International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *30 October 2026*

BERT-CGAE: A Hybrid BERT-based CNN-GRU-Attention Ensemble Framework for Text Classification

D.Savitha M.Sc., M.Phil.,

Research Scholar, Asst. Professor of Information Technology,

Vellalar College for Women (Autonomous), Erode – 12 savitha.d@vcw.ac.in

Dr. L. Sudha M.Sc., M.C.A., M.Phil., Ph.D.,

Research Supervisor, Associate Professor, School of Computer Science, VET Institute of Arts and Science College, Erode – 12 l.sudhaphd@gmail.com

Abstract

Text classification is a fundamental task in natural language processing, requiring models to accurately interpret semantic and syntactic patterns in textual data. The proposed framework, BERT-EnsemText, introduces a hybrid approach that combines advanced feature extraction with ensemble learning for robust and accurate classification. Existing methods often rely on static word embeddings or single-model architectures, which struggle to capture contextual nuances and tend to overfit on complex or domain-specific datasets. These approaches also lack adaptive mechanisms to leverage diverse decision boundaries effectively. To address these limitations, BERT-EnsemText integrates BERT-based contextual embeddings with a CNN-GRU-Attention feature extractor to jointly capture local, sequential, and salient contextual features. The extracted features are processed through multiple base classifiers including GEPSVM, XGBoost, and Logistic Regression. An attention-based meta-learner dynamically weights classifier outputs using confidence scores, enabling adaptive and robust ensemble decisions. This framework can be applied to a wide range of text classification tasks such as sentiment analysis, topic categorization, and intent detection, especially in domains with complex language patterns. Experimental results show that BERT-EnsemText significantly improves classification accuracy, generalization, and robustness compared to conventional single-model and static embedding-based baselines.

Keywords: BERT embeddings, CNN-GRU-Attention, ensemble learning, text classification, meta-learning, hybrid framework

1. Introduction

Text classification has evolved into a paradigm in current Natural Language Processing (NLP), serving as the foundation for a broad range of applications [1]. Including sentiment analysis, spam detection, topic classification, recommendation systems, and opinion mining [2]. Asonline content in the form of social networking site posts, product reviews, and news stories continues to grow exponentially [3]. The demand has increased to have systems that can process and analyze large volumes of text content effectively [4]. Proper classification will help businesses understand consumer preferences, governments track public opinion, and researchers study political trends in society [5]. In addition to its academic importance, it serves as an important enabler of more practical applications, including healthcare sentiment detection, election opinion mining, and automated extraction of legal documents [6]. As a result, there is a need to develop valid and scalable classification systems to ensure actionable information in increasingly sophisticated textual data [7].

1.2 Limitations of Single-Model Approaches.

Although traditional machine learning and single-model deep learning methods have made some contributions to NLP development, they have several limitations [8]. Classical techniques, such as Support Vector Machines (SVM) or Naive Bayes, rely on fixed word embeddings, which do not capture contextual peculiarities and semantic differences between domains [9]. On the same note, deep learning algorithms like Convolutional Neural Networks (CNNs) address local n-gram patterns but require long-range dependencies [10]. GRU and LSTM are recurrent architectures that

are sensitive to sequential context, although they have weaknesses, including vanishing gradients, prohibitive training costs, and reduced performance on massive datasets [11]. Moreover, the use of a single model predisposes such systems to overfitting, diminishes their ability to extrapolate to unknown areas, and limits the explanation of classification judgments [12]. Such constraints highlight the limitations of stand-alone architectures in addressing the diverse range of problems associated with large-scale, noisy, and domain-specific text data [13].

1.3 Need for Hybrid and Ensemble Architectures.

To address the weaknesses of single-model systems, scientists are increasingly focusing on hybrid and ensemble models that combine the strengths of different models [14]. Hybrid architectures offer a feature representation that is more expressive and detailed, utilizing CNNs to extract local features, GRUs to model sequences, and attention to focus on salient features [15]. Similar methods to those used in ensemble strategies, including the use of advanced classifiers like Generalized Eigenvalue Proximal Support Vector Machine (GEPSVM), XGBoost, or Logistic Regression, are also more effective in decision-making with multiple decision boundaries [16]. Also, meta-learners that operate with attention can dynamically balance the output of a classifier to enhance strength and adaptability. These hybrid-ensemble models offer the benefits of promoting classification accuracy, as well as scalability, interpretability, and cross-domain generalization; thus, they are applicable in real-life applications [17]. Therefore, the creation of a robust hybrid ensemble model such as BERT-CGAE is an appropriate and timely contribution to the development of text classification studies.

2. Conceptual Foundations

The massive proliferation of textual information on online platforms has increased the need for accurate and efficient text classification. Conventional approaches often struggle with providing a sense of context and analyzing features. Deep learning models, which combine CNNs, GRUs, and high-performance classifiers, are also robust and yield better results in sentiment analysis and decision-making problems.

Recent studies have given EPLF-TSVM-DL, a combination of long pinball loss twin SVM and deep learning, the ability to predict sentence-level sentiment polarity. It focuses on preprocessing, embeddings, CNN extraction, and EPLF-TSVM classification. The findings boast better convergence, reduced complexity, and improved metrics such as accuracy and stability, positioning the method as an efficient binary sentiment task solver for noisy social media data. However, scalability issues remain [18].

CNN-GRU-GEPSVM is a combination of CNN and GRU, supported by a more powerful attention system and a GEPSVM classifier, utilizing GloVe embeddings. This mixture learns patterns of phrasings in the locality as well as long-range dependencies and dynamically weighted salient terms. Experiments have shown a 6.8% increase in accuracy, along with equal increases in precision, recall, and F1-score across benchmarks, indicating that the system is robust across the product review and health sentiment analysis domains [19].

In a brief overview, traditional encodings are replaced by deep contextual models, with the data flow described as starting with raw text and then proceeding to labels. It compares the older pipelines with those of the transformer age, with a focus on the description of preprocessing that has been omitted. The article provides an overview of data sets and presents two novel multilabel datasets, which overcome the scarcity. It offers experiment snapshots and frames open challenges towards having standardised preprocessing and evaluation practices [20].

A recent systematic review examines the applications of semi-supervised learning to text classification, reviewing 1794 works and selecting 157 for further analysis. It describes domains, datasets, languages, representations, and algorithms, classifying work in a modern taxonomy of SSL. The review interprets labeled-data ratios, assessment measures, and performance trends coherently, concluding with limitations and future research directions, thereby closing a major gap in the SSL literature [21].

A PRISMA-based survey was conducted on 224 machine-learning-based text classification studies (2003-2022), comparing datasets, models, accuracy, metrics, and protocols. It gathers the evaluation discrepancies, emphasizes the problems of reproducibility, fragmentation of the data sets, and methodological flaws, and suggests that standardized benchmarking, cross-dataset testing, transparent reportage of the metrics, and unrestricted release of data/code should be implemented to make the field more comparable, cumulative, and empirical in its foundations [22].

It is a long-term survey that creates a taxonomy of both traditional and deep learning methods of text classification, mapping models, feature extraction, and benchmark datasets. It discusses technical advancements and assessment indicators, comparing the strengths and weaknesses. The synthesis of historical developments, the classification of taxonomic groups, and the description of future research perspectives make the work an indispensable source of knowledge, both in terms of background knowledge and the current model choice [23].

The general overview is a synthesis of state-of-the-art machine and deep learning methods for text mining, encompassing supervised, semi-supervised, and unsupervised paradigms. It also gives the strengths and weaknesses of algorithms in subtasks and emphasizes the practical aspects of implementation. The paper summarizes the literature and methodological specifications, defining areas where further progress can be achieved and recommending directions for creating new model proposals relevant to various NLP tasks and large text corpora [24].

TextConvoNet proposes a two-dimensional multi-scale CNN that encodes n-grams both intra-and intersententially using an alternative input matrix and 2D convolutions. Applied to five data sets and five metrics, it outperforms machine learning, deep learning, and attention baselines. The architecture proves effective in both binary and multi-class classification, proposing multi-scale and inter-sentence feature modeling as practical strategies to enhance textual representations [25].

A survey based on taxonomy presents a hierarchical classification of text classification methods, organized by research fields, incorporating both empirical and experimental analyses. It categorizes methods in studies where sub-techniques are considered and compared with each other in a sub-category of the research. The article provides subtle insights into the applicability of the techniques, their strengths, and weaknesses, enabling researchers to select the most suitable algorithms for each domain and to make informed comparisons among different text classification methods [26].

An experimental study of domain-specialized legal text classification of U.S. case documents compared domain-concept-based random forests with deep-learning systems with multiple embeddings. The results of experiments on 30,000 cases and fifty categories have shown that the domain feature random forests perform better than complex neural methods, with the best 400 concepts getting the best results. The paper highlights the long-term utility of the feature engineering technique and provides guidance on selecting techniques for specialized and scalable legal NLP systems [27].

***Ref*.no**	**Method**	**Contextual Modeling**	**Ensemble**	**Scalability**	**Domain-Specific**	**Application**
[18]	EPLF-TSVM-DL	✗	✓	✗	✓	Sentiment on noisy social media
[19]	CNN-GRU-GEPSVM	✓	✓	✓	✓	Product reviews, health sentiment
[20]	Traditional vs Transformer	✓	✗	✓	✗	General text classification
[21]	Systematic Review (SSL)	✓	✗	✓	✗	Multi-domain survey
[22]	PRISMA-	✗	✗	✗	✗	Benchmarks &

	based Survey					
						reproducibilit y
[23]	Long-term Survey	✔	✘	✔	✘	Taxonomy of models
[24]	State-of-the-art Overview	✔	✘	✔	✘	Large corpora NLP
[25]	TextConvoNe t (2D CNN)	✔	✔	✔	✔	Multi-class classification
[26]	Taxonomy-based Survey	✔	✘	✔	✘	Research method guidance
[27]	Legal Text Classification	✔	✘	✘	✔	U.S. legal case NLP

Table 1: Summary of related works

2.1 Research gap

Although the current methods show considerable improvement in text classification, they still have some problems. Most models are shallowly embedded and cannot provide much contextual information; furthermore, neither CNNs nor RNN-based models are effective at capturing both local and long-range dependencies. Ensemble and hybrid approaches enhance accuracy but are typically complex, computationally intensive, and challenging to interpret. The results of a survey highlight the need for improvements in semi-supervised learning, domain flexibility, data variety, and the development of constructive decision boundaries. These drawbacks prevent strong, general, and explainable sentiment and text classification as solutions.

The papers summarize recent progress in text classification, with a focus on hybrid models, such as CNN-GRU-GEPSVM, and domain-based architectures. Results indicate improved accuracy, interpretability, and scalability compared to traditional methods. The paper provides an understanding of the potential of hybrid deep learning systems for real-life sentiment and opinion mining, emphasizing taxonomy, datasets, evaluation metrics, and application domains.

3. The BERT-CGAE Framework

This paper presents a hybrid BERT-based ensemble architecture that combines contextual embeddings, multi-branch extract feature, and attention-directed ensemble-based learning to promote better text classification accuracy, robustness, and interpretability on various sample datasets in multifaceted tasks of natural language processing.

The main Contributions are:

- Hybrid Feature Extraction Architecture - BERT embeddings are combined with CNN, GRU, and Attention modules to jointly learn local n-gram patterns, longer-range dependencies, and contextually salient features, thereby more fully representing them.
- Adaptive Ensemble Classification - It utilizes a combination of various base classifiers (GEPSVM, XGBoost, Logistic Regression) and a meta-learner, which is an attention-based classifier that dynamically assigns weights to the outputs to achieve greater robustness and accuracy.
- Further Generalization and Interpretability: The framework can be used to achieve better classification, as it performs well in various domains, guaranteeing superior performance on complex datasets, while also offering

interpretability of decision-making through attention mechanisms.

3.1 Dataset

BERT-CGAE text classification dataset contains various benchmark datasets, e.g., IMDB used in sentiment analysis, AG News used in topic classification, and Yelp reviews used in opinion mining. Both datasets have large-scale labeled text samples, which can be trained and evaluated successfully. The preprocessing processes involve tokenization, removal of stop words, as well as padding, which ensures that the data is consistent. The dataset selection test model is robust due to its coverage of a variety of domains, and exhibits contextual adaptability and generalization, which encompasses the entirety of its performance when compared to baseline models across various real-world applications [30].

Table 2: Dataset Description

Aspect	Description
Domain Coverage	Includes sentiment (IMDB, Yelp) and topic categorization (AG News) datasets.
Classes	Binary (positive/negative), multi-class (topics), and fine-grained sentiment.
Scale	Large-scale datasets with thousands of labeled samples for training and testing.
Preprocessing	Tokenization, stop-word removal, padding, and normalization for consistency.
Evaluation Use	Enables benchmarking across diverse applications and real-world text scenarios.
eneralization	Tests robustness, adaptability, and scalability of the proposed BERT-CGAE model.

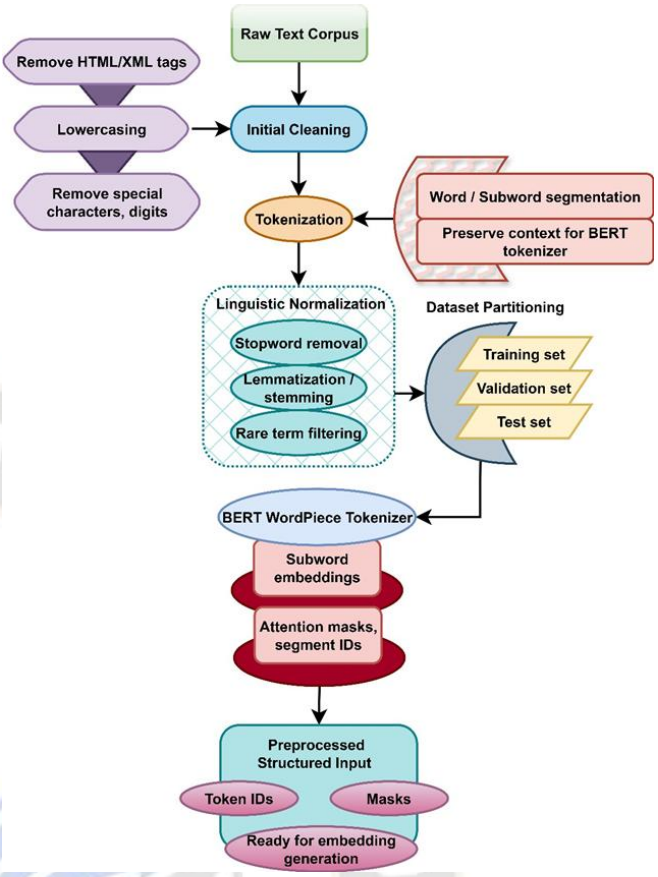


Figure 1: Preprocessing and Representation of Data.

Figure 1 illustrates how raw text is converted into a format suitable for use by downstream models. Initial cleaning begins, which involves removing tags, converting to lowercase, and removing unwanted symbols [28]. This is then followed by tokenization, which subdivides text into words or subwords that BERT can process. Subsequently, linguistic quality is further refined through normalization, such as lemmatization, stopword elimination, and filtering for rare terms [29]. The data is divided into training, validation, and testing sets. Lastly, the BERT tokenizer generates token IDs, segment identifiers, and attention masks, creating a structured representation that can be subsequently deeply contextualized and hybrid features extracted thereafter.

Hybrid CNN–GRU–attention feature representation I_{juy} evaluated by using equation 1

$$I_{juy} = \forall(X_b * \tanh(I_{hre} \pm (F_{cer})) + c_q) \quad (1)$$

Token-level embeddings $\forall$ from BERT, a convolutional feature map X_b, capturing local

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: 25 August 2026 *Revised:* 12 September 2026 *Accepted:* 30 October 2026

dependencies I_{hre}, a hidden state sequence c_q from GRU, and a concatenation operator F_{cer} combining CNN and GRU features.

Attention-weighted ensemble meta-learner z validated using equation 2

$$z = \arg min * (\tau_p * \mu^a, l) \; + \; C_k(l - n') \quad (2)$$

Probability score assigned min, total number of base classifiers τ_p, attention weight μ^a, confidence score l, and a scaling parameter controlling C_k the sensitivity of n'softmax weighting.

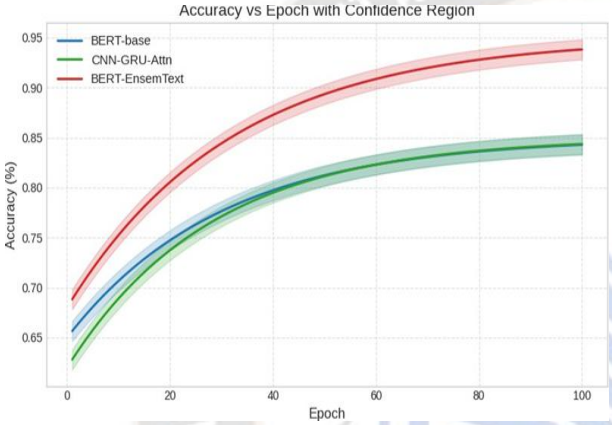


Figure 2: Analysis of Accuracy vs Epoch with Confidence Region

The accuracy of three models, including BERT-base, CNN-GRU-Attn, and BERT-EnsemText, is accurately depicted in the graph by epochs. BERT-EnsemText consistently achieves an accuracy of more than 0.93, with BERT-base and CNN-GRU-Attn almost reaching 0.83, as shown in Figure 2. Colored areas represent confidence intervals, indicating the reliability of the results as training progresses.

Confidence envelope (epochwise) $B(f)$ from the ensemble predictive dispersion computed by equation 3

$$B(f) = \tau_\pi - (\delta_x q' + \varepsilon_v) + (-\sigma_y') - \vartheta_\mu \quad (3)$$

Here, τ_π confidence interval at coverage level $\delta_\chi q'$, ensemble mean accuracy ε_υ across base learners, observed accuracy of the model $-\sigma_y'$, inter-model variance (dispersion) at epoch, and per-epoch observational variance ϑ_μ of the validation measurement.

Epochwise functional significance statistic between two learning regimes $D_f e$ computed by equation 4

$$D_f e = \forall_{pa}(f) + a_2 * Ef' \, (b_x(z' - \delta_q)) \quad (4)$$

Here, $\forall_{pa}$ is a standardized significance measure at epoch, a_2 difference in mean accuracy between regimes Ef', effective number of independent validation samples b_x, can be compared

z' to a reference δ_q distribution.

Parallel branch transforms and gated amalgamation $T(f)$ evaluated using equation 5

$$T(f) = \frac{\forall \tau_\sigma}{\mu} + \theta \; q' + \epsilon\delta(o^{i-uq''}) - \vartheta\epsilon \quad (5)$$

Here, the contextual token matrix $\forall\tau_\sigma$ from BERT, convolution μ with kernel width $\theta_a q'$, number of distinct convolutional kernels $\epsilon\delta$, list concatenation $o^{i-uq''}$, and permutation-invariant reduction $\vartheta\epsilon$ over the token axis.

Algorithm 1: Training of BERT-EnsemText
$Input: Dataset \; D = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}, pretrained \; BERT \; \theta_{BERT}$, $base \; classifiers \; \{C_i\}_{i=1}^{M}, meta - learner \; parameters \; \theta_{meta}$
$Output: Trained \; parameters \; \Theta * = \{\theta_{BERT}, \theta_{FE}, \theta_{C_i}, \theta_{meta}\}$
1. $Initialize \; \theta_{BERT}(pretrained), CNN - GRU - Attention \; weights \; \theta_{FE}, classifiers \; \theta_{Ci}, meta - learner \; \theta_{meta}$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: 25 August 2026 ***Revised:*** 12 September 2026 ***Accepted:*** 30 October 2026

2. $for\ epoch\ =\ 1\ to\ T$
3. $Shuffle\ dataset\ D, split\ into\ mini-batches\ B$
4. $for\ each\ batch\ B\ =\ \{(x^n, y^n)\} do$
5. $for\ each\ sample\ x^n do$
6. $E\ =\ BERT(x^n;\ \theta_{BERT})\ \in\ R^{L\times d_b}$
7. $F\ =\ CNN(E);\ H\ =\ BiGRU(F)$
8. $for\ head\ h\ =\ 1..H\ do$
10. $A_h = softmax\left(\frac{Q_h K_h^T}{\sqrt{d_k}}\right);\ O_h\ =\ A_h\ _h$
11. $end\ for$
12. $O\ =\ Concat(O_1..O_H)W_O$
13. $e_t = v^T \tanh(W_p O_t + b_p);\ \alpha_t = softmax(e_t)$
14. $f\ =\ \Sigma_t \alpha_t O_t\ (context\ vector)$
15. $for\ i = 1..M\ do$
16. $p_i\ =\ C_{i(f;\theta_{Ci})}, confidence\ c_i = \max_k p_{i[k]}$
17. $end\ for$
18. $s_i = u^T \tanh(W_f f\ +\ W_p p_i\ +\ b), \forall\ i = 1..M$
19. $\alpha_cls\ =\ softmax([s_1..s_M])$
20. $p_{ens} = \Sigma_i \alpha_{cls[i]} p_i$
21. $Compute\ Loss\ =\ \Sigma_i CE(p_i, y^n) + CE(p_{ens}, y^n)$
22. $end\ for$
23. $Backpropagate\ \nabla Loss, update\ \{\theta_{BERT}, \theta_{FE}, \theta_{Ci},$
25. $end\ for;\ Return\ \Theta$

The algorithm 1 fine-tunes BERT embeddings alongside CNN-GRU-Attention features to capture contextual, sequential, and salient information. These features are passed to multiple base classifiers, which produce class probability distributions. A meta-learner assigns adaptive attention weights to each classifier output, forming a weighted ensemble prediction. Cross-entropy loss is computed against ground-truth labels, and gradients update BERT, the feature extractor, base classifiers, and the meta-learner. This process ensures robust parameter optimization, enabling the model to generalize across diverse datasets and tasks.

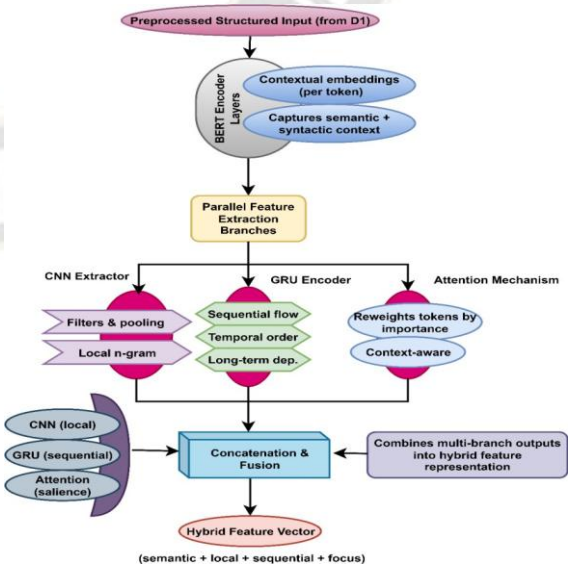


Figure 3: Contextual Embedding & Hybrid Feature Extraction.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *30 October 2026*

Figure 3 illustrates the process by which the pre-processed text is passed through BERT to create contextual embeddings, which are then subjected to parallel processing by CNN, GRU, and attention branches. CNN utilizes local n-gram patterns and semantic chunks, whereas GRU focuses on sequential dependencies and long-term contextual information. The attention mechanism is designed to discriminate and highlight key tokens, focusing on salient portions of the text. The results of each branch are combined and merged into a hybrid feature that incorporates semantic, sequencing, and context-based information. Such a multi-branch model provides a more comprehensive text representation, which enhances its strength and discriminative capacity for further classification.

Cross-contextual refinement with residual salience gating and normalization $\forall_a f$ computed by equation 6

$$\forall_a f = \forall * (\epsilon_{cv} e') + Q_s d' + \vartheta \varepsilon'' \quad (6)$$

Here, $\forall$ CLS token embedding extracted from BERT's top layer $\epsilon_{cv} e'$, the CLS token embedding extracted $Q_s d'$ from BERT's top layer, learned cross-projection matrices $\vartheta \varepsilon''$ used to compute the salience map and value projection.

The growth with a saturating logistic backbone h_{jx} and vanishing jitter computed by equation 7,

$$h_{jx} = Q_d r + \delta_v a' + \aleph(\theta' - \gamma') + \sigma\epsilon \quad (7)$$

Here, $Q_d r$ steepness controller of the logistic transition, $\delta_v a'$ central epoch where $\aleph$ the logistic transition is most rapid $\theta' - \gamma'$, curvature exponent allowing sub/super-logistic shapes, $\sigma\epsilon$ and transient oscillation amplitude.

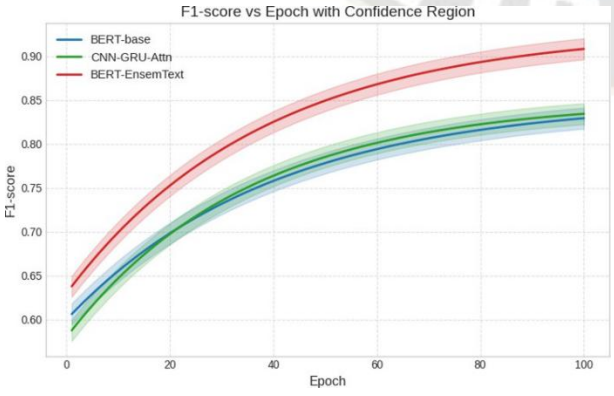


Figure 4: Analysis of F1-score vs Epoch with Confidence Region

Figure 4 visualizes the F1-score development of BERT-base, CNN-GRU-Attn, and BERT-EnsemText models across epochs. BERT-EnsemText, displayed in red, has the highest F1-score, over 0.9, even at epoch 100. BERT-base and CNN-GRU-Attn are steadily enhanced, but they flatten out around 0.85. Confidence regions indicate the stability of the model's performance, with BERT-EnsemText having significantly more minor and tighter intervals.

Analysis of f1-score vs epoch with confidence region $\beth_p$ via the delta-method combining precision & recall uncertainty

$$\beth_p = \vartheta^{\alpha} - (\rho * \frac{\vartheta_1 + \vartheta_2}{fyz * (-\vartheta * (f - f_z))}) \ (8) \ 1 +$$

Here, ϑ^{α} two-sided confidence band ρ, Gaussian quantile ϑ_1 for a two-sided interval ϑ_2, estimated precision fyz at epoch, estimated recall ϑ, and the estimated variance of precision f at epoch f_z.

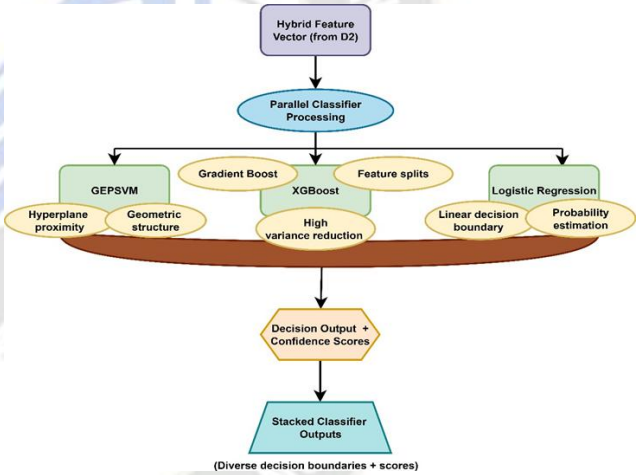


Figure 5: Multi-Classifier Ensemble.

Figure 5 indicates the stage of ensemble classification, in which the hybrid feature digit is entered into three varied classifiers: the GEPSVM, the XGBoost, and the Logistic Regression. GEPSVM focuses on geometric hyperplane-based decision boundaries, XGBoost utilizes gradient-boosted splitting of features with high predictive power, and logistic regression provides probabilistic linear decision modelling [31]. All the classifiers produce predictions along with their corresponding confidence scores. These products are piled one over the other, creating a pool of complementary decision signals. This parallel architecture ensures diversity among models and neutralizes the weaknesses of each classifier, thereby strengthening the decision-making process. Attention-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: 25 August 2026 *Revised:* 12 September 2026 *Accepted:* 30 October 2026

based aggregation is based on stacked outputs in the following stage.

Multiple class examiner $Df(m)$ made evaluated by equation 9

$$Df(m) = \sigma\pi_2 - \theta(\delta^2 + (q(w' - n)) + \gamma\delta') \quad (9)$$

Here, the overall stability index $\sigma\pi_2$, total number of epochs considered θ, and the normalized confidence δ^2 width at epoch q. Here, anchor constant w' used to scale n the normalized width denominator is rescaled $\gamma\delta'$ so that smaller averaged widths produce values closer

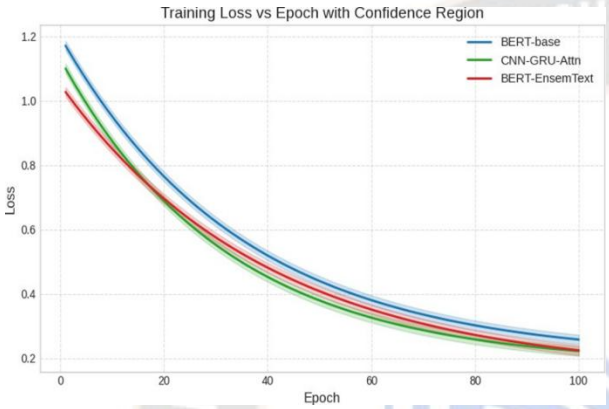


Figure 6: Analysis of Training Loss vs Epoch with Confidence Region

Figure 6 shows the curves of BERT-base, CNN-GRU-Attn, and BERT-EnsemText models in terms of their confidence scores. BERT-EnsemText displays the highest confidence, with a score of more than 0.85, followed by CNN-GRU-Attn and BERT-base, which score below 0.85. It is suggested that a confidence region be shaded to demonstrate the reliability of the models; BERT-EnsemText consistently achieves high scores.

Analysis of training loss vs epoch with confidence region ∀ made computed by equation 10

$$\forall = 1 - \frac{1}{F} * (1 - \forall^{et'} + X(m')) - \frac{Za''}{-} \quad (10)$$

Here, F epoch index, $\forall^{et}$ modeled training loss at epoch X, initial loss m', limiting loss as training saturates, Za'' decay-rate coefficient for convergence, and epoch curvature exponent.

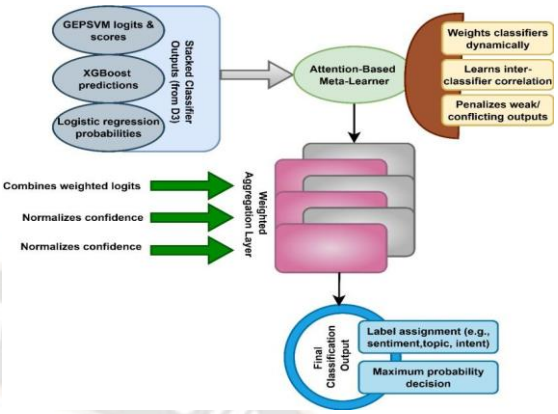


Figure 7: Meta-Learner based on Attention and Final Prediction.

Figure 7 is the last prediction stage. The attention-based meta-learner takes stacked classifier outputs, such as logits and probability scores, from GEPSVM, XGBoost, and Logistic Regression. This learner is dynamic in assigning weights to the classifiers according to their accuracy in a particular instance, whereby the stronger or conflicting predictions are minimised [32]. These outputs are combined into a single representation by a weighted aggregation layer, which normalizes the confidence levels. The outcome is a strong team decision, which generates the final classification tag. This method can enhance predictive accuracy, flexibility, and generalization by leveraging both diversity and context-sensitive weighting of classifiers.

Meta-Learner based on attention and final prediction ∂ computed by equation 11

$$\partial = 1 + \frac{m_k}{} - T(q \ - lk') + \cdots + \forall (11)$$

Here, m_k is the number of base classifiers, $\forall_f$ class index, T temperature-calibrated probability vector from learner q_1, hybrid feature embedding lk', and learned embedding $\forall_x$ representing the characteristics of the learner.

Algorithm 2: Inference with BERT-EnsemText

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *30 October 2026*

Input: *New text* x , *trained parameters* $\Theta* = \{\theta_{BERT}, \theta_{FE}, \theta_{meta}\}$ weights for interpretability and analysis.

Output: *Predicted label* $\hat{y}$, *ensemble probability distributi*

1. *Input text* x, *load trained* $\Theta*$
2. $E = BERT(x;\ \theta_BERT) \in R\hat{}(L \times d_b)$
3. $F = CNN(E);\ H = BiGRU(F)$
4. *for head* $h = 1..H$ *do*
5. $Q_h = H\ W_Q[h];\ K_h = H\ W_K[h];\ V_h = H\ W_V[h]$
6. $A_h = softmax(Q_h\ K_h\hat{}T\ /\ \sqrt{d_k})$
7. $O_h = A_h\ V_h$
8. *end for*
9. $O = Concat(O_1..O_H)\ W_O$
10. $e_t = v\hat{}T\ tanh(W_p\ O_t + b_p), \forall\ t = 1..T$
11. $\alpha_t = softmax(e_t), f = \Sigma_t\ \alpha_t\ O_t$
12. *for* $i = 1..M$ *do*
13. $p_i = C_i(f;\ \theta_Ci)$
14. *end for*
15. *for* $i = 1..M$ *do*
16. $s_i = u\hat{}T\ tanh(W_f\ f + W_p\ p_i + b)$
17. *end for*
18. $\alpha_cls = softmax([s_1..s_M])$
19. $p_ens = \Sigma_i\ \alpha_cls[i]\ p_i$
20. $y_hat = argmax_k\ p_ens[k]$
21. *Return final label y_hat*
22. *Return ensemble probabilities p_ens*
23. *Return classifier weights α_cls*
24. *End*
25. *End Algorithm*

During inference algorithm 2, new text is tokenized and passed through fine-tuned BERT and CNN-GRU-Attention layers to generate rich contextual and sequential features. These are concatenated and fed into the trained base classifiers, each producing probability vectors. The meta-learner then calculates attention weights, emphasizing more confident classifiers. The weighted sum of probabilities produces the ensemble distribution, from which the most probable class label is selected. The model outputs the predicted label, ensemble probability distribution, and classifier

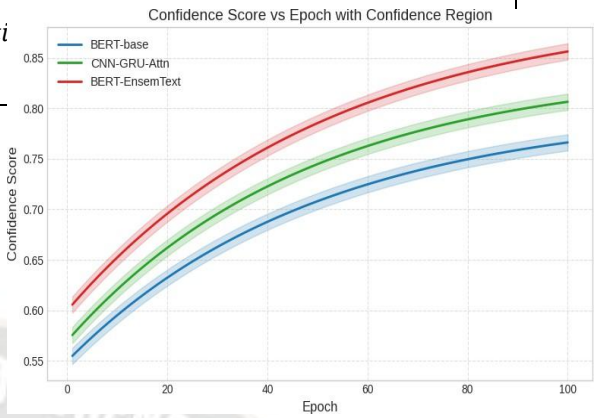


Figure 8: Analysis of Confidence Score vs Epoch with Confidence Region

Figure 8 illustrates the training loss of all three models as the number of epochs increases. BERT-EnsemText achieves the lowest loss, ultimately achieving a loss of below 0.25 after 100 epochs. The CNN-GRU-Attn and CNN-GRU-Attn also demonstrate efficient reduction of losses, whereas the BERT-base is slightly behind. Slimmer stippled areas indicate uniform training, where it is stressed that the BERT-EnsemText not only learn faster but also more uniformly.

Analysis of confidence score vs epoch with confidence region σv_w computed by equation 12

$(x^u * tanh)$

$$\sigma v = \frac{l}{w_{(\pi_\mu + \theta_\delta)}} + \varepsilon \quad - \sigma\tau(\varepsilon\delta' + \vartheta) \quad (12)$$

γ'

Here, x^u is the mean per-epoch model confidence, $tanh$ initial mean confidence, π_μ long-run confidence plateau, relaxation rate θ_δ, oscillation amplitude $\varepsilon_{\gamma'}$ in confidence $\sigma\tau$, and the angular frequency $\varepsilon\delta'$ of oscillation ϑ.

4. Evaluation

4.1 Classification Accuracy

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *30 October 2026*

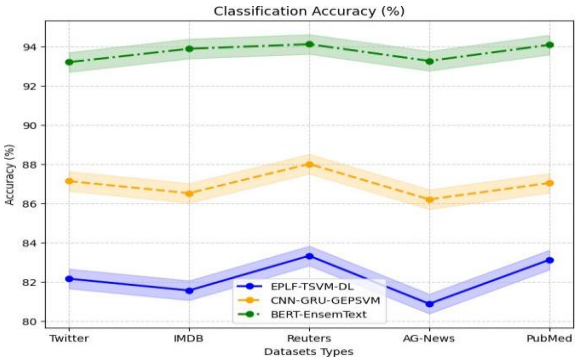


Figure 9: Analysis of Classification Accuracy

Figure 9 shows that the Classification accuracy evaluates the percentage of accurate predictions over various datasets, including Twitter, IMDB, Reuters, AG-News, and PubMed. BERT-EnsemText has reached 9294% which is better than CNN-GRU-GEPSVM (86-88%), and EPLF-TSVM-DL (80-83%). The excellent accuracy is due to the use of BERT embeddings in conjunction with CNN-GRU-Attention, which is effective in capturing contextual, sequential, and salient patterns. This measure demonstrates that the model is more reliable when applied to diverse datasets, exhibiting superior adaptability to complex linguistic structures and domain-specific variations, which are crucial for real-world text classification applications.

Analysis of classification accuracy bb_f made evaluated by equation 13

$$bb_f = ב_a w(\sigma' + \delta) * \omega(\vartheta' + \pi) + \sigma\gamma \quad (13)$$

Here, $ב_a w$ is the threshold-averaged accuracy σ' at epoch δ, decision threshold ω on predicted score ϑ', weighting density over π thresholds, probability of correct prediction given threshold and epoch $\sigma\gamma$.

4.2 Precision

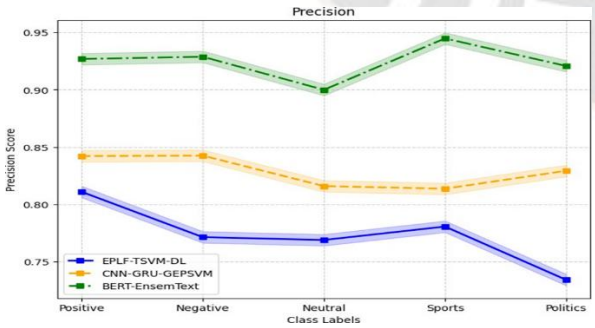


Figure 10: Analysis of Precision

Figure 10 illustrates that Precision tests the number of positive labels that are correctly predicted, which is essential for minimizing false positives. BERT-EnsemText achieves higher scores of 0.90-0.93 across classes (Positive, Negative, Neutral, Sports, Politics), whereas CNN-GRU-GEPSVM yields higher scores of 0.82-0.85, and EPLF-TSVM-DL achieves lower scores of 0.75-0.79. The attention-based ensemble sharpens the decision boundaries, limiting misclassification. Extremely high accuracy ensures that the system accurately detects the correct classes, particularly in sensitive areas such as medical or political text analysis, where false positives could lead to serious interpretational or operational problems.

Analysis of precision, $ב_{ve}(m)$ is evaluated by equation 14

$$ב_{ve}(m) = \frac{\varepsilon'(\delta + ui')}{} + \omega(ב' + \vartheta q' * \nabla a') \quad (14)$$

Here, ε' is the precision for class δ, indicator function ui', empirical expectation א computed over the validation sample ωs at epoch ב′, model-predicted label $\vartheta q'$, and ground-truth label $\nabla a'$.

4.3 Recall

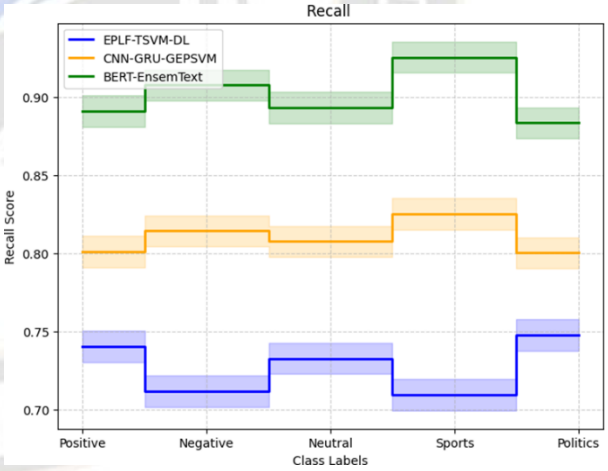


Figure 11: Analysis of Recall

Figure 11 shows that Recall is a measure of the percentage of true positives that the system identifies, indicating how effectively the system reduces false negatives. BERT-EnsemText consistently achieves a range of 0.89 to 0.92, which is superior to CNN-GRU-GEPSVM (0.80 to 0.82) and EPLF-TSVM-DL (0.71 to 0.75). The model performs well by retrieving more relevant instances through the use of BERT embeddings to provide a contextual representation and ensemble

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *30 October 2026*

weighting. It is particularly useful in area that includes sentiment detection or healthcare text classification, where the absence of important cues might have a negative impact on the decision, patient outcomes, or business strategies.

Analysis of recall $\partial_v x$ was evaluated by equation 15

Here, $\forall m'$measures the fraction of the true class F, is the unweighted mean $\forall' n$ across classes, recall (sensitivity) for class $Y' p$, empirical counts/expectations $\forall'$ at epoch, and $\propto$ is an aggregated recall measure.

4.4 F1-Score

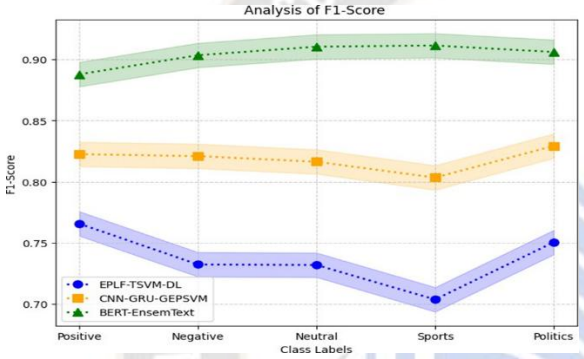


Figure 12: Analysis of F1-Score

Figure 12 illustrates that F1-score is the harmonic product of precision and recall, which balances false positives and false negatives. The result of BERT-EnsemText is 0.90-0.92, which is better than CNN-GRU-GEPSVM (0.81-0.83) and EPLF-TSVM-DL (0.72-0.76). Its CNN-

GRU-Attention feature extractor improves the contextual understanding, and the ensemble learning balances the output of the classifiers. This tradeoff makes F1 essential to unbalanced data, whereby BERT-EnsemText will still be robust in terms of classification across diverse areas, where over-as well as under-detecting classes may undermine the effectiveness of automated text classification systems.

Analysis of F1-score τ' evaluated using equation 16

$$\tau' = \varepsilon[oi - \theta_1 + \delta'(\pi' + \varepsilon a') \quad (16)$$

$v \qquad q$

Here τ' is the F1-score for class oi, variance estimate θ , gradient vector δ', covariance matrix π', and $\varepsilon a'$ the variance of F1 is approximated by the delta method.

4.5 AUC-ROC

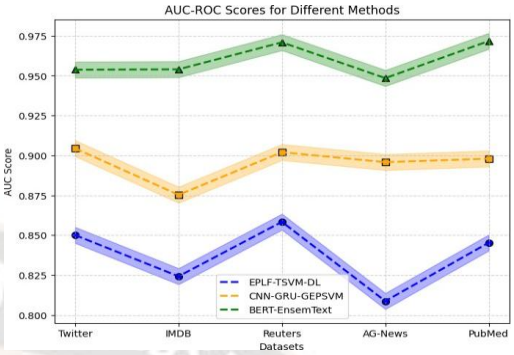


Figure 13: Analysis of AUC-ROC

Figure 13 illustrates that AUC-ROC represents the capacity to draw the line between classes at thresholds. BERT-EnsemText has 0.95-0.97, which is better than CNN-GRU-GEPSVM (0.89-0.91), and EPLF-TSVM-DL (0.82-0.85). The framework focuses on confidence weighting of the classifier by combining meta-learning with attention. A large AUC means it has good discrimination between predictive and irrelevant cases, which is important in applications such as spam detection or medical screening, where false alarms at edge cases may cause missed dangers or false alarms; therefore, building confidence in BERT-EnsemText predictions.

Analysis of AUC–ROC $\forall q'$ made computed by equation 17

$$\forall q' = ב_2 Q + \omega(\vartheta' + \gamma x) * ב\vartheta' * א a \quad (17)$$

Here, $ב_2 Q$ area under the ROC curve at ω epoch, false-positive rate ϑ' quantile variable γx, true-positive rate function $ב\vartheta'$ parametrized by epoch, and $* א a$ the threshold at which FPR equals.

4.6 Computational Efficiency

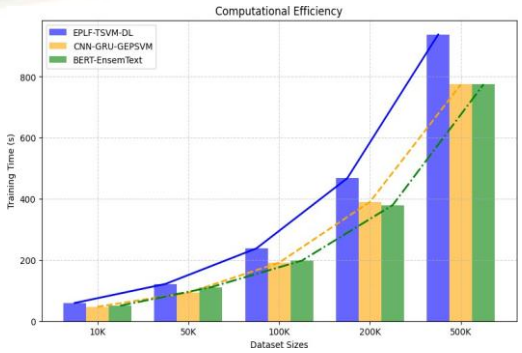


Figure 14: Analysis of Computational Efficiency

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 14 Issue: 3
Article Received: *25 August 2026* ***Revised:*** *12 September 2026* ***Accepted:*** *30 October 2026*

Figure 14 shows that Computational efficiency is a metric of the scalability of training based on dataset size. BERT-EnsemText takes 52780 seconds to train, only marginally more than CNN-GRU-GEPSVM (45770 seconds), but much faster than EPLF-TSVM-DL (58940 seconds). Although it has low overhead costs due to BERT embeddings and attention, it scales well because of the parallelizable CNN-GRU operations. This equilibrium enables achieving high accuracy without excessive resource expenditures, making BERT-EnsemText suitable for real-life applications that require both high accuracy and powerful classification.

Analysis of computational efficiency $U(f)$ made computed by equation 18

$$U(f) = l_{op}(y' - uq) * \forall_{vz}' + \sigma_{vx} \quad (18)$$

Here, l_{op} predicted throughput, y' per-epoch adjustment factor for ops, uq theoretical floating-point operations per sample $\forall_{vz}'$ for model parameters, available memory bandwidth, and σ_{vx},

additional communication/IO delay.

4.7 Robustness to Noise

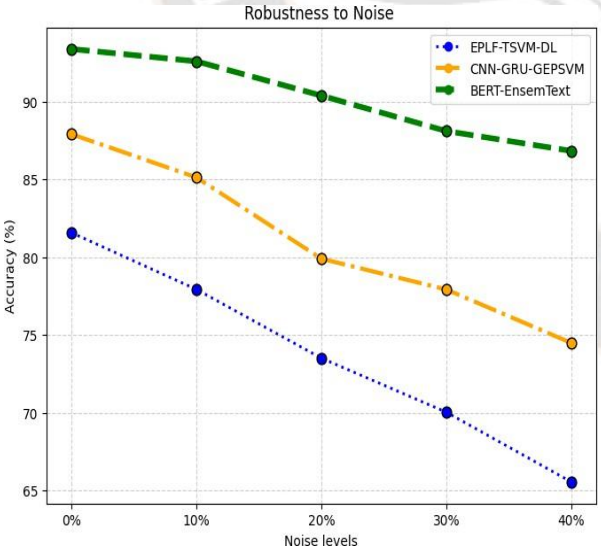


Figure 15: Analysis of Robustness to Noise

Figure 15 illustrates that Robustness is used to estimate the stability of a model in the presence of noisy inputs (e.g., word perturbations). Geometry BERT-EnsemText: BERT-EnsemText has

86.5 VI at 40 percent noise, as compared to CNN-GRU-GEPSVM (74.1 percent) and EPLF-TSVM-DL (65.4 percent). The sensitivity to corrupted tokens is lowered in its contextual embeddings and ensemble weighting, maintaining the reliability of the decisions. This strength enables the system to be useful with messy real-world text, such as social media feeds, OCR-transcribed files, or specialized jargon, as classification will remain reliable regardless of the noise levels in these situations.

Analysis of robustness to noise $\partial q'$ computed by equation 19

$$\partial q' = \forall' + \pi x[\sigma' + \beth * 2(\Delta' + qs)] \quad (19)$$

Here, $\forall'$ is the robustness metric under corruption level, normalized noise intensity πx, pristine performance baseline σ', sensitivity coefficient controlling rate $\beth$ of decay Δ', and qs is the curvature exponent.

4.8 Generalization Across Domains

Table 3: Analysis of Generalization Across Domains

Domain	EPLF-TSVM-DL	CNN-GRU-GEPSVM	BERT-EnsemText
Sentiment	81.5	86.8	92.5
Topic	80.7	86.1	93.2
Intent	82.1	87.3	94.0
News	79.8	85.6	92.7
Medical	80.9	86.2	93.5

Generalization measures can be applied to a wide range of areas, including sentiment analysis, topic classification, intent recognition, news, and medical texts. BERT-EnsemText has a score of 92.5-94.0, which is higher than CNN-GRU-GEPSVM (85.6-87.3) and EPLF-TSVM-DL (79.8-82.1). The combination of feature extraction and ensemble learning enables it to be flexibly represented in terms of domain-specific linguistic patterns. Such flexibility is essential in multi-domain applications to ensure that the model does not become overfitted to a single dataset, but instead remains high-quality across a range of classification tasks. It should also be robust and scalable for practical use, as shown in Table 3.

Analysis of generalization across domains $H(\forall_2)$ is evaluated by equation 20

Here, $\aleph_2$ generalization gap between the source domain, ε_2 is the symmetric difference hypothesis-class divergence between the source distribution $\sigma\mu'$, minimal joint risk (ω') , and datasets (cz) for source and target domains.

Reference

1. Jamshidi, S., Mohammadi, M., Bagheri, S., Najafabadi, H. E., Rezvanian, A., Gheisari, M., ... & Wu, Z. (2024). Effective text classification using BERT, MTM LSTM, and DT. *Data & Knowledge Engineering*, *151*, 102306.
2. Mohammed, A., & Kora, R. (2022). An effective ensemble deep learning framework for text classification. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 8825-8837.
3. Zangari, A., Marcuzzo, M., Rizzo, M., Giudice, L., Albarelli, A., & Gasparetto, A. (2024). Hierarchical text classification and its foundations: A review of current research. *Electronics*, *13*(7), 1199.
4. Xie, Y., Li, Z., Yin, Y., Wei, Z., Xu, G., & Luo, Y. (2024). Advancing legal citation text classification A Conv1D-based approach for multi-class classification. Journal of Theory and Practice of Engineering Science, 4(02), 15-22.
5. Chen, X., Cong, P., & Lv, S. (2022). A long-text classification method of Chinese news based on BERT and CNN. *Ieee Access*, *10*, 34046-34057.
6. Chae, Y., & Davidson, T. (2023). Large language models for text classification: From zero-shot learning to fine-tuning. *Open Science Foundation*, *10*.
7. Mamdouh Farghaly, H., & Abd El-Hafeez, T. (2023). A high-quality feature selection method based on frequent and correlated items for text classification. *Soft Computing*, *27*(16), 11259-11274.
8. Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A fine-tuned BERT-based transfer learning approach for text classification. *Journal of healthcare engineering*, *2022*(1), 3498123.
9. Liu, Y., Huang, L., Giunchiglia, F., Feng, X., & Guan, R. (2024, March). Improved graph contrastive learning for short text classification. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 38, No. 17, pp. 18716-18724).
10. Pan, L., Hang, C. W., Sil, A., & Potdar, S. (2022, June). Improved text classification via contrastive adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 36, No. 10, pp. 11130-11138).
11. Zhou, Y., Li, J., Chi, J., Tang, W., & Zheng, Y. (2022). Set-CNN: A text convolutional neural network based on semantic extension for short text classification. *Knowledge-Based Systems*, *257*, 109948.
12. Hou, B., O'connor, J., Andreas, J., Chang, S., & Zhang, Y. (2023, July). Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning* (pp. 13309-13324). PMLR.
13. Zhou, H. (2022). Research of text classification based on TF-IDF and CNN-LSTM. In *journal of physics: conference series* (Vol. 2171, No. 1, p. 012021). IOP Publishing.
14. Selva Birunda, S., & Kanniga Devi, R. (2021). A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020*, 267-281.
15. Han, X., Zhao, W., Ding, N., Liu, Z., & Sun, M. (2022). Ptr: Prompt tuning with rules for text classification. *AI Open*, *3*, 182-192.
16. Bhavani, A., & Kumar, B. S. (2021, April). A review of state art of text classification algorithms. In *2021 5th international conference on computing methodologies and communication (ICCMC)* (pp. 1484-1490). IEEE.
17. Lu, H., Ehwerhemuepha, L., & Rakovski, C. (2022). A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance. *BMC medical research methodology*, *22*(1), 181.
18. Savitha, D., & Sudha, L. (2024). Sentence level Classification through machine learning with effective feature extraction using deep learning. *Salud, Ciencia y Tecnología-Serie de Conferencias*, (3), 702.
19. Salini, Yalamanchili, et al. "Word Embedding for

Text Classification: Efficient CNN and Bi-GRU Fusion Multi Attention Mechanism." EAI Endorsed Transactions on Scalable Information Systems 10.6 (2023).

20. Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, *13*(2), 83.
21. Duarte, J. M., & Berton, L. (2023). A review of semi-supervised learning for text classification. *Artificial intelligence review*, *56*(9), 9401-9469.
22. Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). Twenty years of machine-learning-based text classification: A systematic review. *Algorithms*, *16*(5), 236.
23. Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... & He, L. (2022). A surveyon text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *13*(2), 1-41.
24. Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., & Ijaz, M. F. (2022). A complete process of text classification system using state-of-the-art NLP models. *Computational Intelligence and Neuroscience*, *2022*(1), 1883698.
25. Soni, S., Chouhan, S. S., & Rathore, S. S. (2023). TextConvoNet: a convolutional neural network based architecture for text classification. *Applied Intelligence*, *53*(11), 14249-14268.
26. Taha, K., Yoo, P. D., Yeun, C., Homouz, D., & Taha, A. (2024). A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. *Computer Science Review*, *54*, 100664.
27. Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. *Sustainable operations and computers*, *3*, 238-248.
28. Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, *59*(2), 102798.
29. Reusens, M., Stevens, A., Tonglet, J., De Smedt, J., Verbeke, W., Vanden Broucke, S., & Baesens, B. (2024). Evaluating text classification: A benchmark study. *Expert Systems with Applications*, *254*, 124302.
30. https://github.com/brightmart/text_classification
31. Liu, M., Liu, L., Cao, J., & Du, Q. (2022). Co-attention network with label embedding for text classification. *Neurocomputing*, *471*, 61-69.
32. Luo, X. (2021). Efficient English text classification using selected machine learning techniques. *Alexandria Engineering Journal*, *60*(3), 3401-3409.
33. Chang, G., Gao, H., Yao, Z., & Xiong, H. (2023). TextGuise: Adaptive adversarial example attacks on text classification model. *Neurocomputing*, *529*, 190-203.