that proactively understand student needs and respond to them. ChatGPT is one of the most widely used generative models and has become widely popular for educational use in Indonesia [1][2]. While ChatGPT has tremendous potential, it cannot classify student questions accurately because of the ambiguous way they are phrased, the overlapping sentence structures, and the difficulty in recognizing intent even with educational questions [3]. This highlights the need for defined classification algorithms, such as CNN and SVM, to improve the effectiveness of learning tools that use ChatGPT.

Selecting a proper algorithm is important for building a question text classification model for ChatGPT. Since CNN can apply convolutions to extract locally important features from word sequences, it is recognized as an efficient model source for text classification. [4]. Some of the shortcomings of CNNs are their inability to model long-range dependencies between words, lack of explicit word order, and reliance on variables that can change quite a bit (kernel size and number of layers) [5]. Support Vector Machine (SVM) is another text classification algorithm that is well known for handling high-dimensional data and finding the optimal hyperplane for efficiently separating the classes [6]. SVM is often a dependable and stable algorithm across a variety of text classification situations because it also performs well with labeled data and uses kernel functions to loss classify non-linear data [6].

Besides selecting the algorithms themselves, supporting techniques like data augmentation, hyperparameter tuning, and appropriate data split ratios also affect model performance. Thanks to data augmentation techniques such as synonym replacement and back translation, improvements have been made with text classification models by augmenting the training data and diversifying the training and distributing classes evenly. This has been demonstrated to work well mainly with data in the Indonesian language [7]. To improve accuracy, efficiency, and overall generalizability and stability of model performance when applied to unseen data, hyperparameter tuning is a necessary step in machine learning and deep learning model development [8] In this study, Keras Tuner is used to tune the CNN, while GridSearch is used to tune the SVM. Since the composition of the training and testing data has an important influence on the classification model performance, the split ratio is important too. The best ratio depends on the characteristics of the dataset and the algorithm [9].

Numerous studies have centered on text classification using CNN and SVM algorithms. Sari et al. (2024) state that the combination of CNN-LSTM, GloVe and hyperparameter tuning significantly increased accuracy of text classification which indicates that it is relevant for query classification in the ChatGPT context [10]. Seno et al. (2024), for example, used the SVM algorithm to detect emotions using Indonesian text and demonstrated that the SVM model was viable for NLP and classification, with the Sigmoid kernel providing the best accuracy [11]. García-López et al. [12], through a systematic literature review, identified that ChatGPT often fails to interpret ambiguous or under-specified educational questions correctly, which can lead to misclassification or the generation of irrelevant

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

responses—particularly when student prompts lack explicit structure or intent. As pointed out by Rahma and Suadaa (2023), augmentation methods such as synonym replacement and back translation can significantly advance the performance of Indonesian text classification models, particularly when dealing with unbalanced data [13]. Aminudin and Wijayanti (2022) also explained that students use various questioning strategies when employing mathematical reasoning, and that hypothesizing questions are particularly good gauges of students' overall level of understanding and exploration [14]. The results indicate that, especially when communicating with ChatGPT, selecting the appropriate algorithms, applying data augmentation strategies, and understanding the characteristics of user queries are key components of a successful text classification system.

Previous research used CNN and SVM in a variety of text classification tasks, including emotion detection, general natural language processing (NLP), and even non-text areas, although we cannot find any study applying both models to directly compare them in the task of classifying student question types during interaction with ChatGPT. Furthermore, most studies also did not control for important performance factors such as hyperparameter search, data augmentation, or variations in training-test splits. Accordingly, there is an obvious gap in the research, and it is a timely opportunity to conduct a narrow, experimental comparison of CNN and SVM in the specific context of utilizing educational questions generated and/or provided to ChatGPT.

This study aims to analyze and compare the performance of *CNN* and *SVM* algorithms in classifying student question types in the context of *ChatGPT*. It also evaluates the effects of varying training–test data split ratios and hyperparameter tuning on model performance. Furthermore, the study examines the impact of data augmentation techniques on improving the accuracy and consistency of classification results.

2. RESEARCH METHODOLOGY

This study was carried out through several systematic stages, beginning with data acquisition, preprocessing, text augmentation, dataset splitting, implementation of CNN and SVM models, hyperparameter tuning, and evaluation and comparison of results. An overview of these steps is presented in Figure 1 below:

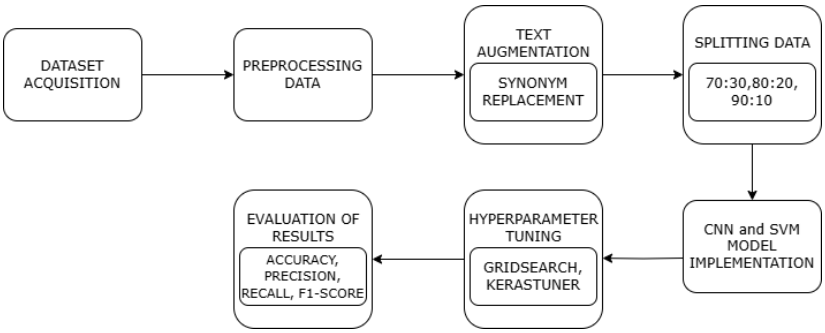

Figure 1. Research Flowchart

2.1 Dataset Acquisition

The dataset analyzed in this study came from 43 student interactions with ChatGPT from Telkom University. The dataset comprised 2,797 entries; each included a column number, the respective prompt(student's question), and the corresponding system reaction. All data was labeled by the researchers based on the question quality classification criteria, as derived from Bloom's Revised Taxonomy cognitive approach, as well as an additional indicator for the presence of 5W1H elements (What, Why, When, Where, Who, and How). While labeling the data, we used two labels: label 0 for questions that were clear and specific - included all 5W1H elements with at least the 'understanding' level. Label 1 was assigned to questions that were unclear, vague, or surface - typically at the 'remembering' level (C1) or lacked incomplete 5W1H elements [15]. Out of the total set of data, 1,990 (71.1%) were clearly question, and 805 (28.8%) were unclear questions. To enhance the consistency and reliability of the labeling process, we used three annotators to independently label the data, and Cohen's Kappa was used to measure inter-rater agreement. The resulting κ score was 0.80, indicating substantial agreement.

2.2 Preprocessing Data

Data preprocessing is a crucial step aimed at cleaning and standardizing raw data to ensure its optimal use in machine learning modeling [16]. The following is the process used in data preprocessing:

a. Case Folding
   Case folding is the process of converting all letters in a text to lowercase in order to standardize the representation of words that are identical but differ in capitalization [16].
b. Stopword Removal
   Stopword removal is a preprocessing step aimed at eliminating common words with low informational value, such as conjunctions and prepositions, to enhance the efficiency and accuracy of feature extraction in text modeling [16].

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

c. Stemming
Stemming is the process of removing inflections from derived words and reducing the form variation of words that have the same meaning for the sake of reducing the complexity of representation in text [16].

d. Tokenizing
Tokenization is a preprocessing procedure that breaks up a sentence or document into unit words (token) so that the system can run further analysis such as feature extraction and classification [16].

Table 1 provides concrete examples of data before and after each preprocessing step, illustrating the transformation from raw student questions to normalized input suitable for modeling. These steps collectively reduce noise, harmonize structure, and enhance semantic clarity in the training data.

Table 1. Example of Preprocessing data

Process	Before	After
Case Folding	Apa itu algoritma?	apa itu algoritma
Stopword Removal	bagaimana algoritma yang efisien digunakan	bagaimana algoritma efisien digunakan
Stemming	berikan contoh penerapan algoritma	beri contoh terap algoritma
Tokenizing	beri contoh terap algoritma	(beri), (contoh), (terap), (algoritma)

2.3 Text Augmentation

In this study, in order to obtain diversity in the training data, synonym augmentation was applied to account for class imbalance. Synonym augmentation is demonstrated to potentially be one way to improve model performance, particularly for text classification tasks in the Indonesian language [7]. This was done by randomly swapping out chosen words in the dataset with their synonyms using WordNet [17], which is a lexical database of words that groups words into "synsets" where each synset groups words based upon shared semantic concepts of synonymy and antonymy [18]. Using synonym augmentation allows the sentence to retain its original semantic meaning while introducing lexical variety, as the replacement words were selected based on semantic similarity from WordNet, ensuring contextual alignment [17]. However, since WordNet was originally designed to work with English, using WordNet synonyms for Indonesian—especially for informal language or language created by students—runs the risk of synonyms generating some semantic drift or replacements that make no sense altogether if the synonym used is contextually inappropriate [19]. For this study, in order not to completely undo its benefit, augmentation was only applied to entries thought to be "unclear" (label 1), thus providing an opportunity to expose the model to more representations of under-represented data. As the last thing to note, it is important to be careful to filter which synonym replacements are used with augmentation to ensure that the text being augmented retains meaningful integrity.

2.4 Splitting Data

Data splitting involves dividing the original dataset into two primary subsets: training data and test data. This process will determine the model's ability to detect patterns from the training data and evaluate those patterns in the test data. The data split ratio we chose will have a big influence on model performance. The percentage of training data will increase the model's ability to learn the whole data structure; accordingly, this must be weighed against having sufficient test data to ensure a reliable evaluation [20]. A higher proportion of training data generally improves the model's ability to capture the underlying structure of the dataset; however, it must be balanced with having sufficient test data to ensure robust evaluation. If the training ratio is too high (e.g., 90% training and only 10% testing), the model may overfit—learning not only the general patterns but also the noise and specific idiosyncrasies in the training data, resulting in poor generalization to new data. Conversely, if the training set is too small (e.g., 60% training and 40% testing), the model may underfit due to insufficient exposure to the full data distribution, failing to learn meaningful representations [21]. In this study, the chosen split ratios of 70:30, 80:20, and 90:10 were selected to provide a practical balance, allowing the model to train effectively while minimizing the risks of both overfitting and underfitting. These ratios also help produce more consistent and reliable evaluation metrics, particularly in terms of accuracy, precision, and recall.

2.5 CNN and SVM Model Implementation

The CNN model utilized in this study was designed to manage padded and tokenized input text with a specific length sequence. CNNs make accurate class predictions in text classification with convolution, pooling, and dense layers to find relationships or patterns with the words in the text [22]. A 1D convolutional layer that extracts local spatial features from word sequences comes after an embedding layer that transforms tokens into fixed-dimensional vector representations. Following the convolution process, a dense layer is used for additional feature processing after Global Max Pooling is used to reduce dimensionality and capture the most important features in a single dimension. To lessen the possibility of overfitting, the model is additionally outfitted with a dropout layer as a regularization method. Because the classification task is binary, the network output employs the sigmoid activation function. The Adam algorithm is used for optimization, and binary crossentropy—a loss function appropriate for binary classification tasks is used.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

The TF-IDF (Term Frequency–Inverse Document Frequency) framework is used for feature extraction from the SVM model that takes a feature pair characterization. SVM allows for classifying documents based on the patterns of words that are distributed in these documents, distinguishing between two categorical classes utilizing an optimal hyperplane based on feature vector representations of the text [23]. In order to help the model identify the most discriminative words for classification, each text is converted into a numerical vector, which reflects the importance of words both in the document and in the collection. The Support Vector Classifier (SVC) from the scikit-learn library is employed to build the model by focusing on determining the best hyperplane to separate the two data classes. The parameters C and gamma are changed or otherwise tuned as required but the default kernel will be applied.

2.6 Hyperparameter Tuning

Managing hyperparameters is critical to obtaining the best performance from any model and in this research, we applied two different methods for hyperparameterization depending on the models' characteristics. For the CNN model, I decided to implement the Keras Tuner library with a Random Search algorithm. This method looks for optimal values of important parameters such as the number of filters, kernel size of the convolution layer, number of neurons in the dense layer, dropout rate, and learning rate within a defined hyperparameter search space. Since the CNN model was built using a custom function, I was able to look at a multitude of architectural combinations, train the model on augmented data, and measure accuracy performance on validation data (for each configuration). Random Search was chosen due to the complex hyperparameter search space that is characteristic of deep learning models like CNN. Géron [24], illustrates that existing literature supports Random Search is often a more efficient search method than Grid Search in high dimensional spaces, as it is sampled from a wider portion of hyperparameter space in fewer samples/design points and focuses computational expense on parameters that have the greatest impact on performance. This is especially suitable for hyperparameter tuning deep neural networks since an exhaustive search would be computationally intractable.

In contrast, the SVM model is optimized using the Grid Search Cross Validation available in the scikit-learn library. We opted for Grid Search because SVM has fewer hyperparameters with known parameters; the hyperparameters are kernel type (linear, RBF, or sigmoid), the regularization parameter CCC, and the kernel coefficient. The model was trained using text data transformed using TF-IDF (Term frequency-inverse document frequency) and evaluated measures of classification that include accuracy, precision, recall, and the F1 score. Grid Search examined combinations of parameters in a parameter grid previously defined, covering all the possible combinations. We selected the parameters that achieved the best performance on the test data and produced the final model using SVM for subsequent technical and research steps.

2.7 Evaluation of Result

The evaluation of the CNN and SVM models was performed to evaluate the models' classification types of student questions in regard to ChatGPT. The testing data was evaluated under 70:30, 80:20, and 90:10 data splits. In addition to the baseline model, the model trained on augmented training data with synonyms was also evaluated. To objectively evaluate the model performances, many binary classification evaluation metrics were used including accuracy, precision, recall, and F1-score. Both models were assessed via 5-fold stratified cross-validation to provide strong and generalizable results. The models' performance—accuracy, precision, recall, and F1-score—within each fold were extracted and then averaged to account for sampling variance and minimize overfitting to a specific data split. Accuracy shows the overall percentage of correct predictions, while precision and recall show a general idea of accurate and complete predictions for each class.

However, accuracy might not be a proper measure of real model performance on imbalanced datasets—such as in this study, where most questions are classified as "clear"—because, as noted by Muller et al. [25], there are limitations to judging the models based only upon accuracy. Noting the level of correctness is not sufficient when conducting analysis on imbalanced datasets and a confusion matrix provides a much better representation of a classifier's performance. The confusion medatrix also allows us to measure the misclassifications between classes of data, especially when the class is underrepresented (e.g., unclear questions). Thus, accuracy derived from the confusion matrix (e.g., precision, recall, and F1-score) is considered much more valid when evaluating models trained on data with imbalanced distributions. Additionally, comparing metrics between models is another method of determining the effect of parameter tuning and data augmentation on the success of the CNN and SVM models.

3. RESULT AND DISCUSSION

3.1 Data Splitting Ratio

The dataset is divided into 70:30, 80:20, and 90:10 proportions in the first scenario. Convolution layers and one-dimensional embedding are used to construct CNN models, which also include dropout regularization and training monitoring via early stopping. In the meantime, SVC kernels from the scikit-learn library were used to classify SVM models that were created using TF-IDF features. Three data split scenarios (70:30, 80:20, and 90:10) were used to assess both strategies using a variety of classification metrics, such as accuracy, precision, recall, and F1-score. The 70:30 data proportion for the CNN and SVM algorithms is shown in Table 2.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

Table 2. Result of 70:30 Data Split

Model	Accuracy	Precision	Recall	F1-Score
CNN	0.83	0.81	0.74	0.76
SVM	0.83	0.80	0.74	0.76

Table 2 indicates that both models, CNN and SVM, performed similarly, achieving the same performance score of 83% on question classifications using a 70:30 split. However, CNN performed better than SVM in regards to precision with a score of 0.81 compared to SVM at 0.80. Both models also shared equal levels of sensitivity to classify the classified questions from each class having recall values of 0.74. The F1-scores for both models were 0.76, showing a similar level of recall balance with precision.

Table 3. Result of 80:20 Data Split

Model	Accuracy	Precision	Recall	F1-Score
CNN	0.84	0.81	0.80	0.81
SVM	0.85	0.83	0.76	0.79

Based on Table 3, the SVM model had the highest accuracy (85%) slightly ahead of the CNN which had an accuracy of 84%. However, for recall, CNN recorded a higher score of 0.80 and the SVM recorded a lower score of 0.76 which reflects a higher sensitivity of CNNs ability to identify the target class, but lower diagnostic accuracy when predicting overall. For precision, again SVM had a slightly higher score at 0.83 than CNN at 0.81, but when looking at the F1-score which is the harmonic mean of precision and recall CNN recorded a score of 0.81 which was slightly higher than SVM at 0.79. This demonstrated that CNN provides a better balance of precision and sensitivity in the 80:20 data split instance. So which overall account testified the superiority of SVM in accuracy and precision, CNN was also outperformed in recall and overall performance balance, which is reflected by the highest F1-score.

Table 4. Result of 90:10 Data Split

Model	Accuracy	Precision	Recall	F1-Score
CNN	0.85	0.82	0.81	0.81
SVM	0.84	0.79	0.84	0.76

Table 4 provides the evaluation results of the CNN and SVM models in the 90:10 data split case. CNN achieved the best accuracy of 85% by a slim margin over SVM at 84%. In addition, CNN showed a precision value of 0.82 in contrast to SVMs 0.79, indicating that CNN is better at predicting false positives than SVM. Meanwhile, SVM exhibited higher recall at 0.84 compared to CNN 0.81, suggesting that SVM was able to classify all true positives marginally better while sacrificing precision. In terms of the F1-score, CNN again had a better value of 0.81, while SVM had 0.76. The higher F1-score of CNN indicates that the model is better at managing the trade-off between precision and recall and is therefore more stable for the task of question classification in the 90:10 data split case.

A closer look at the results in Tables 2–4 provides some insights into the reasons for CNN's success in the 90:10 scenario. CNN is a data-hungry model, and the convolutional filters (also known as the filters) of CNN have the advantage of many more training examples, enabling them to learn richer n-grams over word sequences. As such, CNN can learn many more parameters when the training sample is larger (90%), resulting in the highest and most linear balanced F1-score to the best of our analysis (0.81). In contrast, SVM uses sparse TF-IDF vectors. As the dimensionality of the feature space increases from synonym-based data augmentation and/or long inputs, the hyperplane separating the two classes becomes progressively less stable, especially when the minority sample is under-represented. Thus, the explanation for SVM's F1-score dropping, while recall remained similar, is that it processes unique inputs where competitive aspects were more effective.

3.2 Hyperparameter Tuning

To gain indicators about model performance in classifying student question types on ChatGPT, both models, CNN, and SVM were hyperparameter tuned. This stage of the modeling process sets out to determine the hyperparameters of the model that yields the best performance, according to the relevant measures of accuracy, precision, recall, and F1 score. The tuning of hyperparameters of the CNN was done using Keras Tuner with Random Search as the tuning approach. The hyperparameters tuned were: number of convolution filters; kernel size; number of units in the dense layer; drop rate; and learning rate. The best configuration achieved an accuracy of 0.86, precision of 0.88, recall of 0.93, and F1 score of 0.90, highlighting a significant overall improvement on the post-tuning performance relative to pre-tuning performance.

Thus, the SVM model was optimized using the GridSearchCV method from the scikit-learn library by testing the parameters C, gamma, and kernel type. The best combination found was the RBF kernel with C=10 and gamma=0.01. The tuning metrics returned an accuracy of 0.81, precision of 0.88, recall of 0.86, and F1-score of 0.87 showing a significant improvement, especially in precision and recall.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

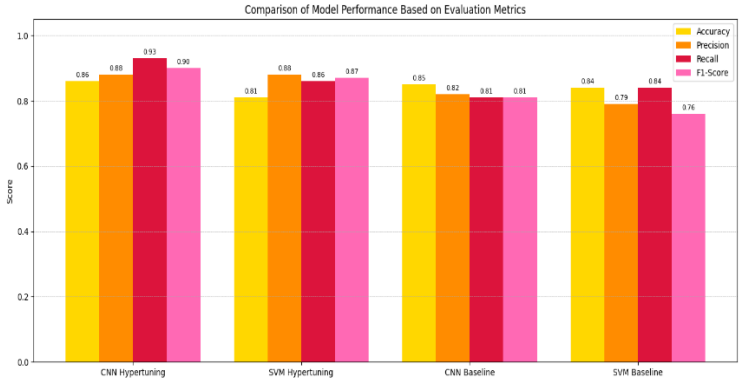

Figure 2. Hyperparameter Tuning Result

Figure 2 displays the visual outcomes of the model performance comparison before and after tuning. From the graph we can see that both models improved overall performance, gaining increases in recall and F1-score in CNN and precision for SVM.

3.3 Text Augmented

At this point, the effects of using synonym replacement types of data augmentation methods on the performance of models based on CNN and SVM are assessed. The CNN model, parameter tuning completed and using augmented data, had performance metrics of 0.90 for accuracy, 0.90 for precision, 0.90 for recall, and 0.90 for F1-score. These equally balanced metrics indicate augmenting data not only improved the model's ability to gain sensitivity towards both positive and negative data but also shows it maintained the consistency of the model's precision when classifying data. Figure 3 illustrates the differences in the performance of the CNN model after going through the data augmentation phase.

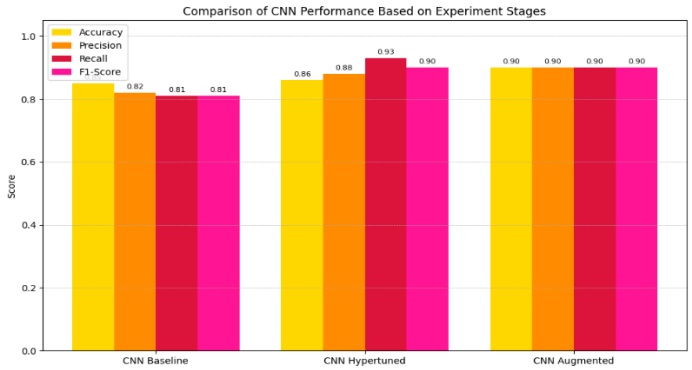

Figure 3. CNN Text Augmentation Result

In contrast, the SVM model performed poorly, even after tuning and training on some augmented data. The evaluation results revealed an accuracy of 0.81, precision of 0.81, recall of 0.81, and F1-score of 0.81; however, the minority class (labeled 1) still has reasonably low precision and recall scores of 0.61 and 0.65 respectively. This illustrates that augmented data within SVM did not provide an improvement on the model's ability to manage class imbalance. Figure 4 presents the changes in SVM model performance after the data augmentation stage.

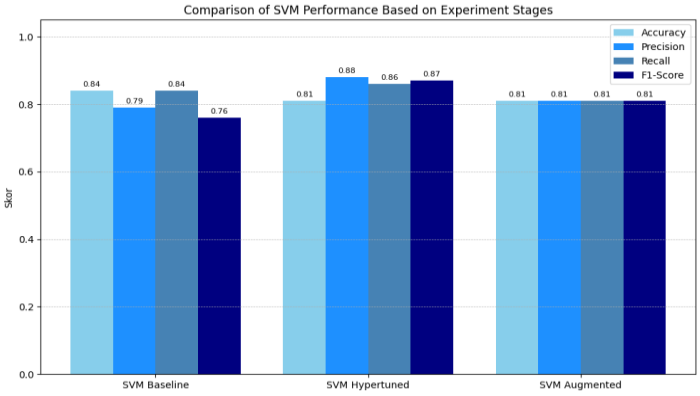

Figure 4. SVM Teks Augmentation Result

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

Initial evaluations across multiple data splits showed that the CNN model achieved its best performance under the 90:10 ratio, benefiting from a larger training set due to its data-hungry architecture. Its convolutional layers and word embeddings were able to learn richer patterns, resulting in strong and stable performance, followed by hyperparameter tuning of both models. Based on this result, synonym-based data augmentation was applied only on the 90:10 split after hypertuning. The CNN model remained robust after augmentation, as embeddings preserved semantic relationships while introducing lexical diversity, allowing the model to reinforce meaningful clusters. This led to balanced metrics across accuracy, precision, recall, and F1-score, each reaching 0.90.

Conversely, SVM's performance plateaued after augmentation. Since SVM relies on TF-IDF, which treats each word independently, synonym replacement introduced feature sparsity and reduced token overlap between training and test sets. This weakened the model's ability to form clear decision boundaries, especially for the minority class, with precision and recall dropping to 0.61 and 0.65. These outcomes highlight how CNN's architecture enables it to adapt to data augmentation, while SVM's reliance on sparse representations limits its effectiveness in such scenarios.

3.4 Error Analysis

To gain insight into common types of errors across both models, this study leveraged class-specific evaluation metrics. The SVM model consistently demonstrated lower precision and recall for the minority class (label 1) compared to the majority class. Even after tuning hyperparameters and data augmentation, SVM's performance decreased for the unclear or ambiguous questions. Given the sparse feature representation in the TF-IDF vector space, it seemed that SVM struggled with classifying these questions effectively. In contrast, the CNN model exhibited more balanced precision and recall between both classes, especially after synonym-based data augmentation. While SVM struggled classifying ambiguous questions, CNN's embedding-based architecture presented less of a notable distinction between semantically related tokens. The slight increase in CNN's precision and recall suggests it was becoming better at discerning and distinguishing the output of unclear or ambiguous questions, but still requires future tuning and testing on its sensitivity to ambiguous questions and its ability to recall and distinguish similar question types. Although this study did not conduct a manual inspection, these class-level metrics were an informative proxy for model weaknesses. It suggests opportunities for improvement in future models that incorporate attention mechanisms such as multi-layer perceptrons in conjunction with contextualized embeddings to account for conversational, semantic nuance, and imbalance across unbalanced class distributions.

4. CONCLUSION

This research demonstrates that Convolutional Neural Networks (CNN) are more effective than Support Vector Machines (SVM) at classifying a variety of student questions directed at ChatGPT when the dataset has a 90% training and 10% test split, tuning the model with Keras Tuner, and synonym replacement data augmentation. After tuning, the balanced score for the CNN was 0.90 for accuracy, precision, recall, and F1-score, whereas the SVM model achieved an overall 0.81 while misclassifying a number of minority-class questions. The CNN model is better suited to augmentation as its embedding layer represents synonyms as relational vectors; the convolution filters are then able to generalise across some of the lexical variation, while the SVM's sparse TF-IDF features would only widen the feature space every time a new form of word was encountered, without providing an equivalent semantic signal. These findings highlight the significance of pairing deep-learning architectures with data preprocessing, split ratio, systematic hyperparameter tuning, and augmentation to build robust educational question-classification models. In future work, it would be interesting to test transformer-based models, larger multilingual datasets, and deployment in real-time in classroom settings to improve upon the practical reach of this research.

REFERENCES

[1] E. Erlina *et al.*, "Penerapan Artificial Intelligence pada Aplikasi Chatbot sebagai Sistem Pelayanan dan Informasi Online pada Sekolah," *Journal of Information System and Technology*, vol. 4, no. 3, pp. 221–230, Nov. 2023, doi: 10.37253/joint.v4i3.6296.

[2] E. Finanti *et al.*, "Efektivitas Peran Chatgpt Sebagai Alat Bantu Penyelesaian Tugas Akademik Mahasiswa," *Kebumian dan Angkasa*, vol. 3, no. 2, pp. 74–85, Mar. 2025, doi: 10.62383/algoritma.v3i2.445.

[3] G. Raposo, R. Ribeiro, B. Martins, and L. Coheur, "Question Rewriting? Assessing Its Importance for Conversational Question Answering," Advances in Information Retrieval, pp. 199–206, Jan. 2022, doi: 10.1007/978-3-030-99739-7_23.

[4] V. Ramadhan, A. Asriyanik, and A. Pambudi, "Implementasi Algoritma Convolutional Neural Network Untuk Mengidentifikasi Berita Hoaks Berbahasa Indonesia," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 5, pp. 10945–10952, Oct. 2024, doi: 10.36040/jati.v8i5.10889.

[5] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," *Artif Intell Rev*, vol. 57, no. 4, Apr. 2024, doi: 10.1007/s10462-024-10721-6.

[6] N. Fathirachman Mahing *et al.*, "Klasifikasi Tingkat Stres Dari Data Berbentuk Teks Dengan Menggunakan Algoritma Support Vector Machine (Svm) Dan Random Forest," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 11, p. 5, Oct. 2024, doi: 10.25126/jtiik2024118010.

Building of Informatics, Technology and Science (BITS)
Volume 7, No 2, September 2025 Page: 1049−1056
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v7i2.7841

[7] I. A. Oktariansyah, F. R. Umbara, and F. Kasyidi, "Klasifikasi Sentimen Untuk Mengetahui Kecenderungan Politik Pengguna X Pada Calon Presiden Indonesia 2024 Menggunakan Metode IndoBert," *Building of Informatics, Technology and Science (BITS)*, vol. 6, no. 2, pp. 636–648, Sep. 2024, doi: 10.47065/bits.v6i2.5435.

[8] M. Desiawan and A. Solichin, "SVM Optimization with Grid Search Cross Validation for Improving Accuracy of Schizophrenia Classification Based on EEG Signal," *Jurnal Teknik Informatika*, vol. 17, no. 1, pp. 10–20, May 2024, doi: 10.15408/jti.v17i1.37422.

[9] R. Oktafiani, A. Hermawan, and D. Avianto, "Pengaruh Komposisi Split data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning," *Jurnal Sains dan Informatika*, pp. 19–28, Jun. 2023, doi: 10.34128/jsi.v9i1.622.

[10] W. Kurniasari, I. Saladin, B. Azhar, and N. Afifah, "Optimasi Hyperparameter Convolutional Neural Network-Long Short-Term Memory dengan Fitur GloVe untuk Klasifikasi Berita Palsu," *JSI : Jurnal Sistem Informasi (E-Journal)*, vol. 16, no. 1, 2024, [Online]. Available: http://jsi.ejournal.unsri.ac.id/index.php/jsi/index

[11] Seno B.A, Widodo, and Adhi B.P, "Penerapan Algoritma Support Vector Machine Untuk Mendeteksi Emosi Dari Teks Bahasa Indonesia," *PINTER : Jurnal Pendidikan Teknik Informatika dan Komputer*, vol. 8, no. 1, pp. 72–80, Jun. 2024, doi: 10.21009/pinter.8.1.8.

[12] I. M. García-López, C. S. González González, M.-S. Ramírez-Montoya, and J.-M. Molina-Espinosa, "Challenges of implementing ChatGPT on education: Systematic literature review," *International Journal of Educational Research Open*, vol. 8, p. 100401, Jun. 2025, doi: 10.1016/j.ijedro.2024.100401.

[13] I. A. Rahma and L. H. Suadaa, "Penerapan Text Augmentation untuk Mengatasi Data yang Tidak Seimbang pada Klasifikasi Teks Berbahasa Indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 6, pp. 1329–1340, Dec. 2023, doi: 10.25126/jtiik.2023107325.

[14] M. Aminudin and D. Wijayanti, "Identifikasi Pertanyaan Yang Diajukan Mahasiswa dalam Memecahkan Masalah Matematika," *JNPM (Jurnal Nasional Pendidikan Matematika)*, vol. 6, no. 4, p. 594, Dec. 2022, doi: 10.33603/jnpm.v6i4.7132.

[15] R. A. Widiyanti and W. Hadi, "Analisis Keterampilan Bertanya Siswa Berdasarkan Taksonomi Bloom Pada Pembelajaran Bahasa Indonesia Siswa Kelas VIII SMP Negeri 6 Torgamba Tahun Pembelajaran 2020/2021," *KODE: Jurnal Bahasa*, vol. 11, 2021, doi: 10.24114/kjb.v10i3.28448.

[16] D. Rifaldi, A. Fadlil, and Herman, "Teknik Preprocessing Pada Text Mining Menggunakan Data Tweet 'Mental Health,'" *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 3, no. 2, pp. 161–171, Apr. 2023, doi: 10.51454/decode.v3i2.131.

[17] I. Risma Huriah, A. Ismania Sita Widianingrum, T. Informatika, and U. Muhammadiyah Riau, "Optimasi Augmentasi Data Berbasis Synonym Replacement pada Klasifikasi Teks Berita Menggunakan Neural Network Optimization of Data Augmentation Based on Synonym Replacement in News Text Classification Using Neural Network," Jurnal Ilmiah Komputer dan Informatika, vol. 14, no. 1, pp. 2715–7849, 2025, doi: 10.34010/komputa.v14i1.

[18] S. M. Pamungkas, M. A. Yaqin, K. Z. Matondang, A. N. Anggraini, and Abd. C. Fauzan, "Analisis dan Perancangan Software WordNet Bahasa Indonesia dengan Graph Database," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 2, no. 2, pp. 198–209, Aug. 2020, doi: 10.28926/ilkomnika.v2i2.52.

[19] H. T. Kesgin and M. F. Amasyali, "Iterative Mask Filling: An Effective Text Augmentation Method Using Masked Language Modeling," *arXiv preprint arXiv:2401.01830*, pp. 450–463, Jan. 2024, doi: 10.1007/978-3-031-50920-9_35.

[20] Y. A. Prasetyo, E. Utami, and A. Yaqin, "Pengaruh Komposisi Split Data Terhadap Performa Akurasi Analisis Sentimen Algoritma Naïve Bayes dan SVM," *Journal homepage: Journal of Electrical Engineering and Computer (JEECOM)*, vol. 6, no. 2, 2024, doi: 10.33650/jeecom.v4i2.

[21] H. Bichri, A. Chergui, and M. Hain, "Investigating the Impact of Train / Test Split Ratio on the Performance of Pre-Trained Models with Custom Datasets," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 2, 2024, doi: 10.14569/IJACSA.2024.0150235.

[22] S. Paabanan Simanjuntak, S. Sandino Berutu, and G. C. Setyawan, "Implementasi Metode CNN pada Klasifikasi Sentimen terhadap Pelaksanaan Piala Dunia U-17 (Implementation of the CNN Method in Classifying Sentiments Regarding the Implementation of the U-17 World Cup)," *Journal of Engineering and Emerging Technology*, vol. 02, no. 01, 2024, [Online]. Available: www.jeet.unram.ac.id

[23] F. Abdusyukur, "Penerapan Algoritma Support Vector Machine (Svm) Untuk Klasifikasi Pencemaran Nama Baik Di Media Sosial Twitter," *Komputa : Jurnal Ilmiah Komputer dan Informatika*, vol. 12, no. 1, pp. 73–82, May 2023, doi: 10.34010/komputa.v12i1.9418.

[24] A. Géron, "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems," 2nd ed., Sebastopol, CA, USA: O'Reilly Media, 2019, ch. 10, pp. 321–325.

[25] A. C. Müller and S. Guido, "Introduction to Machine Learning with Python: A Guide for Data Scientists," 1st ed., D. Schanafelt, Ed., Sebastopol, CA, USA: O'Reilly Media, 2016, ch. 5, pp. 277–284.