Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 5 (2025) 1109 - 1120 e-ISSN: 2580-0760

Comparing Data Preprocessing Strategy on T5 Architecture to Classify ICD-10 Diagnosis

I Gusti Ngurah Lanang Wijayakusuma[1]*, Made Sudarma[2], I Ketut Gede Darma Putra[3], Oka Sudana[4], Ni Putu Dian Astutik[5]

[1]Department of Doctoral Engineering, Faculty of Engineering, Universitas Udayana, Bali, Indonesia
[2]Department of Electrical Engineering, Faculty of Engineering, Universitas Udayana, Bali, Indonesia
[3,4]Department of Information Technology, Faculty of Engineering, Universitas Udayana, Bali, Indonesia
[5]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Udayana, Bali, Indonesia

[1]lanang_wijaya@unud.ac.id, [2]msudarma@unud.ac.id, [3]ikgdarmaputra@unud.ac.id, [4]agungokas@unud.ac.id, [5]dianastutik03@gmail.com

Abstract

Manual ICD-10 coding in healthcare systems remains time-consuming, error-prone, and inefficient, particularly in resource-constrained settings. This study investigates the effect of various preprocessing strategies on the performance of the Text-to-Text Transfer Transformer (T5) model for primary diagnosis classification using structured clinical data. A total of 7,263 clinical records were collected from two high-density regions in Bali (Badung and Gianyar) between January 2023 and March 2024, then converted into descriptive text prompts for model training. Four experimental scenarios combined variations of input features and label configurations, comparing T5 with Oversampling against T5 with Easy Data Augmentation (EDA) plus Oversampling. Results showed that T5 with Random Oversampling consistently outperformed the EDA-based configuration across all scenarios, with performance gaps ranging from 8% to 19%. Scenario 4, which excluded body system features and the semantically overlapping E860 label, achieved the highest balance, reaching 84.7% accuracy, 85.1% precision, 84.7% recall, and 84.3% F1-score. Conversely, the EDA-based approach reduced training time by up to 72%, indicating a clear trade-off between performance and efficiency. Both configurations frequently misclassified semantically similar codes within the same ICD-10 categories, underscoring the difficulty of distinguishing clinically related diagnoses. Overall, the results suggest that careful selection of preprocessing strategies can enhance transformer-based medical text classification, while striking a balance between model performance and training efficiency. This work may serve as an initial reference for developing more efficient semi-automated medical coding systems in the Indonesian regional healthcare context.

Keywords: ICD-10 classification; medical NLP; preprocessing strategy; T5; text-to-text transformer

How to Cite: I. G. N. Lanang Wijayakusuma, M. Sudarma, I. K. G. Darma Putra, O. Sudana, and D. Astutik, "Comparing Data Preprocessing Strategy on T5 Architecture to Classify ICD-10 Diagnosis", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 5, pp. 1109 - 1120, Oct. 2025. *Permalink/*DOI: [https://doi.org/10.29207/resti.v9i5.6919

Received: July 7, 2025
Accepted: September 25, 2025
Available Online: October 25, 2025

This is an open-access article under the CC BY 4.0 License
Published by Ikatan Ahli Informatika Indonesia

1. Introduction

Medical diagnosis classification is a crucial component of healthcare systems, as it enables disease identification, statistical reporting, and standardized clinical decision-making [1]. To support this function, classification systems such as the International Classification of Diseases (ICD), developed by the World Health Organization (WHO), are widely used to standardize disease codes across countries, including Indonesia [2]. The implementation of ICD codes is vital for both patients and hospitals, particularly in encoding primary diagnoses that reflect the main condition of patients during hospitalization [3]. However, ICD code labeling is still predominantly carried out manually, a process that is not only time-consuming and costly but also prone to errors [4]. Such errors in medical coding can introduce bias and inconsistency into health data, potentially distorting disease prevalence estimates and undermining healthcare resource allocation and quality measurement [5], [6]. These issues underscore the need for efficient, accurate, and scalable solutions to support ICD coding in clinical practice.

To address these challenges, a variety of artificial intelligence-based automation approaches have been developed, particularly through the integration of Natural Language Processing (NLP) and deep learning

in the healthcare domain [7]. NLP enables computers to interpret and process unstructured medical texts, thereby presenting significant opportunities to improve the efficiency and accuracy of diagnosis classification. Various deep learning architectures have been applied for medical text classification. For example, Convolutional Neural Networks (CNNs) have demonstrated promising performance with precision ranging from 0.53 to 0.96, recall from 0.85 to 0.99, and F1-scores between 0.65 and 0.98 [8]. Similarly, Bidirectional Long Short-Term Memory (BiLSTM) models with word embeddings have achieved accuracy rates of up to 92.3% [9]. Building on these advances, the emergence of transformer architectures has further improved the capability to capture global context in text through self-attention mechanisms [10], [11]. The BERT (Bidirectional Encoder Representations from Transformers) model, as an encoder-only architecture, has been reported to achieve an accuracy ranging from 80.0% to 95.2% and an F1-score ranging from 81% to 94.3% in ICD-10 classification tasks across multiple studies [12] - [14]. These results demonstrate the strong potential of transformer-based models for classifying complex medical texts. Among various transformer models, encoder-decoder architectures such as T5 (Text-to-Text Transfer Transformer) offer greater flexibility in processing both textual input and output [15]. T5 reformulates all NLP tasks into a text-to-text format, enabling unified applications for classification, sentiment analysis, and even question answering. This model has achieved up to 91% accuracy in classifying tweets related to smoking and 94% accuracy in sentiment analysis tasks [16] Innovations such as SensoryT5, which integrates sensory knowledge into the attention mechanism, have shown superior performance in tasks involving emotion, sarcasm, and opinion classification [17]. Additionally, T5 has demonstrated the ability to convert natural language medical queries into SQL queries with an exact match accuracy of 80.63% on the MIMICSQL dataset [18]. These findings highlight the high versatility and adaptability of the T5 model, making it a promising candidate for contextual and generative ICD code classification tasks.

Although T5 offers strong adaptability, model performance is not solely determined by architecture; it is also heavily influenced by the data preprocessing strategies employed [19]. In NLP, preprocessing techniques such as lowercasing, stemming, lemmatization, removal of special characters, and systematic punctuation handling have been shown to improve the accuracy of neural network-based models [20]. In other domains, such as water potability prediction, preprocessing strategies including data cleaning, feature engineering, and dimensionality reduction (e.g., PCA, autoencoders) have consistently contributed to improved accuracy, especially in computationally constrained environments [21]. In the context of medical text analysis, preprocessing must be tailored to the characteristics of clinical language and the objectives of the study. For example, tokenization or stopword removal can significantly affect the accuracy of information extraction [22]. Applying text normalization and TF-IDF-based term selection to BERT has been shown to improve classification accuracy by up to 3% while reducing training time by 17% [23]. The complexity and high dimensionality of clinical datasets also hinder the development of decision support systems. Dimensionality reduction techniques, such as feature selection and feature extraction, are therefore essential for minimizing feature redundancy and simplifying computation [24], [25]. Additionally, domain-specific embeddings and medical terminology normalization have been shown to boost transformer model F1-scores to as high as 98.2% [26]. Another persistent challenge in clinical data processing is class imbalance, which can be caused by factors such as the low prevalence of rare diseases, incomplete datasets, and ethical constraints related to sensitive health information [27]. Effective preprocessing, including balancing techniques and targeted feature selection, plays a critical role in mitigating these issues [28], [29]. Despite the growing literature on NLP-based ICD-10 classification, most existing studies have either not examined preprocessing in depth or have focused on BERT rather than T5, as shown in Table 1.

Table 1. Related Work on BERT and T5 Models with Emphasis on Preprocessing Impact

Reference	Title	Task	Model	Focus on Preprocessing Impact
[12]	Automatic Assignment of ICD-10 Codes to Diagnostic Texts using Transformers-Based Techniques	ICD-10 Classification	BERT	No
[13]	Improving clinical documentation: automatic inference of ICD-10 codes from patient notes using BERT model	ICD-10 Classification	BERT	No
[14]	International Classification of Diseases Prediction from MIMIC-III Clinical Text Using Pre-Trained ClinicalBERT and NLP Deep Learning Models Achieving State of the Art	ICD-10 Classification	BERT	No
[23]	An Efficient Text Cleaning Pipeline for Clinical Text for Transformer Encoder Models	Clinical Text Classification	BERT	Yes
[16]	Enhancing sentiment and intent analysis in public health via fine-tuned Large Language Models on tobacco and e-cigarette-related tweets	Tweet Classification, Sentiment Analysis	T5	No
[17]	Leveraging sensory knowledge into Text-to-Text Transfer Transformer for enhanced emotion analysis	Emotion Classification	T5	No

No prior work has systematically evaluated multiple preprocessing strategies for the T5 model in ICD-10 primary diagnosis classification, particularly using real-world clinical data from Indonesia. This constitutes the primary research gap addressed in this study.

Therefore, this study aims to analyze the impact of various data preprocessing strategies on the performance of the T5 model in ICD-10 code classification tasks through controlled experimental scenarios that combine different input feature sets and label configurations. The expected benefits include providing healthcare institutions with practical guidance for designing more accurate and efficient semi-automated medical coding systems, thereby reducing manual coding errors and improving clinical data quality. The main contributions of this research are: presenting the first systematic evaluation of preprocessing strategies for the T5 model in ICD-10 primary diagnosis classification based on clinical data from Bali, demonstrating the impact of various data preprocessing scenarios on model performance, and offering empirical evidence to inform the design of future clinical NLP pipelines in resource-constrained environments.

2. Methods

The overall research workflow is outlined in Figure 1. It begins with data collection and cleaning, followed by the application of four distinct preprocessing strategies (S1, S2, S3, and S4). The preprocessed data is then transformed into prompt-style inputs compatible with the T5 model. The model training phase and subsequent performance evaluation follow this.

2.1 Data Collection

This research utilized primary data collected from medical clinics in two regions of Bali: South Bali (Badung Regency) and Central Bali (Gianyar Regency), between January 2023 and March 2025. These areas were selected based on data from the Bali Provincial Tourism Office (2023), which indicated that Gianyar and Badung ranked among the top two regencies in terms of receiving the highest numbers of international tourists. Furthermore, according to the 2022 statistics from the Central Bureau of Statistics, Badung and Gianyar are also among the most densely populated regencies in the province. To ensure data reliability and clinical accuracy, the dataset was reviewed and validated by a licensed medical clinician with relevant professional experience.

This context reflects the high level of mobility and interaction among both domestic and international tourists, resulting in a socially heterogeneous environment.

Such diversity can indirectly increase the risk of disease transmission, particularly in healthcare settings that serve populations with varied geographic backgrounds and health behaviors [30]. For this reason, the selected region was considered representative for exploring the dynamics of disease diagnosis and classification within a public health system utilizing ICD-based data.

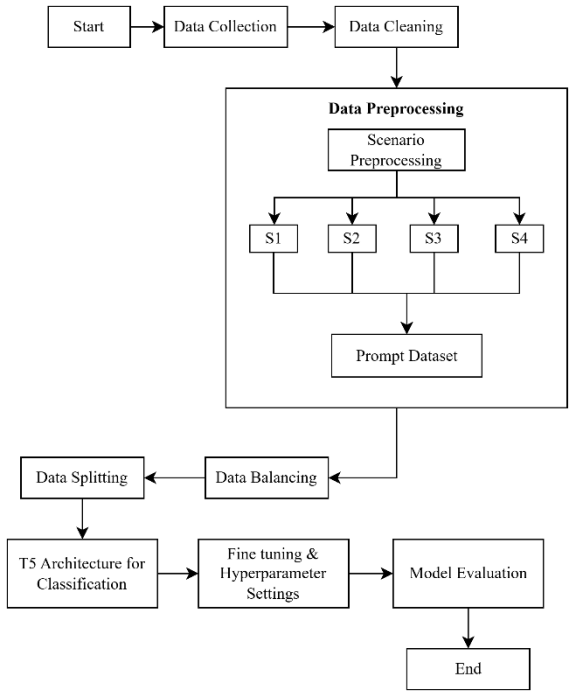


Figure 1. Research Workflow

A total of 7,263 samples were collected, with each sample comprising a combination of clinical features structured in a descriptive format, as illustrated in Table 2.

The distribution of diagnostic labels based on the 20 ICD-10 codes is visualized in Figure 2, providing an overview of the proportion of each category within the dataset.

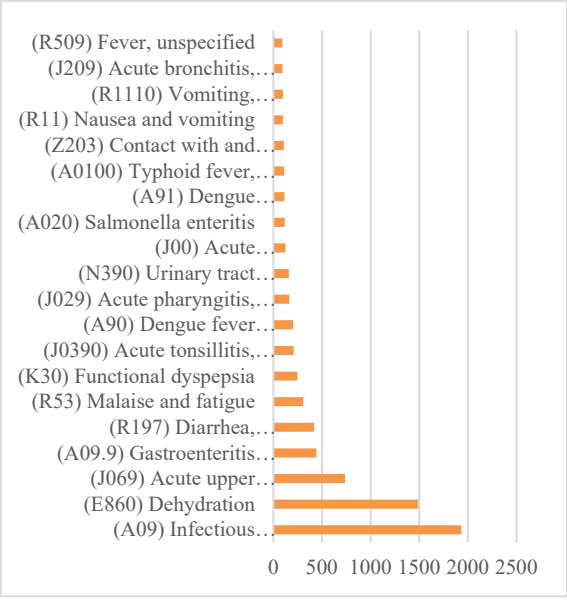


Figure 2. Distribution of Diagnoses by ICD-10 Code

Table 2. Feature Dataset

Feature Name	Description	Unit / Format
alergiObat	History of drug allergy	Text
keluhanUtama	Main complaint	Text
deskripsiKeluhan	Detailed description of the complaint	Text
historyOfRecentIllness	Recent illness history	Text
treatmentSoFar	Treatment or interventions previously given	Text
levelOfConsiusness	Patient's level of consciousness	Description / GCS Scale
respiratoryRate	Respiratory rate	Breaths/minute
bloodPressure	Blood pressure	mmHg
temperature	Body temperature	°C
pulse	Pulse rate	Beats/minute
o2Saturation	Oxygen saturation	%
VAS	Visual Analog Scale for pain	Scale 0–10
generalAppearance	General physical appearance	Text
eye, ear, nose, throat, tonsil	ENT and eye examination findings	Clinical observation text
cor	Heart (cardiac) examination	Text
pulmo	Lung (pulmonary) examination	Text
abdomen	Abdominal examination	Text
extrimities	Extremities examination	Text
statusLokalis	Localized clinical findings	Text
tinggiBadan	Body height	cm
beratBadan	Body weight	kg
sistol	Systolic blood pressure	mmHg
diastol	Diastolic blood pressure	mmHg
diagnosis	ICD-10 diagnosis label (top 20 most common codes)	ICD-10 code

2.2 Data Cleaning

The data cleaning process was carried out to ensure the quality and consistency of the dataset before it was used for modeling. Given the diverse formats and types of features—including free text, numerical values, and medical descriptions—cleaning procedures were tailored to the specific characteristics of each feature. In general, the cleaning steps included converting all text to lowercase, removing excessive whitespace, and eliminating irrelevant or unreadable characters that the model could not process.

For descriptive features derived from clinical observations— such as 'extrimities', 'generalAppearance', and 'levelOfConsiusness'—medical terminology was normalized to standardize various expressions that convey the same clinical meaning. Meanwhile, for numerical features such as 'tinggiBadan', 'beratBadan', 'o2Saturation', 'temperature', and 'bloodPressure', value validation was performed to ensure that the data fell within reasonable physiological ranges. This cleaning approach was designed to ensure that the data used for model training is immaculate, consistent, and clinically relevant.

2.3 Scenario Preprocessing

To evaluate the impact of feature composition on classification performance, this study implemented four distinct preprocessing scenarios, as summarized in Table 3. Each scenario differs in terms of the combination of input features used and the variation in ICD-10 diagnostic labels.

Table 3. Scenario Preprocessing

Alias	Input Feature	Label
S1	All Features (28)	20 (ICD-10 Code)
S2	All Features (28)	19 (without E860 Code)
S3	Without body system feature (20)	20 (ICD-10 Code)
S4	Without body system feature (20)	19 (without E860 Code)

Scenario 1 (S1) utilizes the complete set of input features and includes all 20 ICD-10 code categories available in the dataset as target labels. Scenario 2 (S2) also employs the full set of features but limits the classification task to 19 labels by removing one category, E860, which in this dataset corresponds to cases of dehydration.

Clinically, gastroenteritis is a medical condition that can lead to dehydration, indicating a clear causal relationship between the two conditions [31]. Dehydration symptoms are often closely related to and frequently co-occur in patients with gastroenteritis, particularly when significant fluid loss occurs, thereby creating the possibility of overlapping diagnoses [32], [33]. Consequently, there is a potential semantic overlap between E860 (Dehydration) and other labels such as A09 (Infectious gastroenteritis and colitis, unspecified) and A09.9 (Infectious gastroenteritis and colitis, unspecified). Based on this possible clinical overlap, the exclusion of E860 in S2 was designed as an experimental approach to minimize ambiguity in model predictions.

Scenario 3 (S3) omits features related to physical examinations of specific organ systems, such as the eyes, ears, nose, throat, tonsils, heart (cor), lungs (pulmo), and abdomen. Physical examination features describing body systems were excluded because their content was largely repetitive across patient records, resulting in low variance and minimal discriminative value for classification. Moreover, such generic descriptions offer limited diagnostic specificity, rendering them less helpful in predicting ICD-10 codes.

Scenario 4 (S4) combines the label removal approach of S2 (excluding E860) with the feature reduction strategy of S3, to investigate the combined impact of label and feature simplification on classification performance.

2.4 Prompt Dataset

All cleaned clinical features were then combined into a single structured text format, serving as the model's prompt or input, as illustrated in Figure 3.

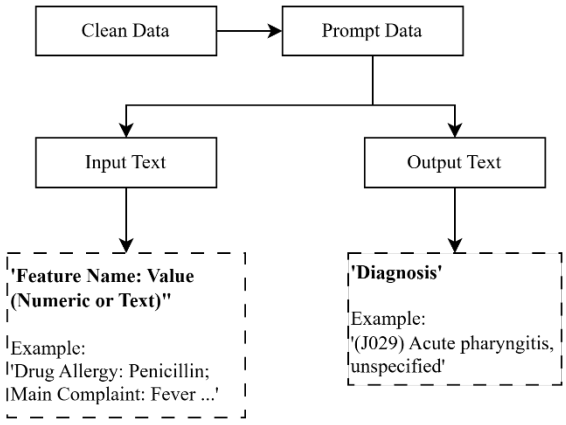


Figure 3. Structuring Clean Data into an Input-Output Prompt

This prompt format was constructed as a descriptive sentence that holistically represents the patient's medical condition by embedding the values of each feature into a unified text. The structure follows a repetitive pattern of feature name-value pairs, separated by semicolons.

A total of 28 features were used in the construction of the full input prompt: 'alergiObat', 'keluhanUtama', 'deskripsiKeluhan', 'historyOfRecentIllness', 'treatmentSoFar', 'levelOfConsiusness', 'respiratoryRate', 'bloodPressure', 'temperature', 'pulse', 'o2Saturation', 'VAS', 'generalAppearance', 'eye', 'ear', 'nose', 'throat', 'tonsil', 'cor', 'pulmo', 'abdomen', 'extrimities', 'statusLokalis', 'tinggiBadan', 'beratBadan', 'sistol', and 'diastol'.

Two variations of the prompt format were designed, depending on the preprocessing scenario applied. In scenarios that included all features (S1 and S2), the input prompt covered all observational data, including features related to body system examinations. In contrast, for scenarios excluding body system features (S3 and S4), the prompt was constructed without the following eight features: 'eye', 'ear', 'nose', 'throat', 'tonsil', 'cor', 'pulmo', and 'abdomen'. This prompt design allows the model to receive clinical information in the form of extended text resembling structured medical notes, aligning with the text-to-text paradigm employed by the T5 model.

2.5 Data Balancing

The issue of class imbalance in datasets—where the distribution of classes is uneven—can negatively impact the predictive performance of classification models [34]. This condition often leads to overfitting toward the majority classes, resulting in reduced overall model performance [34]. To address this issue, the present study adopted an oversampling approach, a widely used strategy that has been shown to improve model performance by balancing the number of samples across classes [34].

The initial distribution of diagnosis labels in the dataset revealed a substantial imbalance, with the number of samples per diagnostic category ranging from 63 to 1,637. Since the data in this study are represented as text-based tokens rather than vector-based numerical features, alternative oversampling methods such as SMOTE or ADASYN were not employed. Instead, random oversampling was performed to balance class distribution, followed by the application of Easy Data Augmentation (EDA) to the oversampled dataset to preserve semantic integrity while increasing data diversity [35]. EDA applies simple yet effective text-level transformations—such as synonym replacement, random insertion, random swap, and random deletion—to increase data diversity and reduce the risk of overfitting caused by duplicated samples [36], [37].

Following the oversampling process, each diagnostic category was represented equally in the dataset, allowing the model to learn from all classes in a balanced manner. Before oversampling, the dataset consisted of only 5,912 samples, with uneven class distribution. After oversampling, the total number of samples increased to 32,740, with each of the 20 diagnostic categories represented by exactly 1,637 samples. By applying this strategy, the model is expected to achieve more stable and fair classification performance across all classes, including those that were initially underrepresented.

2.6 T5 Architecture for Classification Task

This study employed the pre-trained T5 (Text-to-Text Transfer Transformer) model, specifically the *t5-small* variant, which contains approximately 60 million parameters, as illustrated in Figure 4. The choice of *t5-small* was motivated by prior findings showing that, despite its smaller size, it can achieve competitive performance compared to larger variants while offering substantially faster training times—making it more suitable for resource-constrained environments and relatively small datasets [38]. The dataset, after being cleaned and reformatted into prompt-based input—such as the example 'Diastolic: 80;'—was processed using the SentencePiece tokenizer. This tokenizer segments the text into subword units and automatically appends a special <eos> token at the end to mark the input boundary.

The resulting tokenized text is converted into numerical IDs and fed into the encoder component. On the decoder side, the model receives a sequence of tokens representing the target text, initialized with a special <pad> token. This setup enables the model to learn the target sequence directionally, leveraging the context of previously generated tokens to predict the next token.

The decoder's output is passed through a softmax layer, which produces a probability distribution over the entire vocabulary. At each prediction step, the token with the highest probability is selected as the output and subsequently used as input for the next decoding step. This autoregressive decoding process continues until the model generates the <eos> token signals that the complete sequence of labels has been successfully predicted.

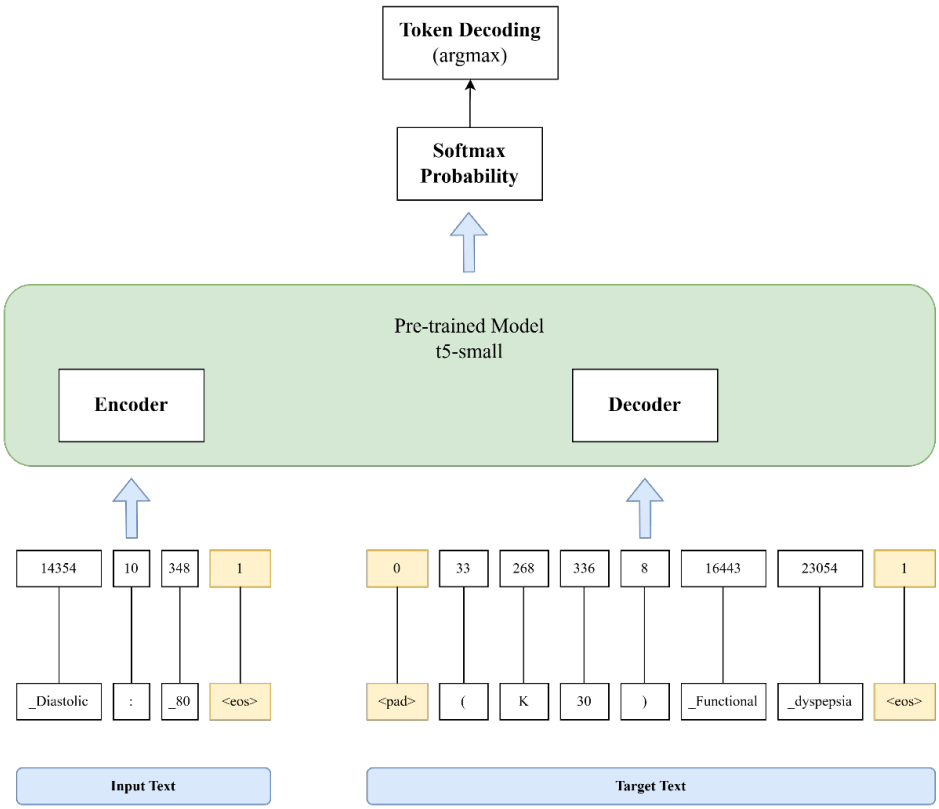


Figure 4. T5 Architecture for ICD-10 Diagnosis Classification

2.7 Fine-tuning and Hyperparameter Settings

The fine-tuning process was applied to the pre-trained *t5-small* model to adapt it specifically for the task of ICD-10 medical diagnosis code classification based on textual input. During this stage, several key hyperparameters were adjusted, as they significantly influence the model's training performance.

Table 4. Hyperparameter Settings

Hyperparameter	Value
Learning rate	5e-05
Epoch	10
Data ratio	80:20

The hyperparameter values listed in Table 4 were selected through a trial-and-error approach, in which various parameter combinations were empirically tested to identify the configuration that yielded the best performance on the validation set. A learning rate of 5e-05 was chosen, as it provided stable convergence without signs of divergence during training. The model was trained for 10 epochs, a duration deemed sufficient for learning meaningful data representations while minimizing the risk of overfitting.

2.8 Evaluation Model

Model performance was evaluated to assess the extent to which the fine-tuned model could accurately classify ICD-10 medical diagnosis codes based on textual input. This study employed a set of commonly used evaluation metrics for classification tasks, particularly in multi-class settings: accuracy, precision, recall, and F1-score. In addition, a confusion matrix was used to provide a visual representation of the model's prediction distribution across all classes.

Accuracy reflects the proportion of correct predictions over the total number of cases, while precision and recall are used to evaluate the quality of predictions for the positive class—a critical consideration in clinical contexts. The F1-score offers a balanced measure between precision and recall, which is particularly valuable when label distributions are imbalanced. Meanwhile, the confusion matrix provides more detailed insights into correct classifications as well as classification errors (false positives and false negatives) for each class. All evaluation metrics were calculated using the test set to ensure an objective assessment of the model's performance

3. Results and Discussions

In this study, the dataset consisted of clinician-validated clinical records from Bali that had been balanced before training. The training and validation loss curves for the four T5 experiments with random oversampling are presented in Figure 5. In Scenario 1 (Figure 5a), the validation loss steadily decreased throughout training, while the training loss initially increased during the early epochs before declining and converging toward the validation loss, reflecting good generalization. Scenario 2 (Figure 5b) showed a sharp decline in both losses until epoch four, followed by a plateau with minimal separation between curves, indicating stable convergence. Scenario 3 (Figure 5c) followed a similar downward trend, with a brief spike in training loss at epoch five—likely due to variability introduced by the

oversampled data—before both losses returned to low and balanced values. Scenario 4 (Figure 5d) demonstrated a pronounced early reduction in loss, with both curves closely aligned throughout most of the training process.

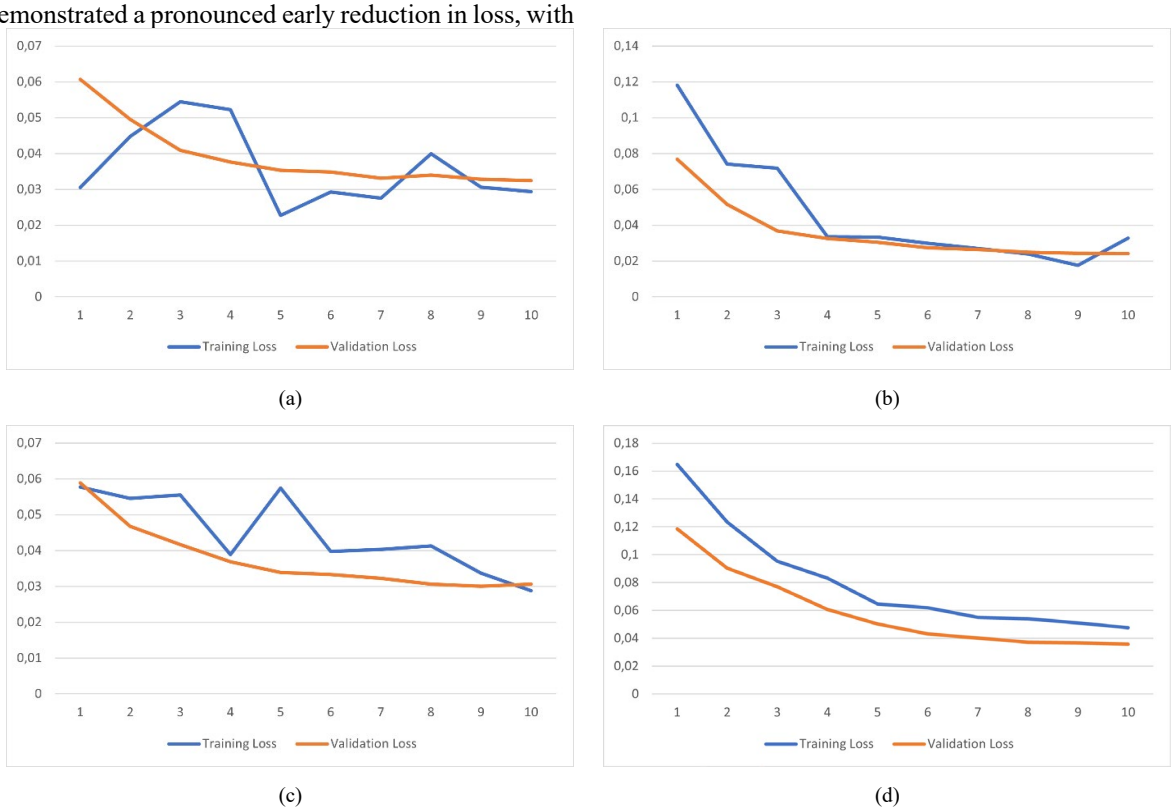


Figure 5. Training and Validation Loss T5 + Random Oversampling (a) S1, (b) S2, (c) S3, (d) S4

Across the four scenarios, the first three did not show signs of overfitting. However, the training loss exhibited noticeable fluctuations at specific points, suggesting the model's sensitivity to variations in the oversampled data. In contrast, Scenario 4 demonstrated the most stable decrease in both training and validation losses, with a small gap between them, which indicates a more consistent learning process.

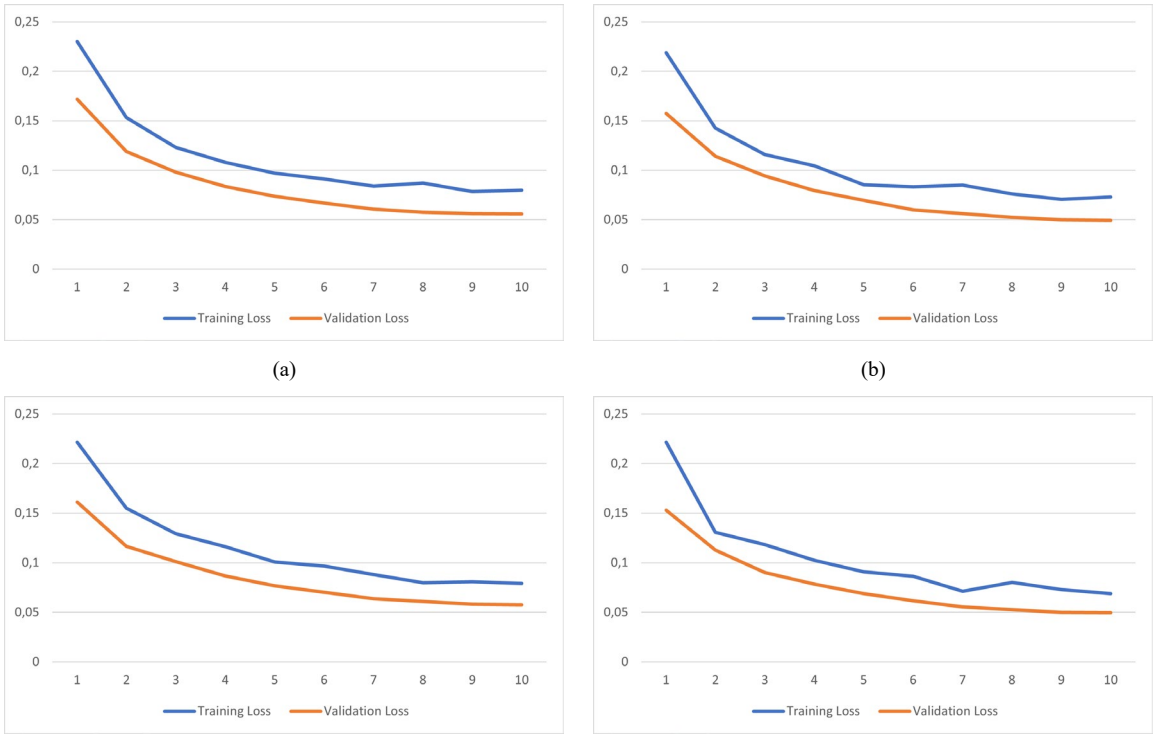


Figure 6. Training and Validation Loss T5 + EDA Oversampling (a) S1, (b) S2, (c) S3, (d)

When compared with the T5 + EDA Oversampling configuration, as shown in Figure 6, the learning curves exhibit a more consistent and stable downward trend across all four scenarios, with the validation loss consistently remaining slightly lower than the training loss throughout the training process. These smoother trajectories suggest that EDA may help stabilize training and improve the model's robustness against potential distribution shifts.

Performance metrics are summarized in Table 5, showing that across all scenarios (S1–S4), the T5 + Random Oversampling configuration consistently outperformed T5 + EDA Oversampling in accuracy, precision, recall, and F1-score, with performance gaps ranging from approximately 8% to 19%. The most significant drop occurred in S1, where accuracy decreased from 84.9% to 71.6% after adding EDA, while the smallest gap was observed in S4.

Table 5. Performance Comparison for Each Scenario

Scenario	T5 + Random Oversampling				T5 + EDA Oversampling			
	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
S1	84,9%	84,3%	84,9%	84,2%	71,6%	71,7%	71,6%	70%
S2	90,2%	90,5%	90,2%	90,1%	75,6%	75,8%	75,7%	75%
S3	85,5%	85%	85,5%	84,9%	72,8%	72,4%	72,8%	71,6%
S4	84,7%	85,1%	84,7%	84,3%	76,2%	76,6%	76,2%	75,7%

These results suggest that, in this context, EDA combined with oversampling reduced classification performance compared to using oversampling alone, likely due to the introduction of synthetic variations that were less representative of the original data distribution, causing the model to learn less relevant patterns and thus lowering predictive accuracy. However, adding EDA improved training stability across all scenarios, indicating its potential to enhance robustness when the model is exposed to new or shifted data distributions. Considering both stability and performance, Scenario 4, which utilized T5 with random oversampling, achieved the best overall results among the four evaluation metrics.

To gain deeper insights into the model's classification errors, confusion matrices for each scenario were analyzed, as shown in Figure 7 (T5 + Random Oversampling) and Figure 8 (T5 + EDA Oversampling). In all scenarios, both models most frequently misclassified disease codes E860, A09, A90, A0100, J029, and J0390. This illustrates the challenge of classifying complex clinical text, where high semantic similarity and overlapping clinical contexts among labels within the same ICD-10 category can cause ambiguity in the model's text representations.

In Scenario 1, both models most frequently misclassified label E860. In Scenario 2, after excluding E860, the highest misclassifications shifted to other labels within the same category—namely A09, A90, J029, J069, and J0390. This may be attributed to the relatively small dataset, which limited the model's ability to generalize well to labels within the same category. Nevertheless, both models performed better in this scenario compared to Scenario 1.

In Scenario 3, which excluded the body system feature, the most frequent misclassifications occurred for E860, A09, A0100, and J0390. Performance declined compared to Scenario 2 but remained higher than Scenario 1, suggesting that excluding the body system feature can improve model performance in specific settings.

In Scenario 4, which excluded both the body system feature and the E860 label, the most frequent misclassifications were similar to those observed in the previous scenario.

For T5 + Random Oversampling, Scenario 4's performance was comparable to Scenario 1, with the best results in Scenario 2, followed by Scenario 3, while Scenarios 1 and 4 performed similarly. This indicates that excluding label E860 had a notable effect on the model's performance. For T5 + EDA Oversampling, Scenario 4 yielded the highest performance, followed by Scenario 2, indicating that the removal of both the E860 label and the body system feature substantially improved the results for this configuration.

In addition to evaluating classification performance, training and inference times were analyzed to assess computational efficiency, as shown in Table 6. Across both models, Scenario 4 had the shortest execution time—24.8 minutes for T5 + Random Oversampling and 16.9 minutes for T5 + EDA Oversampling—significantly faster than the other scenarios. Scenario 3 followed, then Scenario 2, while Scenario 1 had the most extended duration (179.3 minutes and 27.4 minutes, respectively).

This variation is closely tied to the complexity of features and the number of labels used in each scenario. Scenario 4, which excluded both body system features and the E860 label, had the lowest input complexity, resulting in substantially faster training and inference.

Table 6. Computational Time for Each Scenario

Scenario	T5 + Random Oversampling Time (minute)	T5 + EDA Oversampling Time (minute)
S1	179,3	27,4
S2	163,5	25,8
S3	137,8	17,6
S4	24,8	16,9

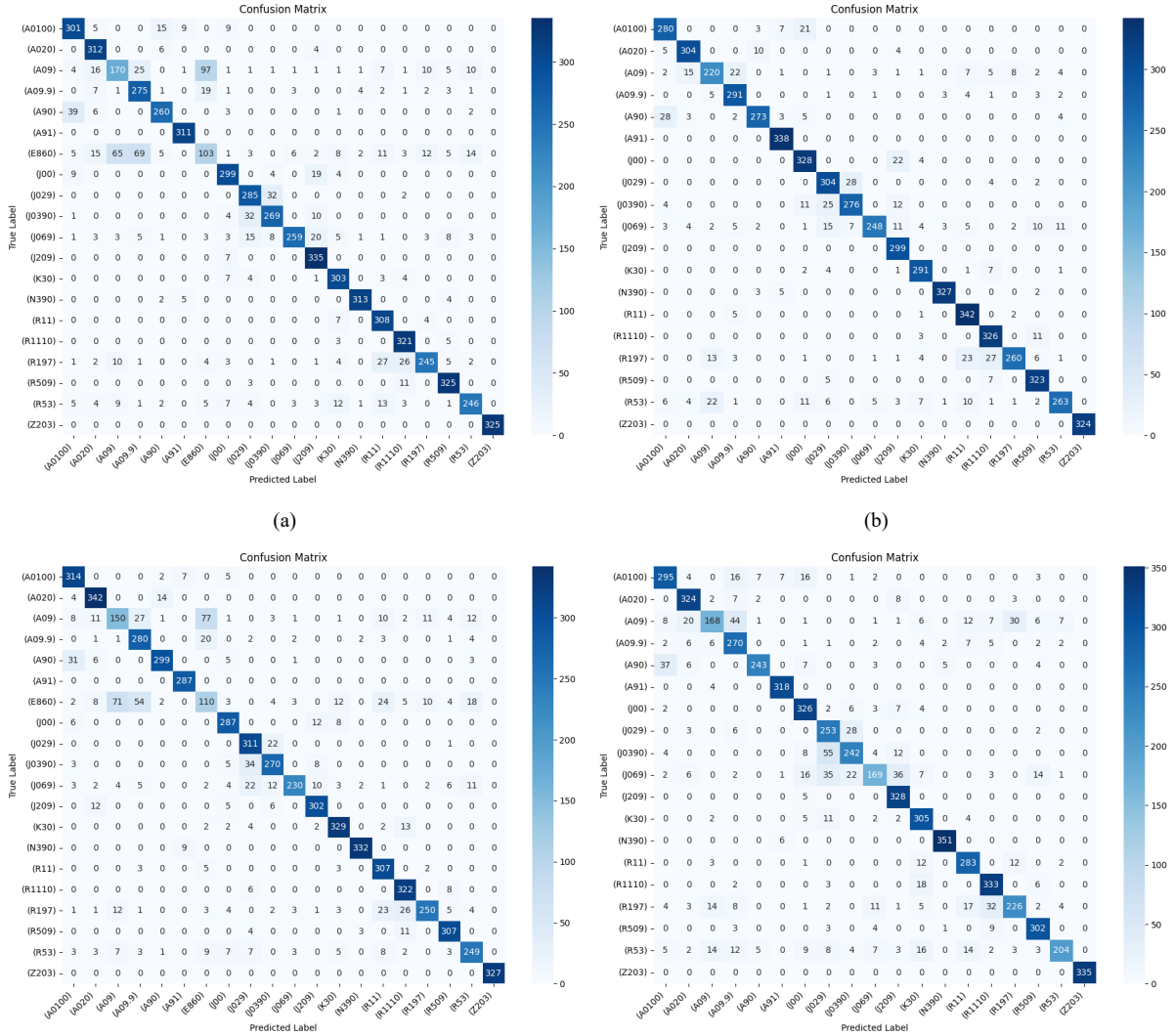


Figure 7. Confusion Matrix Model T5 + Random Oversampling (a) S1, (b) S2, (c) S3, (d) S4

In contrast, Scenario 1, which included the complete set of features and all labels, required more extensive processing due to the larger data volume. From a computational perspective, switching from the non-EDA to the EDA configuration resulted in a reduction of up to 72% in training time—a substantial efficiency gain—although predictive performance declined. In Scenario 4, the time difference between models was only 7.9 minutes, with an 8.5% performance gap.

Overall, the findings suggest that strategic modifications to preprocessing—such as reducing the number of input features or removing semantically overlapping labels—can significantly impact both performance and efficiency in T5-based medical text classification. While EDA reduced predictive metrics in this study, it consistently enhanced stability, which may be advantageous for real-world deployment under distributional shifts.

The results also reveal a clear trade-off between predictive performance and computational speed, a consideration of practical importance in resource-constrained clinical environments.

The study's conclusions should be considered in light of certain limitations. Although clinicians validated the dataset, its relatively small size may have constrained the model's generalizability, especially for underrepresented labels. Furthermore, while random oversampling and EDA aimed to mitigate class imbalance, they may have introduced patterns that are not fully representative of real clinical data, thereby influencing the model's learning behavior. The experiments were limited to the T5 architecture, and results may differ with alternative transformer-based models.

Future research could explore domain-adapted pre-trained models trained on large-scale clinical corpora to improve the representation of specialized medical terminology. Approaches such as hierarchical classification or merging clinically similar ICD-10 codes could reduce semantic overlap and enhance accuracy for closely related diagnoses. Additionally,

advanced augmentation techniques that preserve semantic fidelity while diversifying training samples could be investigated to combine the stability benefits of EDA with improved predictive performance. Finally, evaluating the models in larger and more diverse datasets, as well as in real-world deployment settings, would provide more substantial evidence of their practical utility.

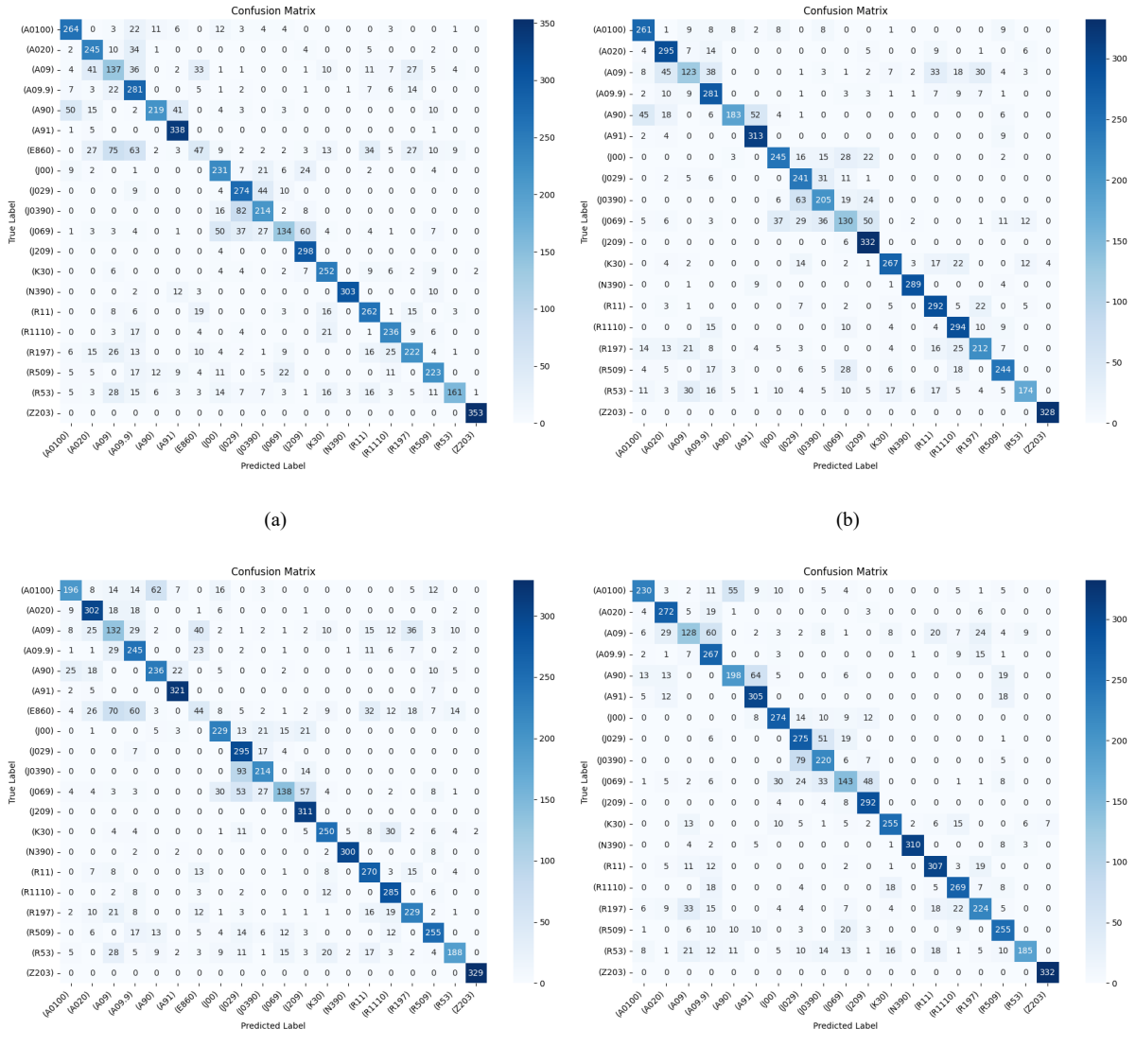


Figure 8. Confusion Matrix Model T5 + EDA Oversampling (a) S1, (b) S2, (c) S3, (d) S4

4. Conclusions

This study systematically evaluated the impact of various data preprocessing strategies on T5 model performance for ICD-10 code classification using clinical data from Bali, directly addressing the research question of how different input feature sets and label configurations affect model accuracy and computational efficiency. The findings reveal that T5 + Random Oversampling consistently outperformed T5 + EDA Oversampling across all evaluation metrics, with performance gaps ranging from 8% to 19%. Switching to the EDA configuration yielded a reduction of up to 72% in training time across all scenarios, demonstrating a clear trade-off between predictive performance and computational efficiency. Scenario 4, which excluded both body system features and the semantically overlapping E860 label, achieved optimal balance between classification accuracy and computational efficiency in both configurations.

Contrary to expectations, EDA combined with oversampling reduced classification performance, likely because synthetic variations were less representative of the original data distributions, although it enhanced training stability across all scenarios. The study demonstrates that strategic preprocessing choices—particularly feature reduction and removal of semantically overlapping labels—can substantially impact both model performance and computational efficiency. In both configurations, codes within the same ICD-10 categories are frequently misclassified due to high semantic similarity.

This research makes three significant contributions to the field: establishing the first systematic evaluation of T5-based preprocessing strategies for medical text classification in a regional Indonesian healthcare context, providing empirical evidence of measurable preprocessing impacts on generative language models, and delivering actionable guidelines for implementing semi-automated medical coding systems in resource-constrained environments.

The practical implications extend to healthcare institutions seeking to reduce manual coding errors and improve clinical data quality, as the identified performance-efficiency trade-offs offer concrete pathways for deploying AI-assisted diagnostic tools within computational limitations. While challenges remain in distinguishing semantically similar medical diagnoses, this work establishes a foundation for optimizing generative language models in clinical applications and identifies future research directions, including hierarchical classification approaches and domain-adapted models, to address the persistent challenge of disambiguating clinically related diagnoses.

Acknowledgements

The author declares no conflict of interest.

References

[1] K. M. Mitura *et al.*, "ICD-10 Classification in the Practice of Emergency Medical Teams: New Insights," *Emerg Med Int*, vol. 2024, no. 1, p. 8506561, 2024, doi: https://doi.org/10.1155/2024/8506561.

[2] Y. Hong and M. L. Zeng, "International Classification of Diseases (ICD)," *Knowledge Organization*, vol. 49, no. 7, pp. 496–528, 2022, doi: 10.5771/0943-7444-2022-7-496.

[3] Z. Ma, J. Li, M. Sufyan, and J. Li, "Automated ICD Coding for Primary Diagnosis Based on Graph Convolution Network," *Conf Proc IEEE Int Conf Syst Man Cybern*, pp. 3216–3221, 2023, doi: 10.1109/SMC53992.2023.10394578.

[4] P. Nie, H. Wu, and Z. Cai, "Towards Automatic ICD Coding via Label Graph Generation," *Mathematics*, vol. 12, no. 15, p. 2398, Aug. 2024, doi: 10.3390/MATH12152398.

[5] K. Burks, J. Shields, J. Evans, J. Plumley, J. Gerlach, and S. Flesher, "A systematic review of outpatient billing practices," *SAGE Open Med*, vol. 10, p. 20503121221099020, 2022, doi: 10.1177/20503121221099021.

[6] J. Quint and A. Brownrigg, "What Should Clinicians Know About How Coding Influences Epidemiological Research?," *AMA J Ethics*, vol. 27, no. 1, pp. E51–E57, Jan. 2025, doi: 10.1001/AMAJETHICS.2025.51.

[7] C. Crema, G. Attardi, D. Sartiano, and A. Redolfi, "Natural language processing in clinical neuroscience and psychiatry: A review," *Front Psychiatry*, vol. 13, 2022, doi: 10.3389/fpsyt.2022.946387.

[8] J. H. B. Masud, C.-C. Kuo, C.-Y. Yeh, H.-C. Yang, and M.-C. Lin, "Applying Deep Learning Model to Predict Diagnosis Code of Medical Records," *Diagnostics*, vol. 13, no. 13, 2023, doi: 10.3390/diagnostics13132297.

[9] J. Jasmir, W. Riyadi, S. R. Agustini, Y. Arvita, D. Meisak, and L. Aryani, "Bidirectional Long Short-Term Memory and Word Embedding Feature for Improvement Classification of Cancer Clinical Trial Document," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 4, pp. 505–510, Aug. 2022, doi: 10.29207/RESTI.V6I4.4005.

[10] S. Ayanouz, A. abdelhakim Boudhir, and M. Ben Ahmed, "Unleashing the Power of NLP and Transformers: A Game-Changer in Medical Research and Clinical Practice and a revolution of Medical Text Analysis: Case Study: Cancer report classification by priority," in *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, in NISS '23. New York, NY, USA: Association for Computing Machinery, 2023. doi: 10.1145/3607720.3607787.

[11] F. Alafari, M. Driss, and A. Cherif, "Advances in natural language processing for healthcare: A comprehensive review of techniques, applications, and future directions," *Comput Sci Rev*, vol. 56, p. 100725, 2025, doi: 10.1016/j.cosrev.2025.100725.

[12] M. H. Popescu, K. Roitero, S. Travasci, and V. Della Mea, "Automatic Assignment of ICD-10 Codes to Diagnostic Texts using Transformers Based Techniques," in *2021 IEEE 9th International Conference on Healthcare Informatics (ICHI)*, 2021, pp. 188–192. doi: 10.1109/ICHI52183.2021.00037.

[13] E. Al-Bashabsheh, A. Alaiad, M. Al-Ayyoub, O. Beni-Yonis, R. A. Zitar, and L. Abualigah, "Improving clinical documentation: automatic inference of ICD-10 codes from patient notes using BERT model," *Journal of Supercomputing*, vol. 79, no. 11, pp. 12766–12790, Jul. 2023, doi: 10.1007/S11227-023-05160-Z.

[14] I. Aden, C. H. T. Child, and C. C. Reyes-Aldasoro, "International Classification of Diseases Prediction from MIMIIC-III Clinical Text Using Pre-Trained ClinicalBERT and NLP Deep Learning Models Achieving State of the Art," *Big Data and Cognitive Computing*, vol. 8, no. 5, p. 47, May 2024, doi: 10.3390/BDCC8050047.

[15] H. Chen, "Comparison of Large Language And Vision Models on Representative Downstream Tasks," *2023 International Conference on Image Processing, Computer Vision and Machine Learning, ICICML 2023*, pp. 307–311, 2023, doi: 10.1109/ICICML60161.2023.10424913.

[16] S. Elmitwalli, J. Mehegan, A. Gallagher, and R. Alebshehy, "Enhancing sentiment and intent analysis in public health via fine-tuned Large Language Models on tobacco and e-cigarette-related tweets," *Front Big Data*, vol. 7, Nov. 2024, doi: 10.3389/FDATA.2024.1501154.

[17] Q. Zhao, Y. Xia, Y. Long, G. Xu, and J. Wang, "Leveraging sensory knowledge into Text-to-Text Transfer Transformer for enhanced emotion analysis," *Inf Process Manag*, vol. 62, no. 1, p. 103876, Jan. 2025, doi: 10.1016/J.IPM.2024.103876.

[18] A. Marshan, A. N. Almutairi, A. Ioannou, D. Bell, A. Monaghan, and M. Arzoky, "MedT5SQL: a transformers-based large language model for text-to-SQL conversion in the healthcare domain," *Front Big Data*, vol. 7, Jun. 2024, doi: 10.3389/FDATA.2024.1371680.

[19] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Inf Syst*, vol. 121, p. 102342, Mar. 2024, doi: 10.1016/J.IS.2023.102342.

[20] H. S. Woo, J. M. Kim, and W. G. Lee, "Validation of Text Data Preprocessing Using a Neural Network Model," *Math Probl Eng*, vol. 2020, no. 1, p. 1958149, Jan. 2020, doi: 10.1155/2020/1958149.

[21] K. Phorah, M. Sumbwanyambe, and M. Sibiya, "Systematic Literature Review on Data Preprocessing for Improved Water Potability Prediction: A Study of Data Cleaning, Feature Engineering, and Dimensionality Reduction Techniques," *Nanotechnol Percept*, vol. 20, no. S11, pp. 133–151, Sep. 2024, doi: 10.62441/NANO-NTP.V20IS11.1945.

[22] C. P. Chai, "Comparison of text preprocessing methods," *Nat Lang Eng*, vol. 29, no. 3, pp. 509–553, May 2023, doi: 10.1017/S1351324922000213.

[23] S. Z. Ridoy, J. Sultana, Z. F. Ria, M. A. Uddin, M. H. Rahman, and R. M. Rahman, "An Efficient Text Cleaning Pipeline for Clinical Text for Transformer Encoder Models," in *2024 IEEE 12th International Conference on Intelligent Systems (IS)*, 2024, pp. 1–9. doi: 10.1109/IS61756.2024.10705199.

[24] B. Kar and B. K. Sarkar, "A Hybrid Feature Reduction Approach for Medical Decision Support System," *Math Probl Eng*, vol. 2022, no. 1, Jan. 2022, doi: 10.1155/2022/3984082.

[25] D. Dhinakaran, L. Srinivasan, S. Edwin Raja, K. Valarmathi, and M. Gomathy Nayagam, "Synergistic feature selection and distributed classification framework for high-dimensional medical data analysis," *MethodsX*, vol. 14, Jun. 2025, doi: 10.1016/J.MEX.2025.103219.

[26] C. Mugisha and I. Paik, "Comparison of Neural Language Modeling Pipelines for Outcome Prediction from Unstructured Medical Text Notes," *IEEE Access*, vol. 10, pp. 16489–16498, 2022, doi: 10.1109/ACCESS.2022.3148279.

[27] M. Salmi, D. Atif, D. Oliva, A. Abraham, and S. Ventura, "Handling imbalanced medical datasets: review of a decade of research," *Artif Intell Rev*, vol. 57, no. 10, pp. 1–57, Oct. 2024, doi: 10.1007/S10462-024-10884-2.

[28] A. Ghavidel and P. Pazos, "Machine learning (ML) techniques to predict breast cancer in imbalanced datasets: a systematic review," *Journal of Cancer Survivorship*, vol. 19, no. 1, pp. 270–294, Feb. 2023, doi: 10.1007/S11764-023-01465-3.

[29] Y. Ren, D. Wu, Y. Tong, A. López-DeFede, and S. Gareau, "Issue of Data Imbalance on Low Birthweight Baby Outcomes Prediction and Associated Risk Factors Identification: Establishment of Benchmarking Key Machine Learning Models With Data Rebalancing Strategies," *J Med Internet Res*, vol. 25, p. e44081, May 2023, doi: 10.2196/44081.

[30] D. Fechner, S. Reid, and S. Dolnicar, "Tourism and Emerging Infectious Diseases: More Connections Than First Meet the Eye," *J Travel Res*, vol. 62, no. 5, pp. 935–948, May 2023, doi: 10.1177/00472875221127718.

[31] T. R. Rao, B. Pratiksha, C. Navyasri, D. J. Kalita, and S. Alam, "Gastroenteritis: A Comprehensive Review," *International Journal of Current Science Research and Review*, vol. 07, no. 10, Oct. 2024, doi: 10.47191/IJCSRR/V7-I10-07.

[32] K. A. Houston *et al.*, "Gastroenteritis aggressive versus slow treatment for rehydration (GASTRO): A phase II rehydration trial for severe dehydration: WHO plan C versus slow rehydration," *BMC Med*, vol. 17, no. 1, Jul. 2019, doi: 10.1186/S12916-019-1356-Z.

[33] L. Hartling, S. Bellemare, N. Wiebe, K. F. Russell, T. P. Klassen, and W. R. Craig, "Oral versus intravenous rehydration for treating dehydration due to gastroenteritis in children," *Cochrane Database of Systematic Reviews*, vol. 2006, no. 3, Jul. 2006, doi: 10.1002/14651858.CD004390.PUB2.

[34] M. Mujahid *et al.*, "Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering," *J Big Data*, vol. 11, no. 1, pp. 1–32, Dec. 2024, doi: 10.1186/S40537-024-00943-4.

[35] Z. Cai, Z. Li, Y. Liu, L. Guo, and Y. Song, "Ustnlp16 at SemEval-2025 Task 9: Improving Model Performance through Imbalance Handling and Focal Loss," Apr. 2025, doi: 10.48550/arXiv.2505.00021.

[36] S. Henning, W. Beluch, A. Fraser, and A. Friedrich, "A Survey of Methods for Addressing Class Imbalance in Deep-Learning Based Natural Language Processing," in *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, A. Vlachos and I. Augenstein, Eds., Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 523–540. doi: 10.18653/v1/2023.eacl-main.38.

[37] J. Wei and K. Zou, "EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 6382–6388. doi: 10.18653/v1/D19-1670.

[38] D. Setiyarini, T. B. Adji, and I. Hidayah, "Evaluating Performance of Transformer Models for Dialogue Summarization: A Comparison of T5-Base, T5-Small, and BART-Base," in *2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 2024, pp. 184–190. doi: 10.1109/COMNETSAT63286.2024.10862384.