INTENSIF, Vol.9 No.2 August 2025
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

Enhancing Multi-Class Classification of Non-Functional Requirements Using a BERT-DBN Hybrid Model

Received:
23 March 2025
Accepted:
8 June 2025
Published:
12 July 2025

[1*]Badzliana Aqmar Suris, [2]Aris Thobirin, [3]Sugiyarto Surono, [4]Mohamed Naeem Antharathara Abdulnazar
[1,2,3]Mathematics,Universitas Ahmad Dahlan,
[4]Dept. Computer Systems, ISMA University, Latvia
E-mail: [1]badzliana2100015016@webmail.uad.ac.id,
[2]aris.thobi@math.uad.ac.id, [3]sugiyarto@math.uad.ac.id,
[4]aamohdnaeem@gmail.com
*Corresponding Author

Background: Software requirements classification is essential to group Non-Functional Requirements (NFR) into several aspects, such as security, usability, performance, and operability. The main challenges in NFR classification are data limitations, text complexity, and high generalization needs. **Objective**: This research seeks to create a classification model using a hybrid of BERT and DBN, optimize hyperparameters, and improve data representation. **Methods**: A BERT and DBN-based approach is used, where DBN enhances BERT's ability to extract hierarchical features. Bayesian Optimization determines the optimal hyperparameters and data augmentation is applied to enrich the dataset variation. The model is tested on the PROMISE dataset consisting of 625 data. **Results**: The BERT-DBN model achieves 95% accuracy on the baseline configuration and 94% on the extensive configuration, better than the previous model, BERT-CNN. The model shows stability without any indication of overfitting. **Conclusion**: The combination of data augmentation, hyperparameter optimization, and DBN's ability to capture hierarchical patterns improves the accuracy of NFR classification, making it more effective than existing methods, and is expected to enhance text-based classification for software requirements.

Keywords—Natural Language Processing; BERT-DBN; Augmentation Data; Bayesian Optimization; Non Functional Requirements

Corresponding Author:
Badzliana Aqmar Suris,
Mathematics,
Universitas Ahmad Dahlan,
Email: badzliana2100015016@webmail.uad.ac.id
Orchid ID: https://orcid.org/0009-0001-5002-5287

INTENSIF, Vol.9 No.2 August 2025
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

I. INTRODUCTION

In the realm of artificial intelligence (AI), natural language processing (NLP) has emerged as a leading discipline focused on text comprehension and analysis [1]. A primary function of NLP involves text classification, which entails organizing text into designated categories [2]. Text classification is crucial in software engineering, particularly in comprehending and methodically prioritizing software requirements. Software requirements are primarily divided into two categories: functional requirements (FR) and non-functional requirements (NFR) [3]. It is essential to classify requirements to ascertain the system's specification behavior.

In previous studies, several approaches have been applied for software requirements classification. Research [4] used the NoRBERT model that adapted BERT for two-class classification, namely FR and NFRs, both multi-class categorization and of NFRs such as usability, security, operability, and performance, with an F1-score reaching 93%. Research [5] developed the FNReq-Net framework that combines traditional feature selection with attention mechanism for two-class classification, namely FR and NFR and NFR subcategories such as scalability and security, which produced an F1-score of up to 91% on the PROMISE and PROMISE-exp datasets. Research [6] compared traditional machine learning algorithms, including SVM, Logistic Regression (LR), Multinomial Naive Bayes (MNB), and kNN for binary classification of FR and NFR. The integration of TF-IDF and LR showed optimal outcomes with an F1-score of 0.91 for FR and NFR classification. Research [7] used word embedding methods such as Word2vec and fastText, with the most significant outcomes attained using fastText, producing an F1 score of 92.8% for FR and NFR classification. In addition, research [8] used the SVM algorithm on the extended PROMISE dataset to classify two classes, FR and NFR, achieving precision and recall of up to 90%.

In the study [9] used the BERT-CNN model for classification, combining BERT for text embedding with CNN to capture local patterns, achieving 88% accuracy. However, this model is limited in capturing global patterns in long texts and has difficulty in balanced representation in categories with limited data. Although previous methods show promising results, NFR classification faces challenges related to restricted data and complex texts. The long and complex text of software requirements makes it difficult for traditional models to comprehensively capture global patterns and relationships between words. To overcome these limitations, this study proposes using Deep Belief Networks (DBN) combined with BERT. BERT uses self-attention to understand the relationship between words in the text. At the same time, DBN can capture hierarchical and non-linear patterns that CNN models cannot capture.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

This method is anticipated to increase the accuracy of NFR classification by fusing DBN's deep feature capture capabilities with BERT's comprehension of text context.

A transformer-based approach called BERT (Bidirectional Encoder Representations from Transformers) is intended to be able to comprehend textual context using [10], both from the left and right sides of words in a sentence [11]. This capability is handy in text analysis that requires deep contextual understanding, such as multi-class classification or long texts. Using a self-attention mechanism, BERT calculates the relationship between words in a single text sequence. This allows for more accurate and efficient semantic representation for complex texts and simultaneously captures syntactic and semantic relationships [12], [13].

A probabilistic generative model made up of several layers of Restricted Boltzmann Machines (RBMs) is called a Deep Belief Network (DBN) designed to extract layer-wise features from input data [14]. DBN can capture hierarchical and non-linear patterns [15], [16], which allows the model to learn from the deep features generated by BERT. This makes DBN an ideal complement to BERT text embedding, especially in dealing with data with complex and non-linear structures. In addition, RBM utilizes a probabilistic energy function to model the joint distribution of input data, making it possible to capture complex and hierarchical patterns [17] from long texts. By integrating the text embedding generated by BERT as input, DBN can learn deep features that other methods, such as CNN, cannot capture. The DBN training process includes two main stages: pre-training using contrastive divergence (CD) for weight initialization and supervised fine-tuning with backpropagation to optimize parameters end-to-end [18].

This research employs data augmentation methods to enhance data quality [19] and ensure adequate representation in each class. The classification results that are produced are greatly influenced by the data that was used to train the model. Results will be improved with a bigger dataset [20]. This study employs Bayesian optimization to identify the optimal hyperparameter. This technique allows efficient exploration of hyperparameters to improve model accuracy and efficiency. This research employs stratified k-fold cross-validation with 10 folds to assess model performance and mitigate overfitting [21].

The purpose of this research is to provide a more precise and trustworthy BERT-DBN-based software requirements classification model by grouping software requirements into four main categories: Operability (O), Performance (PE), Security (SE), and Usability (US). This study utilizes data augmentation techniques to increase data variation and representation, hyperparameter optimization using Bayesian Optimization to achieve the best configuration, and a combination of BERT in representing text context and DBN's ability to capture hierarchical patterns. With this approach, this study is expected to overcome challenges in

INTENSIF, Vol.9 No.2 August 2025
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

software requirements classification, such as data limitations, text complexity, and the need for models with better generalization capabilities.

II. RESEARCH METHOD

This study classifies NFR using the BERT-DBN technique. Figure 1 illustrates the study's phases.

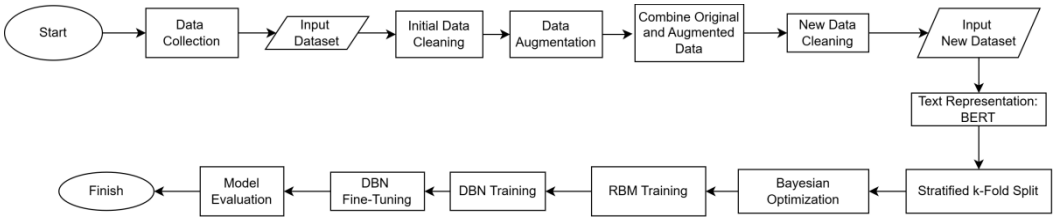


Fig 1. Research Stages Flow

The first step is gathering the study's dataset, sourced via the PROMISE Repository [9], which is available on GitHub. This dataset has an ARFF (Attribute-Relation File Format) format. This dataset consists of 625 data, which include 225 functional requirements with one class and 370 non-functional requirements with 11 classes. Table 1 displays the distribution of classes.

Table 1. Distribution of datasets

NFR_S	Full Name	Amount
A	Availability	21
L	Legal	31
LF	Look and Feel	38
MN/MT	Maintainability/ Maintenance	17
O	Operability	62
PE	Performance	54
SC	Scalability	21
SE	Security	66
US	Usability	67
FT	Fault Tolerance	10
PO	Portability	1
F	Functional	255
Total		625

The data extracted from the dataset is cleaned and ready for further analysis during the preprocessing phase. Initial data cleaning is a crucial step to ensure data quality and consistency. This stage includes converting text to lowercase for consistency, removing excess spaces to tidy up the text, and removing special characters such as punctuation and irrelevant symbols [22].

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

After that, filtration is carried out to ensure that only data from four main classes, namely, "O" (Operability), "PE" (Performance), "US" (Usability), and "SE" (Security). Therefore, a filtration process is carried out to ensure that only data with these classes is used so that the analysis can be more focused on the primary needs.

After filtration, the next stage is to perform data augmentation to increase text variation and overcome the problem of data imbalance in each class. In Table 1, it can be seen that the initial dataset has a varying amount of data in each class. Although there is data from various courses, this study focuses on four main classes. Limited data for the four primary classes may make it more difficult for the model to identify important patterns, which might lower classification accuracy. To overcome this, data augmentation uses the back translation technique, whereby new phrase variants with the same meaning are created by translating the text into another language and then back into the original language. A model that converts text from English to German and back again is used in this study's back translation approach. This process helps create sentence structure variations without changing the original text's meaning. In addition, the paraphrasing technique is also used, which produces new text with a different sentence structure but still maintains its meaning. These augmentation techniques aim to enrich the dataset with a wider variety of text, thereby increasing the representation of each class and helping the model better generalize existing patterns [23].

After the data is augmented, the next stage is text representation using BERT. In this stage, the text in the dataset is transformed into a vector representation using the BERT model [24], [25] implemented through the Hugging Face Transformers library. This model utilizes tokenizers and BERT models such as bert-base-uncased and bert-large-uncased to generate contextual text representations, considering the relationship between words from the left and right sides of a sentence. Masked Language Model (MLM) and Next Sentence Prediction (NSP) are two key methods used by BERT that enable the model to comprehend the link between words and sentences [26], [27]. BERT determines a sentence's word relationships via a self-attention technique. Mathematically, self-attention is calculated with the following formula [28]:

$$Attention\,(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Q is the query, K is the key, V is the value, and d_k is the dimension of the key. This study uses two BERT configurations, namely BERT-Base with 12 transformer layers with a hidden layer size of 768 dimensions [29], so that the total parameters it has are 110 million, and BERT-Large with 24 transformer layers with a hidden layer size of 1024 dimensions [30], which results in a total of 340 million parameters.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

Using stratified 10-fold cross validation, the enhanced dataset is then divided. The augmented dataset of 574 data is divided into 10 balanced folds. Each fold contains approximately 10% of the data taken proportionally from each class. This method guarantees that the class distribution for each fold is comparable to that of the original dataset [31] so the model can be trained and tested fairly in each iteration. In each iteration, nine folds are used for training, while one is used for validation. This process is repeated 10 times, with each data used as validation data once and training nine times. This process is done to reduce the risk of overfitting [32] and ensure that the model is tested on various subsets of the data. Stratified cross-validation uses the following formula to divide the data:

$$n_{fold} = \frac{N}{k} \tag{2}$$

Where N is the total number of data, and k is the number of folds.

Stratified 10-fold cross-validation is followed by Bayesian optimization to identify the ideal set of hyperparameters to improve model performance [33]. Bayesian Optimization is a probability-based optimization method that iteratively evaluates new combinations of hyperparameters by building a surrogate model, such as the Gaussian Process (GP), that models the relationship between hyperparameters and model performance [34]. This process involves an acquisition function, such as Expected Improvement (EI), to select the next hyperparameters to be tested. The EI function is calculated using the formula:

$$EI(x) = \big(f(x^+) - \mu(x)\big)\Phi(Z) + \sigma(x)\phi(Z) \tag{3}$$

Where $f(x^+)$ is the best performance value found, $\mu(x)$ is the predicted performance of the surrogate model on hyperparameter x, σ(x) is the uncertainty of the prediction, $Z = \frac{f(x^+)-\mu(x)}{\sigma(x)}$, and $\Phi(Z)$ and $\phi(Z)$ are the cumulative distribution function and standard normal probability density. The best combination of hyperparameters is obtained by maximizing the EI value so that the model can be trained using the optimal configuration found [35]. In this study, In order to train the Deep Belief Network (DBN) model on each fold of the Stratified Cross-Validation findings, Bayesian Optimization is used to identify the ideal hyperparameters, including learning rate, batch size, number of epochs, and epochs per RBM.

Next, the model will be trained using DBN, or the Deep Belief Network approach after obtaining the best hyperparameters through Bayesian Optimization. As a generative model, DBN [36] consisting of several stacked Restricted Boltzmann Machine (RBM) layers. RBM is a two-layer generative neural network [37] consisting of a visible layer, v, and a hidden layer, h. Each RBM is tasked with capturing deeper feature representations of the data. Overall, the DBN architecture forms a hierarchical structure that composes RBM layers, where each RBM layer can capture increasingly complex features in the input data.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

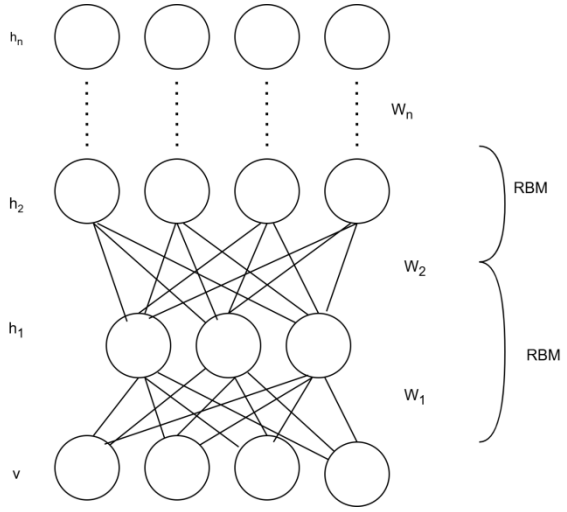


Fig. 2. Deep Belief Network Architecture

Figure 2 illustrates the DBN architecture, showing how the RBM layers are interconnected and interact. The DBN architecture shows the RBM layers tasked with hierarchically extracting features from the input data. Each hidden layer of one RBM becomes the input for the hidden layer of the next RBM, creating deeper representations at each network level. This allows the model to learn more complex representations from unstructured data. For the implementation in this study, the PyTorch library was used to define the RBM and DBN models and optimize the model parameters using the Stochastic Gradient Descent (SGD) algorithm. This model uses energy distribution as the basis for its representation, with the energy function defined as [38]:

$$E(v,h) = -\sum_i v_i b_i - \sum_j h_j c_j - \sum_{i,j} v_i h_j w_{ij} \quad (4)$$

The joint probability between v and h is determined by the Boltzmann distribution function [39]:

$$P(v,h) = \frac{e^{E(v,h)}}{Z}, \quad Z = \sum_{v,h} e^{E(v,h)} \quad (5)$$

The DBN training process is carried out layer-wise. In the initial training stage, each RBM layer is trained unsupervised [40], where the model learns to extract deeper features from unlabeled input data. However, in the fine-tuning stage, training is carried out supervised [41], and the backpropagation method [42] is used to optimize each DBN layer to reduce prediction errors. In the fine-tuning stage, Cross-entropy loss is used to measure prediction errors, which is calculated using the following formula:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (6)$$

Where y_i is the actual label, and $\hat{y}$ is the probability of prediction by the model. This loss function is designed to minimize the prediction error by penalizing predictions far from the actual value. By combining supervised fine-tuning with unsupervised pre-training, DBN can use

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

BERT's text embedding to efficiently capture non-linear and hierarchical patterns, resulting in a classification model with improved accuracy and dependability.

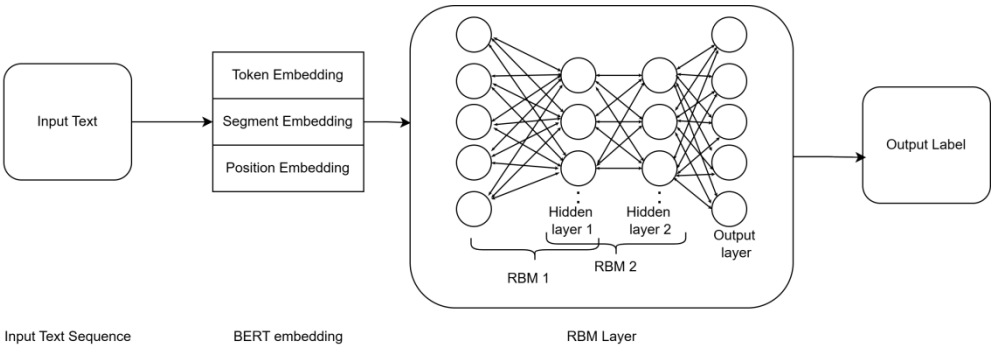


Fig 3. BERT-DBN Architecture

The BERT-DBN model's architecture, which combines DBN's capacity to learn hierarchical and non-linear patterns with BERT's capacity to produce context-based text representations, is shown in Figure 3. The text input is first processed through BERT, which produces three types of embeddings: token embeddings to represent words or subwords, segment embeddings to indicate relationships between sentences, and position embeddings to include information about the order of words in a sentence. These three embeddings are then used as input to the DBN model, which consists of two RBM layers. The first RBM captures the initial patterns of the data, while the second RBM learns more complex features. To produce a class prediction O, PE, SE, or US, the output layer receives the final output from the hidden layer. This architecture makes use of the advantages of both strategies to increase classification accuracy: BERT's comprehension of the text's context and DBN's capacity to identify more intricate patterns

After that, a model assessment is carried out to gauge how well DBN performs in categorizing the four primary NFR classes. This review uses the confusion matrix to obtain many key measures, including accuracy, precision, recall, and F1-Score. The four components of the confusion matrix are False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN).

$$\text{Precision}_{\text{k}} = \frac{TP_k}{TP_k + FP_k} \tag{7}$$

$$\text{Recall}_{\text{k}} = \frac{TP_k}{TP_k + FN_k} \tag{8}$$

$$\text{F1} - \text{Score} = 2.\frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

$$\text{Accuracy} = \frac{\sum_k TP_k}{\sum_k (TP_k + FP_k + FN_k + TN_k)} \tag{10}$$

III. RESULT AND DISCUSSION

The performance of the classification model on NFRs was evaluated by grouping NFRs into four main classes: O, PE, SE, and US. The model used is BERT-DBN, with two main

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

configurations: base and large. Before performing the classification, the researcher carried out a data augmentation process to increase data variation and overcomeances by using back translation and paraphrasing techniques. The distribution of data before and after augmentation is shown in Table 2, which shows an increase in total data from 249 to 574 data, allowing the model to learn patterns better.

Table 2. Distribution after augmentation

Class	Before augmentation	After augmentation
O	62	130
PE	54	128
SE	66	154
US	67	162
Total	249	574

Stratified 10-fold cross-validation was used for model assessment to maintain the balance of the class distribution for each fold. As a result, there is less chance of overfitting, and the model learns patterns more effectively. The model was tuned using Bayesian Optimization to get the optimal set of hyperparameters that yielded the highest performance during cross-validation. Table 3 displays the ideal hyperparameters.

Table 3. Hyperparameter optimal

Konfigurasi	Hyperparameter Optimal
Base	(lr = 0.001996 , batch_size = 62, epoch= 50, epoch_per_rbm = 22)
Large	(lr = 0.005874, batch_size = 111, epoch =19, epoch_per_rbm = 19)

These optimal hyperparameters are used to train the model using a Deep Belief Network (DBN), which can effectively leverage text embedding from BERT to learn hierarchical and non-linear patterns. In each cross-validation iteration, DBN is trained with nine folds as training data, while the remaining one fold is used as validation data.

The confusion matrix, which shows the number of right and wrong predictions for each class in the Base and Large model configurations, depicts the model performance assessment findings. Figure 4(a) displays the confusion matrix results for the BERT-DBN model with the Base configuration. However, Figure 3(b) displays the results for the BERT-DBN model with the Large configuration.

In Figure 4(a) with the BERT-DBN base model, it can be seen that the model successfully predicted the US class with a total of TP of 157 out of a total of 162 data in that class, which shows perfect accuracy in this class. However, some prediction errors in other courses, such as class O, had 4 data misclassified into the PE class. Meanwhile, in the BERT-DBN Large model,

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

shown in Figure 4(b), the prediction results show a total TP of 152 for the US class, slightly lower than the base configuration.

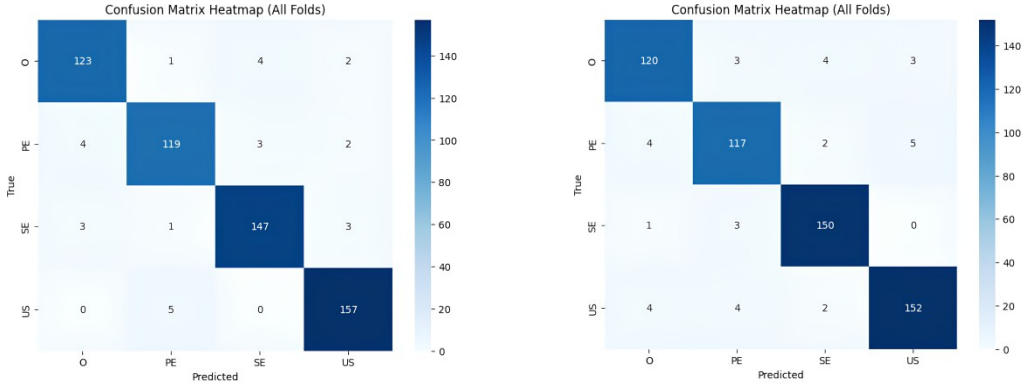
Based on Figure 5(a-d), which shows the accuracy and loss graphs of the BERT-DBN Base and Large models, it is evident that the validation and training accuracies increase consistently as the number of epochs increases until they reach stability at the end of training. The loss values for training and validation also show a significant downward trend, with a slight difference. This indicates that the BERT-DBN Base and Large models can learn well without indicating overfitting or underfitting.

Figure 6 compares the performance between the BERT-CNN model from previous studies and the proposed model, BERT-DBN, in the Base and Large configurations. The BERT-CNN model with the Base configuration shows promising results in all classes, with the highest F1-Score in the SE class, 0.87, and PE, 0.86. However, in the US class, although it has the highest Precision, which is 0.85, the Recall is lower. In the Large configuration, although the F1-Score for the US increases to 0.87, there is a decrease in the PE class, where Precision and Recall decrease. Meanwhile, BERT-DBN gives better results in both configurations. In the Base configuration, BERT-DBN records high Precision and Recall, which is 0.95 for class O, with an F1-Score of 0.95, and high F1-Scores for the PE, SE, and US classes. The Large configuration shows a slight decrease in Precision for class O at 0.93 and PE at 0.91.

Figure 7 illustrates the precision of the comparison between the BERT-CNN and BERT-DBN models, and it can be seen that BERT-DBN is superior in both configurations. In Base, BERT-DBN achieves an accuracy of 0.95, much higher than BERT-CNN, which only records 0.85. In the Large configuration, although there is a slight decrease in BERT-DBN (0.94), this model is still better than BERT-CNN, which only achieves an accuracy of 0.87. This further confirms the superiority of BERT-DBN in terms of accuracy compared to BERT-CNN, both in the Base and Large configurations.

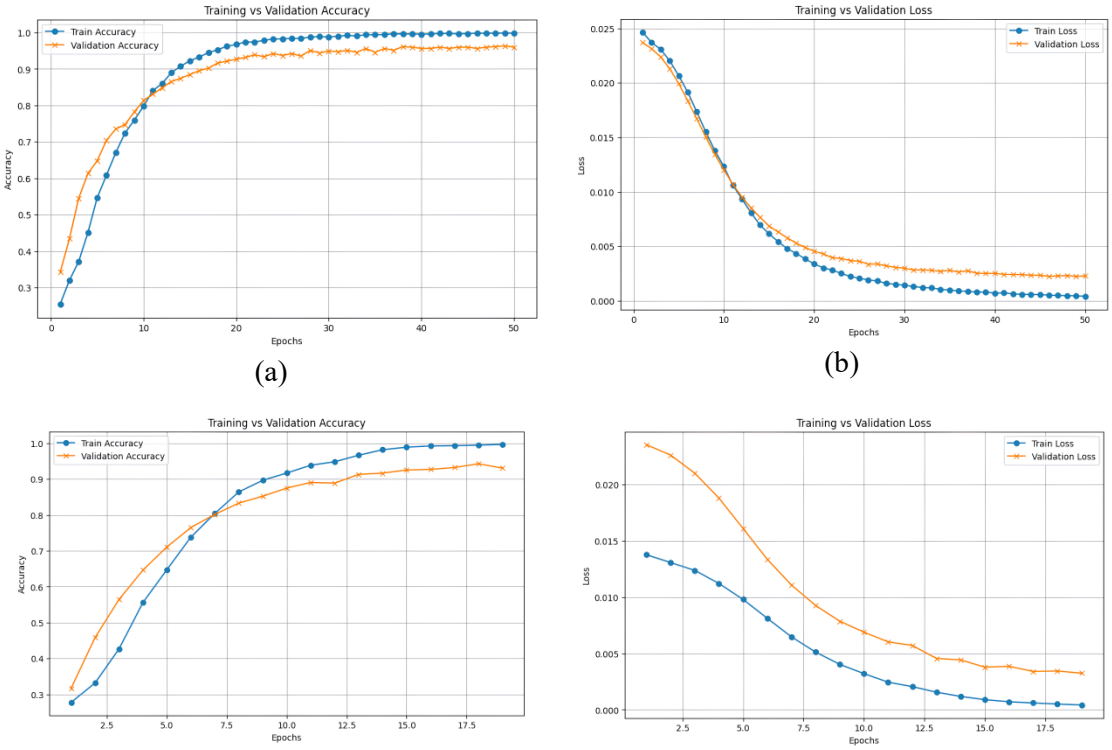
This performance improvement can be attributed to several main factors. First, data augmentation has successfully increased the data variation, thus helping the model learn patterns better. Second, applying Bayesian Optimization allows the model to find the optimal set of hyperparameters, including batch size, number of epochs, learning rate, and epochs per RBM. These optimal hyperparameters ensure the model can be trained effectively, resulting in higher classification performance. In addition, the ability of DBN to capture hierarchical and non-linear patterns in BERT text embeddings significantly contributes to more accurate and stable results. With the combination of data augmentation, hyperparameter optimization, and the advantages of DBN architecture, the proposed model can provide more precise and balanced classification results compared to previous approaches.

INTENSIF, Vol.9 No.2 August 2025
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637



(a) (b)

Fig 4. (a) Confusion matrix results of BERT-DBN Base configuration, (b) Confusion matrix results of BERT-DBN Large configuration.



(a) (b)

(c) (d)

Fig 5. (a) BERT − DBN_{base} accuracy graph , (b) BERT − DBN_{base} loss graph, (c) BERT − DBN_{large} accuracy graph, (d) BERT − DBN_{large} loss graph

INTENSIF, Vol.9 No.2 August 2025
ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

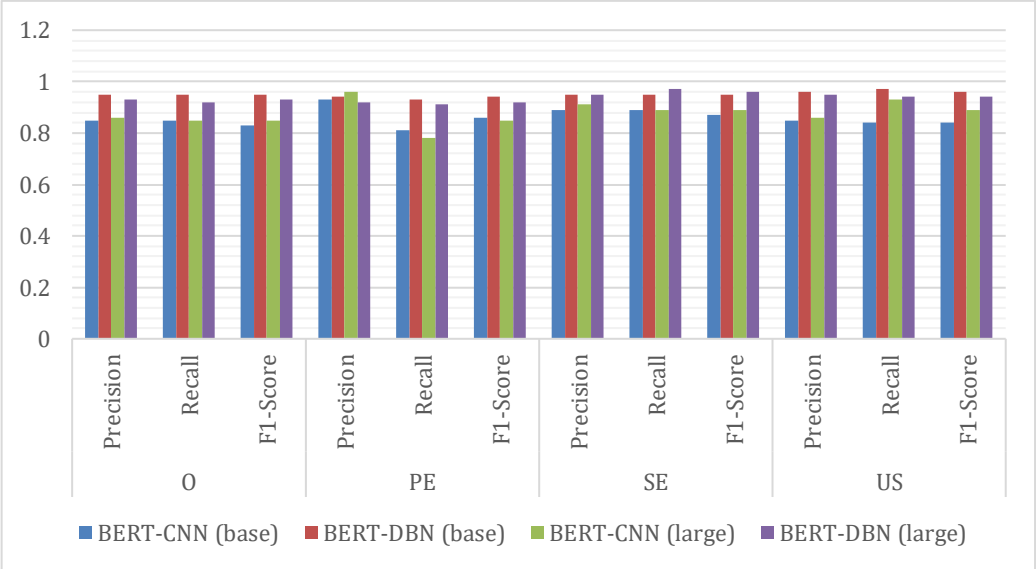


Fig 6. Performance Comparison of BERT-CNN and BERT-DBN Models

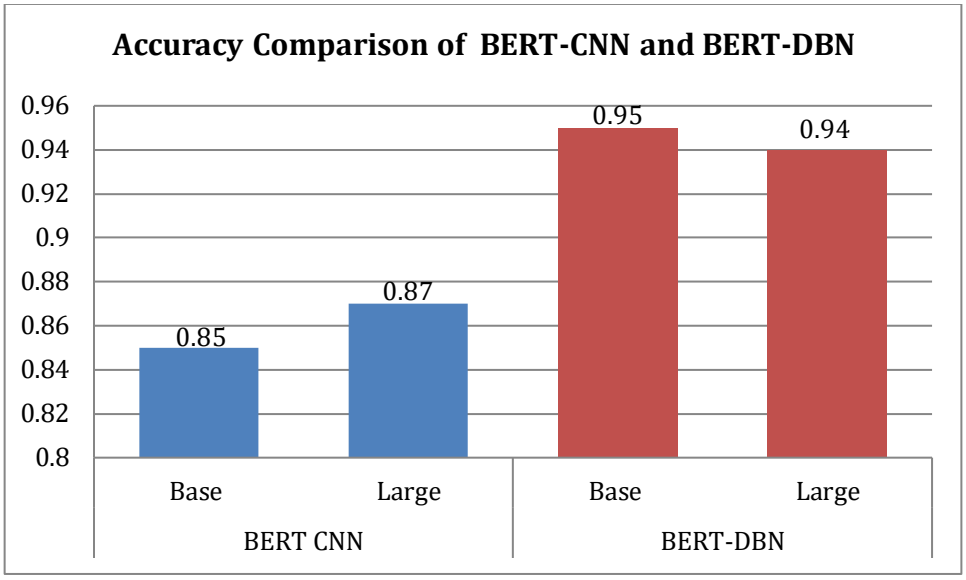


Fig 7. Comparison of BERT-CNN and BERT-DBN Model Accuracy

IV. CONCLUSION

The proposed BERT-DBN model achieved an overall accuracy of 95%, a significant improvement over the BERT-CNN model, which only achieved 85% in the Base configuration. The model also showed excellent precision and recall across classes, with F1 scores reaching 0.95 for class O and 0.96 for class US, while the BERT-CNN model only achieved 0.85 and 0.87 for the same courses. Data augmentation and Bayesian Optimization also contributed significantly to the performance improvement, with optimal hyperparameters allowing the model to work more efficiently. The Base configuration of the BERT-DBN model showed more

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

stable and consistent performance than the Large configuration, with a slightly higher accuracy value of 0.95 compared to 0.94 in the Large configuration.

However, this study has some limitations, such as the limited dataset size and only focus on four major NFR classes, which may affect the model's generalization. For further studies, Checking out the application is advised of this model to larger and more diverse datasets and expand the scope of NFR classes to see how the model can adapt to additional complexity. Further research can also explore the use of other hybrid architectures, including the application of different methods in data optimization and hyperparameters to improve model performance in the future.

Author Contributions: *Badzliana Aqmar Suris*: Conceptualization, Methodology, Data Curation, Software, Writing - Original Draft, Writing-Review & Editting. *Aris Thobirin*: Supervision, Writing - Review & Editing. *Sugiyarto Surono*: Investigation, Supervision, Conseptualization. *Mohamed Naeem Antharathara Abdulnazar*: Review & Editing

All authors have read and agreed to the published version of the manuscript.

Funding: This research received no specific grant from any funding agency

Acknowledgments: The authors sincerely credit FAST Ahmad Dahlan University Indonesia, particularly the Mathematic Laboratory.

Conflicts of Interest: This research received no specific grant from any funding agency.

Data Availability: The data cannot be openly shared for the protection of study participant privacy.

Informed Consent: There were no human subjects.

Animal Subjects: There were no animal subjects.

ORCID:
Badzliana Aqmar Suris : https://orcid.org/0009-0001-5002-5287
Aris Thobirin : https://orcid.org/0009-0006-9823-4746
Sugiyarto Surono : https://orcid.org/0000-0001-6210-7258
Mohamed Naeem Antharathara Abdulnazar: https://orcid.org/0009-0008-1704-5287

REFERENCES

[1] A. Badawi, “the Effectiveness of Natural Language Processing (Nlp) As a Processing Solution and Semantic Improvement,” *Int. J. Econ. Technol. Soc. Sci.*, vol. 2, no. 1, pp. 36–44, 2021, **doi:** 10.53695/injects.v2i1.194.

[2] V. Dogra *et al.*, “A Complete Process of Text Classification System Using State-of-the-Art NLP Models,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, **doi:** 10.1155/2022/1883698.

[3] A. Jarzebowicz and P. Weichbroth, “A Qualitative Study on Non-Functional

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

Requirements in Agile Software Development," *IEEE Access*, vol. 9, pp. 40458–40475, 2021, **doi:** 10.1109/ACCESS.2021.3064424.

[4] T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, "NoRBERT: Transfer Learning for Requirements Classification," *Proc. IEEE Int. Conf. Requir. Eng.*, vol. 2020-Augus, pp. 169–179, 2020, **doi:** 10.1109/RE48521.2020.00028.

[5] S. Saleem, M. N. Asim, L. Van Elst, and A. Dengel, "FNReq-Net: A hybrid computational framework for functional and non-functional requirements classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 8, p. 101665, 2023, **doi:** 10.1016/j.jksuci.2023.101665.

[6] E. D. Canedo and B. C. Mendes, "Software requirements classification using machine learning algorithms," *Entropy*, vol. 22, no. 9, 2020, **doi:** 10.3390/E22091057.

[7] S. Tiun, U. A. Mokhtar, S. H. Bakar, and S. Saad, "Classification of functional and non-functional requirement in software requirement using Word2vec and fast Text," *J. Phys. Conf. Ser.*, vol. 1529, no. 4, 2020, **doi:** 10.1088/1742-6596/1529/4/042077.

[8] R. K. Gnanasekaran, S. Chakraborty, J. Dehlinger, and L. Deng, "Using recurrent neural networks for classification of natural language-based non-functional requirements," *CEUR Workshop Proc.*, vol. 2857, 2021.

[9] K. Kaur and P. Kaur, "BERT-CNN: Improving BERT for Requirements Classification International Conference on Machine Learning and Data Engineering using CNN," *Procedia Comput. Sci.*, vol. 218, no. 2022, pp. 2604–2611, 2023, **doi:** 10.1016/j.procs.2023.01.234.

[10] J. Joo, "International Journal of Medical Informatics Predicting medical specialty from text based on a domain-specific," vol. 170, no. December 2022, 2023, **doi:** 10.1016/j.ijmedinf.2022.104956.

[11] B. Yang, B. Zhang, K. Cutsforth, S. Yu, and X. Yu, "Emerging industry classification based on BERT model," *Inf. Syst.*, vol. 128, no. October 2024, p. 102484, 2025, **doi:** 10.1016/j.is.2024.102484.

[12] M. Escarda, C. Eiras-Franco, B. Cancela, B. Guijarro-Berdiñas, and A. Alonso-Betanzos, "Performance and sustainability of BERT derivatives in dyadic data," *Expert Syst. Appl.*, vol. 262, no. October 2023, p. 125647, 2025, **doi:** 10.1016/j.eswa.2024.125647.

[13] D. Wu, J. Yang, and K. Wang, "Exploring the reversal curse and other deductive logical reasoning in BERT and GPT-based large language models," *Patterns*, p. 101030, 2024, **doi:** 10.1016/j.patter.2024.101030.

[14] M. A. Hamza *et al.*, "Computational Linguistics with Optimal Deep Belief Network Based Irony Detection in Social Media," 2023, **doi:** 10.32604/cmc.2023.035237.

[15] A. Loganthan, "DeepSecure watermarking : Hybrid Attention on Attention Net and Deep Belief Net based robust video authentication using Quaternion Curvelet Transform domain," *Egypt. Informatics J.*, vol. 27, no. July, p. 100514, 2024, **doi:** 10.1016/j.eij.2024.100514.

[16] B. K. Sethi, D. Singh, S. K. Rout, and S. K. Panda, "Long Short-Term Memory-Deep Belief Network-Based Gene Expression Data Analysis for Prostate Cancer Detection and Classification," vol. 12, no. November 2023, 2024, **doi:** 10.1109/ACCESS.2023.3346925.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

[17] N. Mahendran and D. R. Vincent P M, "Deep belief network-based approach for detecting Alzheimer's disease using the multi-omics data," *Comput. Struct. Biotechnol. J.*, vol. 21, pp. 1651–1660, 2023, **doi:** 10.1016/j.csbj.2023.02.021.

[18] A. K. Shukla and P. K. Muhuri, "Deep belief network with fuzzy parameters and its membership function sensitivity analysis," *Neurocomputing*, vol. 614, no. November 2022, p. 128716, 2025, **doi:** 10.1016/j.neucom.2024.128716.

[19] X. Li, W. Zhang, and Q. Ding, "Intelligent rotating machinery fault diagnosis based on deep learning using data augmentation," *J. Intell. Manuf.*, vol. 31, no. 2, pp. 433–452, 2020, **doi:** 10.1007/s10845-018-1456-1.

[20] Sugiyarto, J. Eliyanto, N. Irsalinda, and M. Fitrianawati, "Fuzzy sentiment analysis using convolutional neural network," *AIP Conf. Proc.*, vol. 2329, no. February, 2021, **doi:** 10.1063/5.0042144.

[21] S. Ünalan, O. Günay, I. Akkurt, K. Gunoglu, and H. O. Tekin, "Journal of Radiation Research and Applied Sciences A comparative study on breast cancer classification with stratified shuffle split and K-fold cross validation via ensembled machine learning," *J. Radiat. Res. Appl. Sci.*, vol. 17, no. 4, p. 101080, 2024, **doi:** 10.1016/j.jrras.2024.101080.

[22] A. Kousar *et al.*, "MLHS-CGCapNet: A Lightweight Model for Multilingual Hate Speech Detection," *IEEE Access*, vol. 12, no. August, pp. 106631–106644, 2024, **doi:** 10.1109/ACCESS.2024.3434664.

[23] M. Akram *et al.*, "Uncertainty-aware diabetic retinopathy detection using deep learning enhanced by Bayesian approaches," *Sci. Rep.*, vol. 15, no. 1, p. 1342, 2025, **doi:** 10.1038/s41598-024-84478-x.

[24] A. Hossein, S. Das, J. Liu, and M. A. Rahman, "Using Bidirectional Encoder Representations from Transformers (BERT) to classify traffic crash severity types," *Nat. Lang. Process. J.*, vol. 3, no. April, p. 100007, 2023, **doi:** 10.1016/j.nlp.2023.100007.

[25] A. Subakti, H. Murfi, and N. Hariadi, "The performance of BERT as data representation of text clustering," *J. Big Data*, 2022, **doi:** 10.1186/s40537-022-00564-9.

[26] Y. Kim *et al.*, "OPEN A pre - trained BERT for Korean medical natural language processing," *Sci. Rep.*, pp. 1–10, 2022, **doi:** 10.1038/s41598-022-17806-8.

[27] H. Kang, S. Goo, H. Lee, J. Chae, and H. Yun, "Fine-tuning of BERT Model to Accurately Predict Drug – Target Interactions," pp. 1–15, 2022, **doi:** 10.3390/pharmaceutics14081710.

[28] D. Argade, V. Khairnar, D. Vora, S. Patil, K. Kotecha, and S. Alfarhood, "Multimodal Abstractive Summarization using bidirectional encoder representations from transformers with attention mechanism," *Heliyon*, vol. 10, no. 4, p. e26162, 2024, **doi:** 10.1016/j.heliyon.2024.e26162.

[29] S. Kula, R. Kozik, and M. Choraś, "Implementation of the BERT-derived architectures to tackle disinformation challenges," *Neural Comput. Appl.*, vol. 34, no. 23, pp. 20449–20461, 2022, **doi:** 10.1007/s00521-021-06276-0.

[30] N. Q. K. Le, Q. T. Ho, T. T. D. Nguyen, and Y. Y. Ou, "A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information," *Brief. Bioinform.*, vol. 22, no. 5, pp. 1–7, 2021, **doi:** 10.1093/bib/bbab005.

ISSN: 2580-409X (Print) / 2549-6824 (Online)
DOI: https://doi.org/10.29407/intensif.v9i2.24637

[31] M. T R, V. K. V, D. K. V, O. Geman, M. Margala, and M. Guduri, "The stratified K-folds cross-validation and class-balancing methods with high-performance ensemble classifiers for breast cancer classification," *Healthc. Anal.*, vol. 4, no. July, p. 100247, 2023, **doi:** 10.1016/j.health.2023.100247.

[32] N. F. Idris, M. A. Ismail, M. I. M. Jaya, A. O. Ibrahim, A. W. Abulfaraj, and F. Binzagr, "Stacking with Recursive Feature Elimination-Isolation Forest for classification of diabetes mellitus," *PLoS One*, vol. 19, no. 5 May, pp. 1–18, 2024, **doi:** 10.1371/journal.pone.0302595.

[33] S. Surono, M. Yahya Firza Afitian, A. Setyawan, D. K. E. Arofah, and A. Thobirin, "Comparison of CNN Classification Model using Machine Learning with Bayesian Optimizer," *HighTech Innov. J.*, vol. 4, no. 3, pp. 531–542, 2023, **doi:** 10.28991/HIJ-2023-04-03-05.

[34] D. Xing, W. Chen, C. Tatsuoka, and X. Lu, "Ba-ZebraConf: A Three-Dimension Bayesian Framework for Efficient System Troubleshooting," pp. 1–12, 2024, **doi:** 10.48550/arXiv.2412.11073.

[35] G. T. Ribeiro, J. G. Sauer, N. Fraccanabbia, V. C. Mariani, and L. dos Santos Coelho, "Bayesian optimized echo state network applied to short-term load forecasting," *Energies*, vol. 13, no. 9, 2020, **doi:** 10.3390/en13092390.

[36] G. Coulson and D. Ferrari, *Context-Aware Systems and Applications , and Nature of Computation and Communication*. 2020.

[37] A. K. Shukla and P. K. Muhuri, "A novel deep belief network architecture with interval type-2 fuzzy set based uncertain parameters towards enhanced learning," *Fuzzy Sets Syst.*, vol. 477, no. October 2023, p. 108744, 2024, **doi:** 10.1016/j.fss.2023.108744.

[38] J. Tang, J. Wu, and J. Qing, "A feature learning method for rotating machinery fault diagnosis via mixed pooling deep belief network and wavelet transform," *Results Phys.*, vol. 39, no. May, p. 105781, 2022, **doi:** 10.1016/j.rinp.2022.105781.

[39] A. Punitha and V. Geetha, "Automated climate prediction using pelican optimization based hybrid deep belief network for Smart Agriculture," *Meas. Sensors*, vol. 27, no. February, p. 100714, 2023, **doi:** 10.1016/j.measen.2023.100714.

[40] R. Savitha, A. Ambikapathi, and K. Rajaraman, "Online RBM : Growing Restricted Boltzmann Machine on the fly for unsupervised representation," *Appl. Soft Comput. J.*, vol. 92, p. 106278, 2020, **doi:** 10.1016/j.asoc.2020.106278.

[41] J. Pei *et al.*, "Two-Phase Virtual Network Function Selection and Chaining Algorithm Based on Deep Learning in SDN / NFV-Enabled Networks," vol. 38, no. 6, pp. 1102–1117, 2020, **doi:** 10.1109/JSAC.2020.2986592.

[42] A. Albasir, Q. Hu, K. Naik, and N. Naik, "Unsupervised detection of security threats in cyber- physical system and IoT devices based on power fin- gerprints and RBM autoencoders," pp. 1–25, 2021, **doi:** 10.20517/jsss.2020.19.