SvedbergOpen
DISSEMINATIONOFKNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Adaptive Dual-Script Fusion for Low-Resource Text Classification with Selective State-Space Modelling

Masooda Masarat[1]*, Prof. Rumaan Bashir[2]

[1]Research Scholar Department of Computer Science Islamic University of Science & Technology, Kashmir.
E-mail: masooda.masarat@iust.ac.in
Department of computer Science Islamic University of Science and Technology, Kashmir.
[2]E-mail: rumaan.bashir@islamicuniversity.edu.in
*Corresponding Author

Abstract
Kashmiri is a low-resource language where the amount of labelled data is lower compared to other natural language processing tasks and the language is mainly written in Perso-Arabic (Nastalik) script. In this paper, a proposed model with two classifiers using pooled dual-script transformer embeddings with two state-of-the-art models: LightGBM and CatBoost, as well as a custom sequence model called DSF-Mamba, that combines the Perso-Arabic and Devanagari-transliterated representations using a gated fusion layer followed by a selective state-space encoder is evaluated for multiclass Kashmiri text classification. The three models use pre-trained frozen non-fine-tuned embeddings of XLM-RoBERTa (Perso-Arabic branch) and MuRIL (Devanagari-transliterated branch) to save computation for fine-tuning the end-to-end transformer, while still taking advantage of the pre-trained multi-lingual representations. The self-created dataset consists of 51,000 sentences, which upon cleaning process yielded 50,966 sentences. The proposed DSF-Mamba classifier achieves the highest accuracy (89.25%) on a held-out test set with 7,645 sentences, which comes from an 85/15 stratified split in comparison with LightGBM (87.30%) and CatBoost (83.96%). A theoretical analysis of the dual-script fusion mechanism is also provided, as part of an expanded version of a class-level error analysis, along with discussion of the model architectural and training differences of the three models. The results indicate that simple, frozen embedding sequence models can provide a desirable accuracy compute ratio for low-resource language classification tasks, while still allowing for hybrid and ensemble methods.

Keywords: Multiclass text classification, Kashmiri language, selective state-space models, mamba, dual-script fusion, gradient boosting.

1. Introduction

Advances in Natural Language Processing has become one of the most impactful sub-fields of artificial intelligence, revolutionizing the way computational systems handle human language, such as how they interpret, process, and produce it. In the last decade, remarkable progress has been made in deep representation learning and large-scale pretrained language models, which have shown great promise across a broad range of understanding tasks of a language like machine translation, question answering, text classification information retrieval, sentiment analysis, summarisation and dialogue generation [1]. Transformer models have ushered in a paradigm shift in contextual representation learning for the first time, long-range semantic dependencies can be captured by the self-attention mechanisms used in these architectures [2]. The models have consistently set new performance records over a variety of multilingual tasks, achieving outstanding generalisation ability and powerful semantic representation learning. Pretrained multilingual transformer models are thus the basis of modern NLP systems, bypassing the use of conventional feature engineering techniques to data driven contexts in learning paradigms [3]. Although these astonishing advances have been made, language technologies based on transformers haven't reached every language. There has been a considerable progress in high resource languages like English, Chinese, French and German, as a result of the availability of huge annotated and massive computational resources, many languages of the world remain under-resourced, linguistically under-annotated and under-supported by computational resources [4]. Low-resource languages, as they are typically known, are languages without abundant resources of labelled data, which is the key that enables supervised learning algorithms to reliably generalise [5]. The lack of annotated corpora is often the reason why more complex deep-learning methods can only be applied to specific tasks, and motivates the creation of methods that are theoretically efficient, but require less expensive task-specific optimization, and then are still able to transfer their multilingual knowledge gained from large-scale training [6]. The classification of texts is one of the very few well-formulated problems in NLP and is a critical and fundamental problem in intelligent information processing systems [7]. High volumes of texts are easier to organise automatically and

automatic categorisation of text documents enable management of digital libraries, news recommendation, intelligent search engines, information retrieval systems, content moderation systems, e-governance systems, social media monitoring systems, healthcare informatics systems, and decision-support systems [8]. The performance of these applications critically depends on the capacity of the computational models to correctly establish the semantic relationships between the elements of the textual sequences while also being low-computational to operate in real-world applications [9]. Thus, the research of strong text classification techniques remains a rising and impacting field under the fields of machine learning and artificial intelligence [10]. Results from the traditional machine learning methods for text classification typically depend on the use of manually designed linguistic features extracted from bag-of-words representations, term frequency-inverse document frequency (TF–IDF), n-gram statistics, and syntactic features [11]. While these techniques were competitive for a number of benchmark datasets, they did not have a great ability to capture more contextual semantics, as each word was essentially modelled as an independent feature [12]. These limitations were partially overcome by the emergence of distributed word representations, such as Word2Vec, GloVe and FastText, that map a semantically related words to vectors. However, these static embeddings were still unable to distinguish between the different contexts in which these words were used, treating a word with multiple meanings as the same representation, regardless of the context [13]. The problem was solved by contextual transformer-based language models that generate context-dependent embeddings, i.e., the representation of each token changes dynamically in response to the neighbouring words and the semantic structure of the sentence [14]. However, recently many language model transformers, namely as XLM-RoBERTa, mBERT and MuRIL have pushed contextual representation learning to multiple languages by jointly training from multilingual text corpora that spans hundreds of languages. Such models have shown very good transfer learning potential, with downstream models making use of linguistic knowledge gained from multilingual, large-scale pretraining [15]. They have been found to be useful in low resource language processing because they are able to explicitly represent semantically related concepts in different languages, a property for which task-specific annotated corpora are not available [16]. The ability of multilingual transformers to represent is truly remarkable, but the capacity comes with substantial complexity. These end-to-end fine-tuning strategies usually involve adjusting hundreds of millions of parameters in the model, which can use up a lot of GPU-memory, take a long time to optimise and consume a lot of energy [17]. Computing requirements are often too high to be met by the hardware capacity of most academic institutions and research labs interested in low-resource languages, making it difficult to apply cutting-edge transformer-based approaches. Recognising this limitation, frozen multilingual representation learning has gained traction in recent years, where one can use the pretrained transformer encoders as fixed feature extractors, with the downstream task-specific model trained based on the contextual embedding [18]. This paradigm significantly lowers the computational expenses, as optimization of the transformer backbone is only done once and reused efficiently in various downstream tasks, using pretrained linguistic knowledge. Moreover, Frozen representation learning not only saves memory, reduces the training time, but also helps to reduce the danger of overfitting if the labelled data set is not large [19]. It is therefore a new direction to explore for designing efficient multilingual NLP systems that use high-level representation while maintaining good computation efficiency especially in the research of text classification [20]. The paper addresses Kashmiri multiclass text classification using a curated, sentence-level corpus spanning nine domains: Science & Technology, Sports, Tourism, Politics, Education, Health, Finance, Weather and Other. In contract to transformer-based architecture that require full fine-tuning, the present study instead focuses on evaluating three additional, comparatively lightweight approaches that avoid full transformer fine-tuning: two gradient-boosted decision tree ensembles (LightGBM, CatBoost) operating on pooled embeddings, and a novel Gated Dual-Script Fusion Mamba (DSF-Mamba) architecture that operates on full-sequence embeddings via a selective state-space layer [21]. All three approaches share a common feature extraction backbone that is frozen, pretrained multilingual transformer encoders, decoupling representation learning from classifier training and substantially reducing the compute budget required for experimentation on the full corpus [22].

2. Literature Review

2.1 Recent Advances in Kashmiri NLP

Kashmiri is one of the officially recognized languages of India but from the view of Natural Language Processing (NLP), it is a low resource language [23]. There are a number of computational issues in the language, such as complex morphology, free word order, dialectal diversity, and the lack of annotated corpora, as well as the use of both Perso-Arabic and Devanagari writing systems. They have significant implications for lexical normalization, tokenization, contextual representation learning and language understanding applications [24]. Hence, the progress of developing good NLP based system for Kashmiri is relatively less than that of other major Indian languages. Early research efforts in this domain has centred on the creation of basic linguistic resources, such as linguistic corpora, lexical resources, part-of-speech

tagging resources, language specific pre-processing tools etc., that serve as the prerequisites for other linguistic work [25]. These resources laid the groundwork that was needed for further supervised learning approaches, but they were still not adequate for supporting large-scale neural language modelling [26]. More recently, the research focus has turned to the development of benchmark datasets, multilingual representation learning, and transformer-based architectures, as part of the general shift of research in NLP from handcrafted linguistic features to contextual representation learning [27]. One of the important steps was the creation of a manually annotated news snippet classification corpus with over fifteen thousand labelled instances, spread over various domain categories. In the study, the authors have evaluated the fine-tuned transformer models systematically on Kashmiri news classification: conventional algorithms such as machine learning, transformer architectures, deep neural networks, and large language models, with fine-tuned ParsBERT utilizing the best performance [28]. Importantly, it has shown that multilingual contextual representations for Kashmiri prove to be an effective way to realize practicality, and underscored the ongoing dearth of task specific datasets. Kashmiri NLP research has since extended well beyond text classification with significant efforts added to its scope in recent studies. A complete translation system of neural machine translation was developed, using approximately 270,000-word pairs and testing various translation architectures such as an encoder-decoder, attention and Transformer for Kashmiri-English translation for the first time [29]. The Transformer model proved to be more successful in the context of long-distance contextual dependence and morphologically complex sentence structure, further highlighting the need for context modelling with sequence for low-resource languages. Likewise, recent research has brought in some special linguistic resources such as the first standard Kashmiri word-sense disambiguation corpus with almost 20,000 annotated sentences, which is based upon Kashmiri WordNet [30]. Experiments using machine learning on this corpus verified that using contextual word embeddings is a significant improvement over using bag of words features for semantic disambiguation. Together these advances indicate that Kashmiri NLP is moving towards a focus on advanced representation learning and intelligent language understanding.

2.2 Multilingual Transformer-Based Text Classification:

Transformer-based language models have revolutionized the field of multilingual text classification by paradigm shift in contextual representation learning [31]. Models like BERT, RoBERTa, XLM-RoBERTa, mBEnRT, IndicBERT, MuRIL and various language-specific transformer variants have shown to be consistently the most powerful models for document classification, sentiment analysis, topic categorization, and question answering under the multilingual setup [32]. Their success is due to the self-attention mechanism that is capable of representing long range contextual dependencies alongside learning language-invariant semantic representations by training them on a huge multilingual collection [33]. Multilingual transformers offer a very efficient transfer-learning approach in morphologically rich and low resource languages as they gain knowledge from linguistically similar languages [34]. This Kashmiri text classification study proved that transformer models trained on Persian, Arabic, Urdu and multilingual corpora significantly outperform traditional neural models due to the presence of common lexical and morphological features among the languages. In the context of multilingual and code-mixed text classification tasks involving Indic languages such as Hindi, Kannada, etc., similar results have been reported with the use of multilingual contextual representations to enhance the robustness of the classification task [35]. Although multilingual transformers are successful, they are still expensive due to the large number of parameters that need to be optimized downstream, typically on the order of hundreds of millions. Consequently, there has been a growing interest in finding alternatives that can retain the pretrained linguistic knowledge and lower optimization complexity, which are computationally efficient [36]. Therefore, a considerable amount of research has developed in the direction of finding an optimized alternative that can preserve the pretrained linguistic knowledge while minimizing optimization complexity.

2.3 Parameter-Efficient and Frozen Representation Learning:

Parameter-efficient learning is a recent research direction for LLMs. Frozen representation learning can be performed by only fine-tuning the limited downstream component layers, and only using the pretrained transformer layers as contextual feature extractors for the downstream application [37]. This approach can achieve considerable reductions in GPU memory usage, training time, and optimisation expenses, while maintaining the semantic knowledge gained in the multilingual pre-training stage [38]. Recent multi-lingual classification studies have shown that the contextual embeddings extracted from frozen Transformer encoders are very discriminative if used along with efficient classifiers [39]. These have been especially appealing in low resource domain, where annotated data sets are relatively small and computing resources available are limited. However, most of the existing studies use a single embedding representation or feature concatenation for static integration of complementary multilingual representations, and less or no work has been done on adapting the representations [40].

2.4 Selective State-Space Models for NLP

Recently, an alternative approach to transformer-based sequence modelling, called Selective State-Space Models (SSMs) and the Mamba architecture has been gaining attraction [41]. Unlike self-attention methods, which are quadratically complex with sequence length, Mamba uses input-dependent state transitions that are linear with sequence length and maintains long-range dependencies [42]. The architectural design is good at processing long text sequence with maximum capability to represent. Mamba-based models have been used for long-document classification, legal document understanding, and sequence modelling, achieving performance competitive with transformer baselines and a significant reduction in computation costs [43]. Most of this research is conducted on high-resource languages and uses only single-script textual representations, whereas the present study targets sequence classification for a low-resource language across two distinct scripts Perso-Arabic and Devanagari. Based on the literature review conducted, there is no previous research that has explored a selective state-space architecture on adaptive dual-script multi-lingual representations for the Kashmiri language or similar Perso-Arabic/Devanagari scripts.

2.5 Gradient Boosting with Contextual Embeddings

In terms of structured feature learning, gradient boosting models especially LightGBM and CatBoost are still considered as the most powerful classical machine learning techniques [44]. When they act as features to leverage contextual embeddings, their power to learn highly nonlinear decision boundaries with efficient optimisation and excellent generalisation capacity has made them popular in many NLP applications [45]. These strategies can be divided into two types: they can be applied directly on the source text, or they can use a dense semantic representation learned from a pretrained language model, enabling the tree-based learners to benefit from high-quality contextual information without suffering from the high computational complexity[46]. In the recent years, several multilingual text classification research works have benchmarked such an integration of transformer embedding with gradient boosting approaches and shown that contextual word embedding with efficient ensemble learning delivers competitive results with limited computation resources. However, tree-based classifiers cannot include discrete representation of token-level interactions across the sequence as they are required to operate on pooled sentence representations only [47]. This limitation motivates the exploration of lightweight sequence-learning architectures that can capture other relationships across the entire context beyond the global embedding aggregation [48].

2.6 Dual-Script and Multilingual Representation Learning:

Multiple forms of orthographic representations are difficult for languages with more than one system where semantically similar words are written in different systems. The learning of complementary semantic representations is therefore required in addition to the maintenance of script specific linguistic features, if dual-script processing is to be involved [49]. This limitation is partially addressed by existing multilingual transformer architectures, which project language specific inputs into a shared cross-lingual embedding space through joint pretraining objective. However, the majority of prior approaches collapse multi-script representations into a single, script-agnostic embedding through naïve aggregation operations typically concatenation or mean-pooling thereby conflating orthographically distinct token distributions and failing to preserve script-level inductive biases [50]. The development of adaptive fusion mechanisms that dynamically adjust weights for the complementary representation of different scripts continue to be relatively under explored, particularly in low-resource multilingual domains [51]. In addition, systematic study with the combination of adaptive dual-script representation learning and selective state-space sequence modelling for multilingual text classification hasn't been systematically explored, where the presence of two scripts in the language allows the model to combine semantically similar content expressed through orthographically distinct and in places phonetically complementary writing systems.

3. Research Gap and Novelty:

Even with the huge strides made with multilingual transformer-based models, various methodological challenges are still unaddressed for effective text classification in low-resource multilingual domains [52]. One of the biggest problems with existing approaches is the high computational cost, large memory usage and small scale for deployments with limited resources, as most methods require end-to-end fine-tuning of large pretrained language models [53]. Recently, frozen multilingual representations have been a promising option in terms of computation but they have not been extensively explored in the use of downstream classification architecture.

Second, most current multilingual text classification systems usually use only one of the different contextual representations or concatenate features, if any, statically [54]. Such methods assume an identical level of importance for all embedding spaces, and cannot benefit from complementary semantic information that comes from different scripts or multilingual encoders in an adaptive way. Therefore, the discriminative capability of "complementary contextual representations" is underutilised [55].

Third, there has been recent progress in linear sequence modelling with competitive performance using Selective State-Space Models (SSMs), such as Mamba [56]. Current applications are mostly geared towards

high-resource language and textual stream applications. The coupling of the adaptive dual-representation learning for multilingual text classification has been largely overlooked, and there is yet no systematic study on the interplay of the frozen multilingual embeddings, dynamic feature fusion, and selective state-space sequence modelling in a common experimental setup [57].

Lastly, the majority of existing works comparing the two approaches mainly focus on classification accuracy and offer little understanding of the architectural compromises that must be made with a tree-based ensemble learning and lightweight sequence modelling, when operating on the same multilingual context representation [58]. So far, the relative contribution of the downstream architecture of classifiers is not well understood.

To overcome these drawbacks, a novel sequence-learning framework named DSF-Mamba is proposed, which is lightweight and combines frozen multilingual contextual representations, adaptive gated dual-script feature fusion and selective state-space modelling within a single framework. The proposed framework ensures a consistent multilingual representation backbone for all competing models, thus separating the effect of the classifier architecture from the various aspects of evaluation: computational efficiency, context modelling and multiclass classification performance in a low-resource multilingual setting.

3.1 Novelty Comparison

Table 1 compare DSF-Mamba with the three most closely related prior work along four criteria:

(i) The language and script setting addressed.
(ii) The sequence-modelling backbone employed.
(iii) Whether the study fuses multiple orthographic representations of the same text.
(iv) Whether a gradient-boosted tree baseline is evaluated on identical input features

Therefore, the novelty claimed in the paper is the specific combination of criteria, rather than any individual component considered in isolation.

Table 1. Novelty comparison against the Models

Study	Language or script setting	Sequence backbone	Multi-script fusion	GBM baseline on identical features
Kashmiri new-snippet classification [59]	Kashmiri, Perso-Arabic only	Fine-tuned transformer / LLM	No	**No**
Mamba for long text classification (legal documents) [60]	English, single script	Mamba + segment-level embeddings	No	**No**
Kashmiri Nastalik OCR baseline [61] [62]	Kashmiri, Perso-Arabic only	CNN/CRNN (character recognition, not topic classification.	No	**No**
Proposed Model DSF-Mamba	**Kashmiri, Perso-Arabic + Devanagari transliteration**	**Selective state-space with learned gated fusion.**	**Yes learned, per-timestep gate.**	**Yes, LightGBM and CatBoost on identical pooled features.**

The novelty claimed in the paper is therefore specific and intentionally narrow:

(1) the application of a selective state-space classifier to Kashmiri text classification, which has not been previously reported; and

(2) the Gated Dual-Script Fusion mechanism itself, which combines two script-specific transformer representations of the same sentence via a learned, per-timestep gate prior to sequence modelling, a component with no direct precedent identified in the reviewed literature.

4. Dataset

The corpus used in this study consists of 51,000 Kashmiri sentences written in the Perso-Arabic script, each labelled with one of nine domain categories: Science & Technology, Sports, Tourism, Politics, Education, Health, Finance, Weather and Other. Sentences with missing text or labels outside this fixed class set were filtered out prior to experimentation. The total sentences after cleaning are 50,966 which were partitioned into training and test splits using a stratified 85/15 split (random state fixed for reproducibility), yielding 43,321 training sentences and 7,645 test sentences, with per-class test support ranging from 705 (Weather) to 1,082 (Other), as summarised in Table 2.

Table 2. Class distribution of the held-out test set (n = 7,645).

Class	Test support	Share of test set
Science and Technology	848	11.1%
Sports	840	11.0%
Tourism	835	10.9%
Politics	835	10.9%
Education	834	10.9%
Health	833	10.9%
Finance	833	10.9%
Weather	705	9.2%
Other	1082	14.2%

For dual-script modelling, each Perso-Arabic sentence was additionally transliterated into Devanagari script, providing a second orthographic view of the same underlying text. This dual-script strategy is motivated by the observation that Kashmiri Perso-Arabic orthography is under-standardised in practice, with vowel diacritics frequently omitted; the Devanagari transliteration branch provides a complementary, more phonetically explicit representation that pretrained multilingual encoders like MuRIL, trained extensively on Indic scripts may represent more robustly.

5. Proposed Methodology

The overview of the shared pipeline is shown in Fig 1. in which a Kashmiri sentence is encoded through two frozen, script-specific transformer branches, after which the representations are either pooled for the tree-based models or fused and passed through a selective state-space encoder for DSF-Mamba.

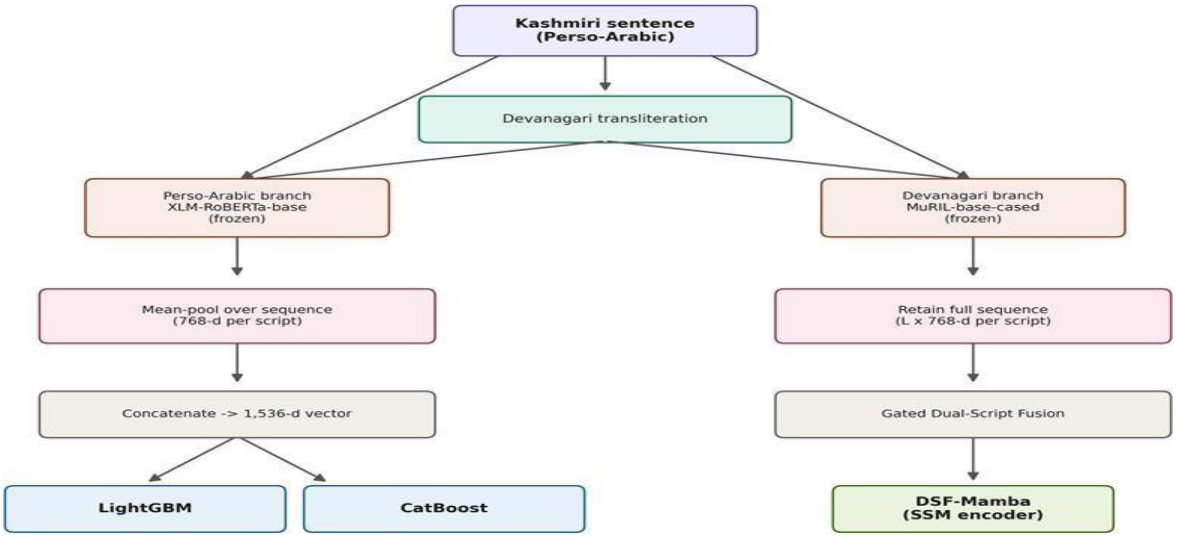


Fig 1. Three Models Pipeline

Figure 2 makes explicit representation about which components of this pipeline are used exactly versus which components were devised specifically for this study. Blue-outlined blocks are standard, unmodified components (pretrained encoders, the transliteration library, the LightGBM and CatBoost libraries themselves); the green-outlined block is adapted from the published selective state-space formulation of Gu and Dao (2023) [63]; the orange-outlined block is a component devised in this study with no direct precedent identified in the reviewed literature.

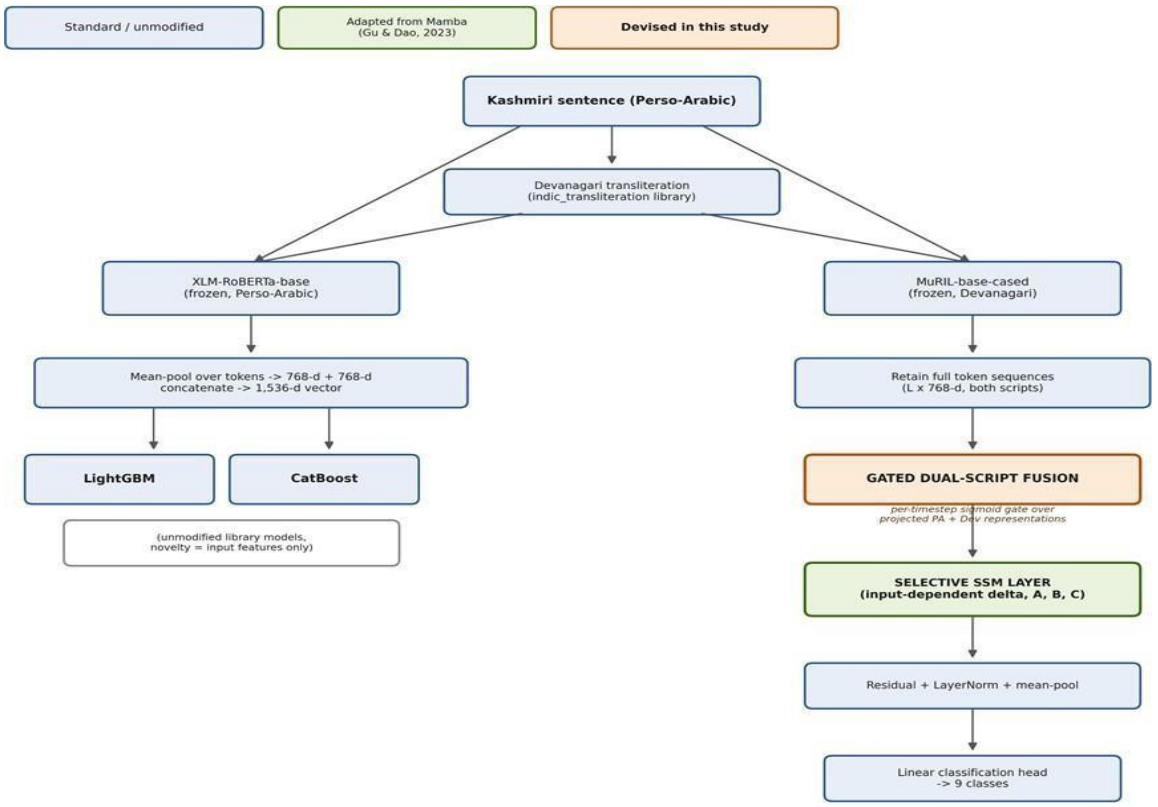


Figure 2. Architecture of the proposed model.

5.1 Shared feature extraction

All three classifiers in this study consume features derived from the same frozen, non-fine-tuned dual-script encoder pipeline, avoiding the substantial compute cost of end-to-end transformer fine-tuning on a corpus of this scale [64]. Two pretrained encoders were used: XLM-RoBERTa-base, applied to the original Perso-Arabic sentence, and MuRIL-base-cased, applied to the Devanagari-transliterated counterpart [65]. Both encoders were loaded with gradients disabled and run-in evaluation mode, having a fixed length of sequence as 32 sub word tokens, which is a configuration to reflect the characteristically short sentence lengths observed in the corpus while keeping the encoding pass computationally tractable at full dataset scale. For the gradient-boosted tree models, each encoder's final hidden states were mean-pooled over the attention mask to produce a single fixed-length vector per sentence per script (768 dimensions each), and the Perso-Arabic and Devanagari pooled vectors were concatenated into a single 1,536-dimensional feature vector per sentence. For the sequence model, full per-token hidden states rather than pooled vectors were retained for both scripts, preserving positional and sub-word-level information for downstream sequence modelling. The algorithm1 used in this method is shown below;

Algorithm 1: Dual-Script Feature Extraction
Input: Kashmiri sentence in Perso-Arabic script
Output: Pooled 1,536-dim feature vector (tree models) or per-token dual-script sequence (DSF-Mamba)
Steps:

1. Transliterate the Perso-Arabic sentence into its Devanagari counterpart.
2. Tokenize both script variants with a maximum sequence length of 32 sub word tokens.
3. Encode Perso-Arabic tokens using frozen XLM-RoBERTa-base (gradients disabled, evaluation mode) to obtain hidden states H_PA.
4. Encode the Devanagari-transliterated tokens using frozen MuRIL-base-cased (gradients disabled, evaluation mode) to obtain hidden states H_DEV.
5. For LightGBM/CatBoost: mean-pool H_PA and H_DEV over the attention mask to obtain two 768-dim vectors and concatenate them into a single 1,536-dim feature vector.
6. For DSF-Mamba: retain the full per-token hidden states H_PA and H_DEV (no pooling) to preserve positional and sub-word-level information.
7. Output the resulting feature representation for downstream classification.

5.2 Gradient-boosted tree baselines: LightGBM and CatBoost

LightGBM was configured as a multiclass classifier (nine classes) using the multi-log loss objective, a rate of learning as 0.05, and 63 leaves per tree, trained for up to 500 boosting rounds with early stopping after 30 rounds without validation improvement. Training converged at iteration 327, with a final validation multi-log loss of 0.3996. The algorithm 2 below depicts the experimentation conducted.

Algorithm 2: For LightGBM
Input: Pooled 1,536-dim dual-script feature vectors X_train, X_test; class labels y_train (9 domain classes.
Output: Predicted topic class and class probabilities
Steps:
1. Initialize a LightGBM multiclass classifier with the multi_logloss objective.
2. Set learning rate = 0.05 and num_leaves = 63.
3. Train for up to 500 boosting rounds with early stopping after 30 rounds without validation improvement.
4. Fit the model on X_train and y_train.
5. Predict class probabilities and labels on X_test.
6. Output predictions and evaluate using Accuracy, Precision, Recall, F1-score.

CatBoost was configured with 500 iterations, tree depth of 8, a learning rate of 0.05, the Multiclass loss function, and early stopping after 30 rounds without improvement in validation accuracy. Training reached its best validation accuracy of 0.8396 at iteration 496, with a widening gap between training accuracy (0.9454) and validation accuracy by the final rounds indicating the onset of mild overfitting consistent with the model's depth and iteration budget [66]. Algorithm 3 shows the steps used in classifier.

Algorithm 3: For CatBoost
Input: Pooled 1,536-dim dual-script feature vectors X_train, X_test; class labels y_train (9 topical classes)
Output: Predicted topic class and class probabilities
Steps:
1. Initialize a CatBoostClassifier with the Multiclass loss function.
2. Set iterations = 500, learning_rate = 0.05, and depth = 8.
3. Enable early stopping after 30 rounds without improvement in validation accuracy.
4. Fit the model on X_train and y_train.
5. Predict class labels on X_test.
6. Output predictions and evaluate using Accuracy, Precision, Recall, F1-score.

5.3 Gated Dual-Script Fusion Mamba (DSF Mamba)

The proposed sequence model, DSF-Mamba, is composed of three stages. First, a Gated Fusion layer independently projects the 768-dimensional Perso-Arabic and Devanagari hidden states into a shared d_model-dimensional space (d_model = 256) and computes a learned, per-timestep sigmoid gate over the concatenation of both projections; the fused representation is a convex combination of the two script-specific projections weighted by this gate, allowing the model to dynamically weight each script's contribution at every token position rather than relying on a fixed combination rule.
Second, the fused sequence is passed through a Selective State-Space (SSM) layer inspired by the Mamba architecture. For each channel, the layer maintains a continuous-time-inspired hidden state updated at every timestep via input-dependent discretisation parameters (delta), input-dependent state-transition dynamics derived from a learned log-diagonal matrix, and input-dependent input/output projection matrices (B and C), combined with a per-channel skip connection (D). This selectivity where the effective dynamics depend on the input at each timestep, rather than being fixed is the key mechanism distinguishing Mamba-style SSMs from earlier linear state-space models.
Third, the SSM output is combined with the fused input via a residual connection and layer normalisation, then mean-pooled over the attention mask and passed through a final linear classification head projecting to the nine target classes. Only the fusion, SSM, normalisation, and classification head parameters are trained; the underlying XLM-RoBERTa and MuRIL encoders remain entirely frozen throughout, meaning the trainable parameter count is orders of magnitude smaller than a fully fine-tuned transformer classifier of comparable input dimensionality. The model was trained with the AdamW optimiser, learning rate 1e-3 and cross-entropy loss, using a state dimension (d_state) of 16 for the SSM layer, for five epochs over the full training set, with sequence embeddings streamed from disk in fixed-size shards to keep memory usage bounded during training on the full corpus.

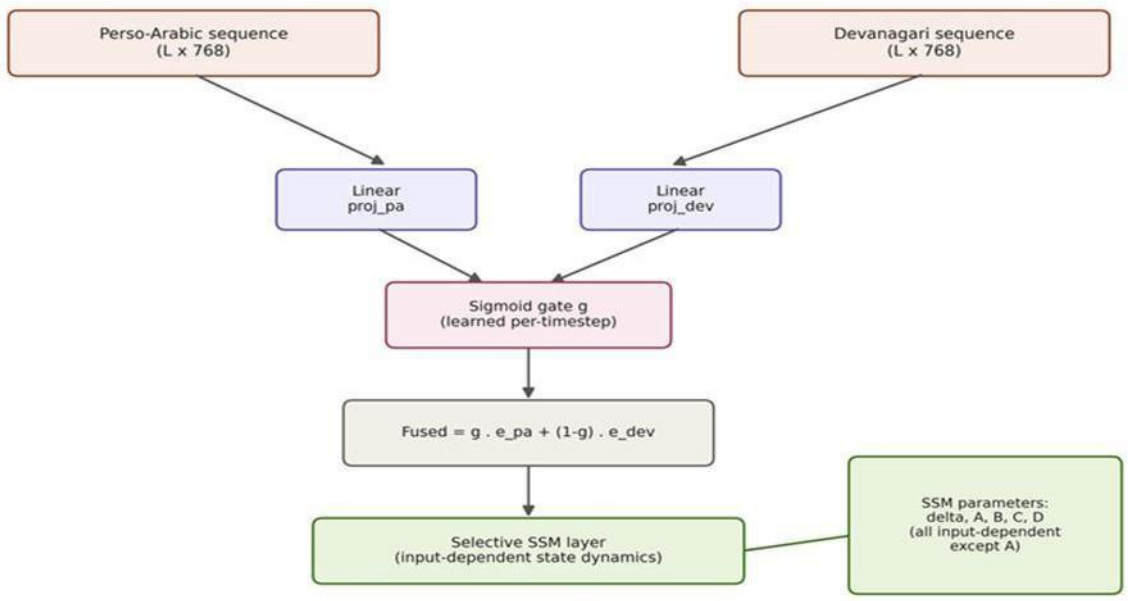


Figure 3. Internal working architecture of our proposed model.

The algorithm used in the proposed model is shown below as algorithm 4.

Algorithm 4: DSF-Mamba Classification (Proposed)

Input: D = (x1,y1), (x2,y2), ..., (xn,yn): dataset of n labelled Kashmiri sentences
xi: a single input sentence (Perso-Arabic).
yi in {0, 1, ..., 8}: topic label across the nine target classes.
Frozen encoders: f_XLM (Perso-Arabic branch), f_MuRIL (Devanagari branch).
Trainable components: gated fusion layer g_fuse, selective state-space layer h_SSM, layer normalization, classification head h_cls.

Output: predicted topic label for a given input sentence

Steps:

1. For each input sentence xi, transliterate it into its Devanagari counterpart x'i.
2. Extract frozen per-token hidden states: H_PA = f_XLM(xi), H_DEV = f_MuRIL(x'i).
3. Compute the gated fusion Z = g_fuse(H_PA, H_DEV) as in Algorithm 4.
4. For each timestep t, update the selective state-space hidden state using input-dependent discretisation delta_t, state-transition parameters derived from a learned log-diagonal matrix A, and input-dependent projections B_t and C_t: s_t = A_bar(delta_t) . s_(t-1) + B_bar(delta_t) . Z_t, and compute the layer output o_t = C_t . s_t + D . Z_t, where D is a per-channel skip connection.
5. Apply a residual connection and layer normalization: O = LayerNorm(Z + [o_1, ..., o_T]).
6. Mean-pool O over the attention mask to obtain a single sentence representation r.
7. Compute class probabilities using the classification head: y_hat = softmax(W_cls . r + b_cls).
8. Train only {g_fuse, h_SSM, LayerNorm, h_cls} using the AdamW optimiser (learning rate = 1e-3) and cross-entropy loss for five epochs, with encoders f_XLM and f_MuRIL kept frozen throughout.
9. Evaluate the trained model using the evaluation metrics.

The overall working architecture of the three models with accuracy results are shown below in fig 4.

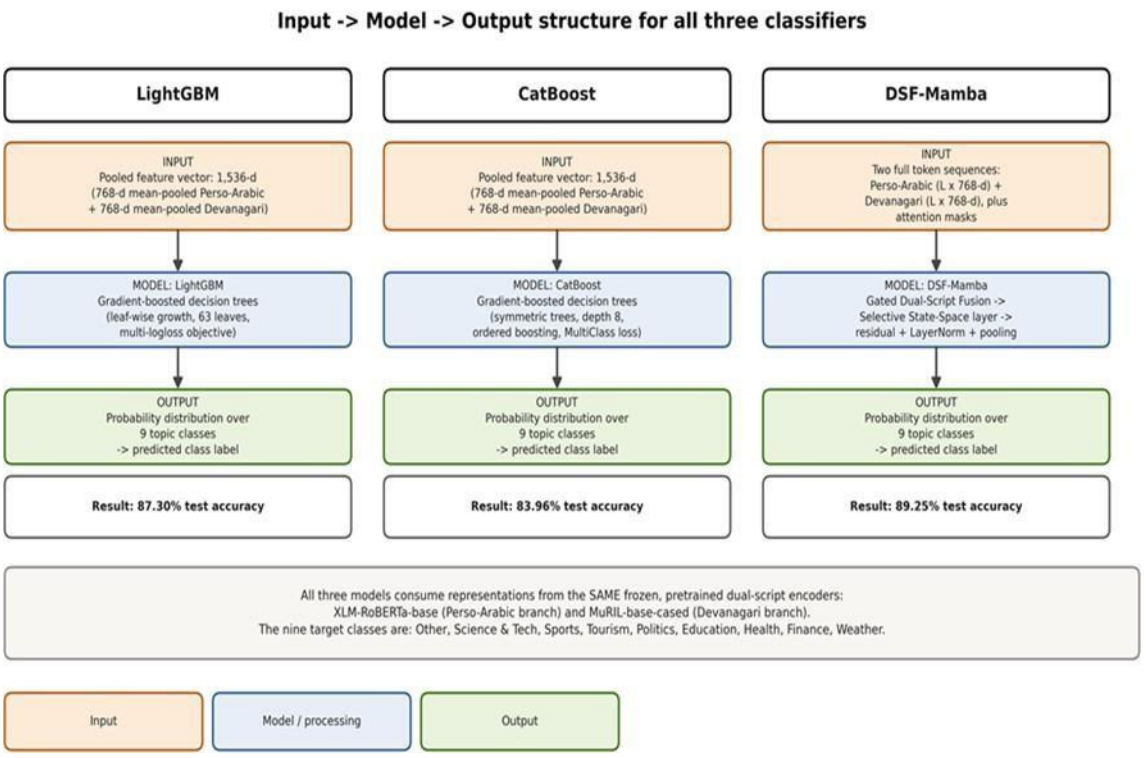


Figure 4. Input-processing-output structure for LightGBM, CatBoost, and DSF-Mamba, with model's overall test accuracy.

6. Mathematical Formulation

This section formalises the gating, fusion, and selective state-space operations used in these methods Notation follows Gu and Dao (2023) where the underlying selective state-space mechanism is adopted directly; the gating and fusion formulation is original for our proposed method.

6.1 Gated dual-script fusion

Let $h^t_PA \in \mathbb{R}^{768}$ and $h^t_DEV \in \mathbb{R}^{768}$ denote the Perso-Arabic and Devanagari hidden states produced by the frozen encoders at token position t = 1, …, L. Each branch is first linearly projected into the shared d_model-dimensional space (d_model = 256):

$$z^t_PA = W_PA\, h^t_PA + b_PA \quad \text{(i)}$$

$$z^t_DEV = W_DEV\, h^t_DEV + b_DEV \quad \text{(ii)}$$

with W_PA, W_DEV $\in \mathbb{R}^{256x768}$. A per-timestep gate is then computed over the concatenation of the two projections:

$$g^t = \sigma(W_g\, [\, z^t_PA\, ;\, z^t_DEV\,] + b_g\,) \in (0,1)^{256} \quad \text{(iii)}$$

where σ is the elementwise logistic sigmoid and W_g $\in \mathbb{R}^{256x512}$. The fused representation passed to the state-space layer is a per-channel convex combination of the two script-specific projections, weighted by the gate:

$$x^t = g^t \odot z^t_PA + (1 - g^t) \odot z^t_DEV \quad \text{(iv)}$$

with ⊙ denoting the elementwise product. Because g^t is computed independently at every timestep from both branches, the model can, in principle, weight the Perso-Arabic branch more heavily at tokens with unambiguous diacritics and shift weight toward the Devanagari branch at tokens where Perso-Arabic orthography is under-specified, rather than applying a single fixed mixing coefficient across the whole sentence.

6.2 Selective State-space layer

The continuous-time state-space system underlying the SSM layer is defined, per channel, by

$$h'(\tau) = A\, h(\tau) + B\, x(\tau) \quad \text{(v)}$$

$$y(\tau) = C\, h(\tau) + D\, x(\tau) \quad \text{(vi)}$$

where A ∈ ℝ ^ (d_model × d_state) is parameterised in log-space as A = −exp(A_log) for training stability, and d_state = 16. The system is discretised via a zero-order hold with an input-dependent step size Δ^t, and B^t, C^t are likewise input-dependent rather than fixed:

$$\Delta^t = \text{softplus}\,(W_\Delta\, x^t + b_\Delta),\ B^t = W_B\, x^t,\ C^t = W_C\, x^t \quad \text{(viii)}$$

$$\bar{A}^t = \exp\,(\Delta^t A),\ \bar{B}^t = \Delta^t B^t \text{ (small-}\Delta\text{ zero-order-hold approximation)} \quad \text{(ix)}$$

giving the discrete-time recurrence actually computed at each timestep:

$$h^t = A^t h_{t-1} + \bar{B}^t x^t, \quad y^t = C^t h^t + D x^t \qquad \text{(x)}$$

It is precisely the input-dependence of Δ^t, B^t and C^t rather than a fixed, input-independent $\bar{A}$, $\bar{B}$, C as in earlier linear state-space models that constitutes the "selective" mechanism distinguishing Mamba-style SSMs, allowing the effective recurrence dynamics to vary token-by-token in response to content rather than being static across the sequence. The final SSM output sequence y^t is combined with the fused input x^t via a residual connection and Layer Norm, mean-pooled over the attention mask, and passed to a linear head $W_cls \in \mathbb{R}^{9x256}$ producing class logits.

6.3 Computational Complexity

Table 3 compares the asymptotic time and memory complexity of the three approaches evaluated in this paper, as a function of sequence length L, model/embedding width d, and for the SSM state dimension n = d_state. Self-attention is included as a reference point because it is the mechanism the selective-SSM literature is most often compared against, even though none of the three classifiers benchmarked here use self-attention directly (the frozen upstream encoders do, but their cost is identical and shared across all three models, and is therefore not a source of differentiation between them). Table 3 shows the time and memory complexity of three classifier mechanisms as a function of sequence length L, embedding width d, SSM state dimension n, and number of boosting trees T.

Table 3. Asymptotic time and memory complexity of the three classifier mechanisms

Mechanism	Time complexity	Memory complexity	Explanation
Self-attention (reference)	$O(L^2 \cdot d)$	$O(L^2)$	Quadratic in sequence length; not used by any classifier head in this study.
Selective SSM (DSF-Mamba, as implemented)	$O(L \cdot d \cdot n)$	$O(d \cdot n)$	Linear in L; the sequential-recurrence implementation used here realises this complexity but not necessarily the fused hardware-aware kernel's wall-clock advantage.
Gradient-boosted trees (LightGBM / CatBoost)	$O(T \cdot depth)$ per sample	$O(1)$ w.r.t. L	Independent of sequence length because both operate on a single pooled 1,536-d vector per sentence, not the full token sequence.

As the corpus in this study uses a maximum sequence length of only 32 sub word tokens, the asymptotic advantage of the linear-in-L selective SSM over quadratic self-attention is not the primary practical driver of any efficiency gain observed here at L = 32 the two-scale similar in practice. The more consequential complexity comparison for this study is between DSF-Mamba/attention-based sequence modelling on one side and pooled gradient-boosted trees on the other: the tree-based models discard within-sequence order at the pooling step and are the cheapest at inference, while DSF-Mamba retains per-token information at the cost of a heavier but still frozen-encoder-dominated forward pass. The linear-in-L complexity of the selective SSM is expected to become the dominant practical consideration if this architecture were applied to longer Kashmiri documents.

6. Experimental Setup

Single-GPU Google Colab runtime was used for the experimentation. To manage compute constraints characteristic of large-scale experimentation on a free or standard-tier cloud runtime, embeddings were computed once and cached to persistent storage rather than recomputed for each model, and long-running extraction and training steps were checkpointed at the batch level so that runtime disconnections did not require full recomputation. The same stratified 85/15 train/test split, the same nine-class label set, and the same frozen dual-script encoder pair were held constant across all three models to isolate the effect of classifier architecture on downstream performance.

Table 4. Training configuration of each model.

Hyper-parameter	LightGBM	CatBoost	DSF-Mamba
Max boosting rounds / epochs	500	500	5 epochs
Learning rate	0.05	0.05	1e-3 (AdamW)
Model capacity	63 leaves	depth 8	d_model=256, d_state=16
Early stopping	30 rounds	30 rounds	Executed without it.
Best iteration / result	327	496	5 epochs (full run)

7. Results

Figure 5 reports overall test-set accuracy for all three models. The DSF-Mamba classifier achieved the highest accuracy at 89.25%, followed by LightGBM at 87.30% and CatBoost at 83.96%.

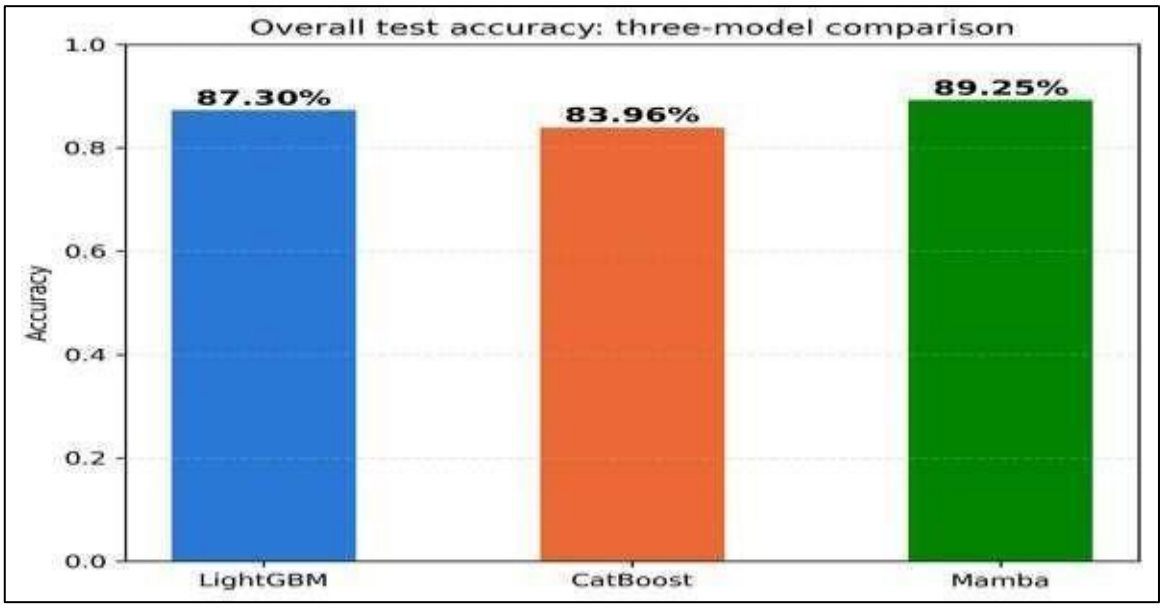


Figure 5. Overall test accuracy across the three models.

F1-score is used as measurement to measure the class wise accuracy of all the three models. Mamba leads on six of the nine classes (Sports, Tourism, Politics, Education, Finance, Other), while LightGBM remains competitive or leading on Science & Tech and is close on several other classes. CatBoost trails both alternatives across nearly every class. Fig 6. Shows the class wise accuracy.

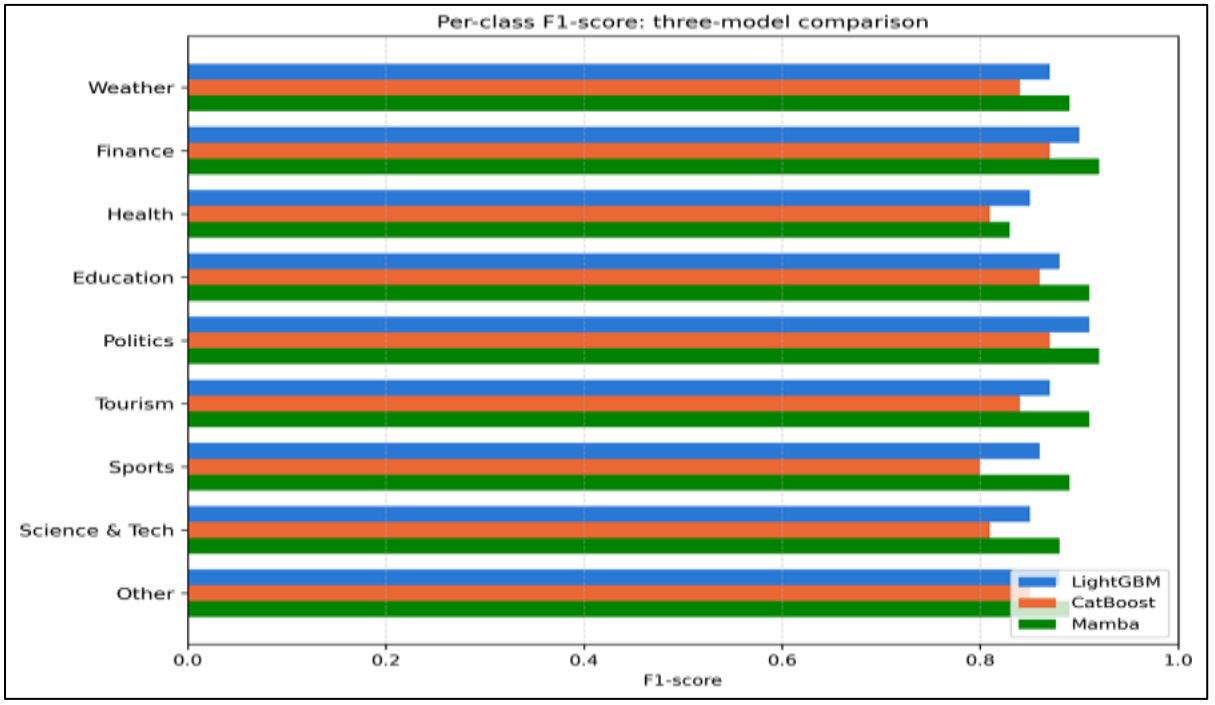
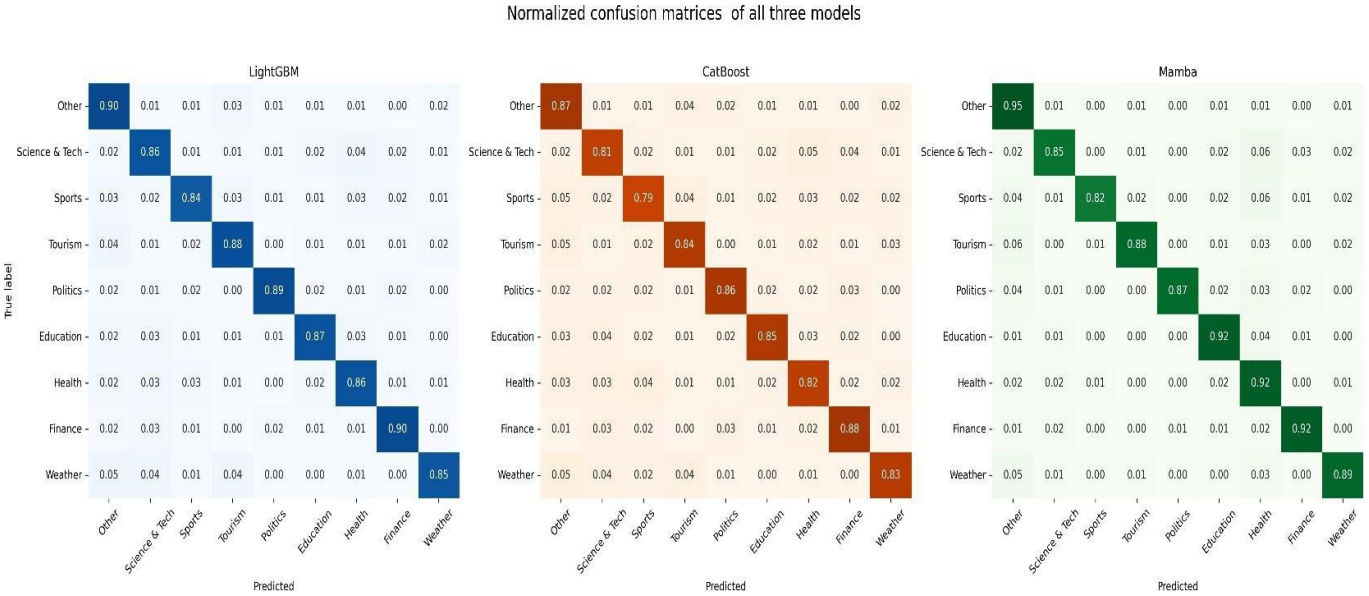


Figure 6. Per-class F1-score comparison.

The normalized confusion matrix of all the three models is shown below in fig 7.



8. Error Analysis

Error analysis was performed to examine the limitations of the proposed DSF-Mamba framework and to identify the factors contributing to classification errors. The analysis is formulated as a set of evidence-based hypotheses rather than an instance-level inspection of misclassified samples.

Hypothesises sources of the Health Precision Drop

The comparatively lower precision for the **health** class is attributed to three plausible factors:

• Inter-class lexical overlap: The health category exhibits substantial semantic and lexical overlap with neighbouring domains such as politics, sports, and weather, primarily due to the widespread use of shared medical terminology, policy-related discourse, and multilingual loanwords. Such overlap reduces class separability and increases the likelihood of false-positive predictions.

• Sequence-Sensitive contextual modelling: Unlike feature-aggregated tree-based baselines, DSF-Mamba preserves sequential contextual dependencies at the token level. While this enhances sensitivity to health-specific contextual cues, it may also amplify the influence of locally discriminative medical expressions embedded within sentences belonging to adjacent domain categories, thereby improving recall at the expense of precision.

• Intrinsic Class-boundary ambiguity: The consistently lower health-class performance across all evaluated architectures indicates that the observed behaviour is not solely model specific but also reflects inherent ambiguity within the corpus annotation and the semantic proximity of the health class to related domains.

9. Discussion

The experimental results demonstrate that DSF-Mamba consistently outperforms the tree-based baselines by combining frozen multilingual contextual representations with lightweight sequence modelling. Unlike conventional classifiers operating on pooled sentence embeddings, DSF-Mamba preserves token-level contextual dependencies before aggregation, enabling more discriminative modelling of semantic relationships while maintaining a relatively small number of trainable parameters.

9.1 Why DSF-Mamba Outperforms the Tree-Based Baselines

The primary performance advantage of DSF-Mamba stems from its ability to model sequential contextual information prior to feature aggregation. Whereas LightGBM and CatBoost operate on fixed pooled sentence embeddings that discard token order, DSF-Mamba processes the complete token sequence through a selective state-space mechanism, preserving contextual interactions across the sentence. This sequential representation enables richer semantic discrimination and more effective separation of closely related domain categories, resulting in superior classification performance.

9.2 Why LightGBM remains Competitive

Despite relying on pooled representations, LightGBM achieves competitive performance due to its efficient leaf-wise tree growth strategy, which effectively models complex nonlinear decision boundaries in the dense transformer embedding space. Furthermore, the combination of gradient boosting, learning-rate regularisation, and early stopping provides robust generalisation, yielding a balanced precision-recall profile across the evaluated classes.

9.3 Why CatBoost Underperforms LightGBM

CatBoost exhibits comparatively lower performance because its principal advantage which is ordered boosting for handling categorical features is not fully utilized, where the input consists exclusively of dense continuous transformer embeddings. Additionally, its symmetric tree architecture offers less representational flexibility that LightGBM's leaf-wise growth for high-dimensional embedding spaces, contributing to mild overfitting and comparatively weaker generalisation performance.

10. Conclusion

The study benchmarks three classifier families i,e. LightGBM, CatBoost, and a novel Gated Dual-Script Fusion Mamba(DSF-Mamba) for multiclass Kashmiri text Classification, all built on a shared, frozen dual-script embedding pipeline that avoids the compute cost of end-to-end transformer fine-tuning. DSF-Mamba achieves the best overall accuracy (89.25%), outperforming LightGBM (87.30%) and CatBoost (83.96%), while training only ~284K parameters atop entirely frozen XLM-ROBERTa and MuRIL Encoders-Mamba leads on six of nine categories but shows a pronounced precision-recall trade-off on health, a pattern largely absent in the tree-based baselines. This is consistent with the sequence model being more sensitive to localised, cross-domain lexical cues that pooled representations average away—evidence that preserving token order provides real classification signal, but also that accuracy alone doesn't capture that full operational trade-off between architectures in a low-resource language. These findings support frozen-embedding sequence models as a favourable accuracy-to-compute alternative to full transformer fine-tuning for low-resource language classification.

11. Future Directions: Three priorities follow directly from this work:

i. Replacing the current per-timestep PyTorch recurrence with a fused, hardware-aware SSM kernel to validate deployment-realistic latency.
ii. An Ablation isolating the Devanagari transliteration branch's contribution from the selective state-space layer's own effect.
iii. Hybrid ensembling of tree-based and sequence-based classifiers to exploit their complementary error profiles, particularly around the health class confusion identified here.

References

1. Lan, M. (2026). Natural language processing and text mining. In *Artificial Intelligence for Drug Design* (pp. 189-215). Singapore: Springer Nature Singapore.
2. Ghojogh, B., & Ghodsi, A. (2026). Attention mechanism and transformers. In *Elements of deep learning* (pp. 231-257). Cham: Springer Nature Switzerland.
3. Kamatala, S., Jonnalagadda, A. K., & Naayini, P. (2025). Transformers beyond nlp: Expanding horizons in machine learning. *Yes it was accepted by IRE Journals*, *8*(7).
4. Xuan, W., Yang, R., Qi, H., Zeng, Q., Xiao, Y., Feng, A., ... & Li, I. (2025, November). Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing* (pp. 1513-1532).
5. Farooq, S., & Bashir, R. (2026). A Novel Stacked Ensemble Framework for Sentiment Classification in Kashmiri. *Journal of Intelligent & Fuzzy Systems*, *50*(5), 1672-1689.
6. Bashir, M. F., Javed, A. R., Arshad, M. U., Gadekallu, T. R., Shahzad, W., & Beg, M. O. (2023). Context-aware emotion detection from low-resource Urdu language using deep neural network. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(5), 1-30.
7. Fields, J., Chovanec, K., & Madiraju, P. (2024). A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *Ieee Access*, *12*, 6518-6531.
8. Choudhary, A., Pamidimokkala, S., & RM, B. (2026). Impact of natural language processing models on diagnosis and decision-making in healthcare, business, education, and sports: a review. *Frontiers in Artificial Intelligence*, *8*, 1706369.
9. Xia, Y., Li, B., Yang, J., Guo, W., Sun, J., Zeng, X., & Zheng, J. (2025). Sustainable service quality assessment of Chinese healthcare e-government: a multi-criteria decision framework based on SERVQUAL model and entropy-weight TOPSIS method. *Frontiers in Digital Health*, *7*, 1611979.
10. Allam, H., Makubvure, L., Gyamfi, B., Graham, K. N., & Akinwolere, K. (2025). Text classification: How machine learning is revolutionizing text categorization. *Information*, *16*(2), 130.
11. Cichosz, P. (2023). Bag of words and embedding text representation methods for medical article classification. *International Journal of Applied Mathematics and Computer Science*, *33*(4), 603-621.
12. Xiao, L., Li, Q., Ma, Q., Shen, J., Yang, Y., & Li, D. (2024). Text classification algorithm of tourist attractions subcategories with modified TF-IDF and Word2Vec. *PloS one*, *19*(10), e0305095.
13. Alayba, A. M., & Altamimi, M. (2025). Optimization of Arabic text classification using SVM integrated with word embedding models on a novel dataset. *Int. j. adv. appl. sci*, *12*, 140-151.

14. Karaoglan, K. M., & Findik, O. (2024). Enhancing aspect category detection through hybridised contextualised neural language models: A case study in multi-label text classification. *The Computer Journal*, *67*(6), 2257-2269.
15. Subramanian, M., Veerappampalayam Easwaramoorthy, S., Subbarayan, N., & Moorthy, U. (2025). Empowering sentiment analysis in social media: a comprehensive approach to enhance the classification of abusive Tamil comments using transformer models. *Journal of Big Data*, *12*(1), 208.
16. Anushka, C. V., John, A. K., Nair, A. R., Gupta, D., & Veena, G. (2025). Comprehensive Integration of Machine Learning, Deep Learning, and Fine-Tuned Large Language Models for Malayalam News Categorization. *Procedia Computer Science*, *258*, 3357-3368.
17. Noman, H., & Ahmed, S. (2026). MISINFORMATION DETECTION IN LOW-RESOURCE LANGUAGES USING MULTILINGUAL TRANSFORMERS AND EXPLAINABLE CLASSIFICATION. *Computers and Education Letters*, *3*(01), 31-42.
18. Shah, A., & Singh, M. (2026). One Instruction Does Not Fit All: How Well Do Embeddings Align Personas and Instructions in Low-Resource Indian Languages? *arXiv preprint arXiv:2601.10205*.
19. Gopinathan, S. V., & Manzoor, M. F. (2025). Efficient Multi-Class Analysis of Consumer Complaints Using Frozen MiniLM Embeddings and Neural Networks. *International Journal of Advanced Computer Science & Applications*, *16*(12).
20. Liu, M., Liu, Q., Gong, X., Luo, Y., & Wang, G. (2025). Mol-L2: Transferring text knowledge with frozen language models for molecular representation learning. *Neurocomputing*, 130837.
21. Ghosal, S., & Jain, A. (2024). CatRevenge: towards effective revenge text detection in online social media with paragraph embedding and CATBoost. *Multimedia Tools and Applications*, *83*(42), 89607-89633.
22. Sundaram, G. (2026). A Personalized and Privacy-Preserving Federated Transformer Framework for Multilingual Sentiment Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
23. Lone, N. A. (2022). Natural language processing resources for the Kashmiri language. *Indian journal of science and technology*.
24. Apidianaki, M. (2023). From word types to tokens and back: A survey of approaches to word meaning representation and interpretation. *Computational Linguistics*, *49*(2), 465-523.
25. Qumar, S. M. U., Azim, M., & Quadri, S. M. K. (2025). Enhancing low-resource neural machine translation with decoding-based data augmentation. *International Journal of Information Technology*, 1-13.
26. Lavanya, C. B., Nagendraswamy, H. S., Manjunath, K. N., & Kumar, M. P. (2026). Machine Translation in Natural Language Processing (1948–2025): a Systematic Survey of Methods, Benchmarks, Neural Machine Translation Advances, Challenges and Future Directions. *SN Computer Science*, *7*(6), 596.
27. Mir, T. A., & Lawaye, A. A. (2025). Word sense disambiguation corpus for Kashmiri. *Natural Language Processing*, *31*(2), 631-654.
28. Deyar, D. U., Ramani, A., Gupta, D., Nair, P. C., & Venugopalan, M. (2025). Dataset creation and benchmarking for Kashmiri news snippet classification using fine-tuned transformer and LLM models in a low resource setting. *Scientific Reports*, *15*(1), 40828.
29. Ul Qumar, S. M., Azim, M., Quadri, S. M. K., Alkanan, M., Mir, M. S., & Gulzar, Y. (2025). Deep neural architectures for Kashmiri-English machine translation. *Scientific Reports*, *15*(1), 30014.
30. Bibi, S., Asghar, S., & Zubair, M. (2024). Sense unveiled: Enhancing Urdu corpus for nuanced word sense disambiguation. *IEEE Access*, *12*, 126329-126343.
31. Kaddour, J., Key, O., Nawrot, P., Minervini, P., & Kusner, M. J. (2023). No train no gain: Revisiting efficient training algorithms for transformer-based language models. *Advances in Neural Information Processing Systems*, *36*, 25793-25818.
32. Bansal, V., Tyagi, M., Sharma, R., Gupta, V., & Xin, Q. (2022). A transformer-based approach for abuse detection in code-mixed indic languages. *ACM transactions on Asian and low-resource language information processing*.
33. Chattu, K., Reddy, K. A. N., Veesam, S. B., Chirumamilla, P. S., Dinesh Babu, V., Prakash, K., ... & Al-Mugren, K. S. (2025). Sentiment classification for telugu using transformed based approaches on a multi-domain dataset. *Scientific Reports*, *15*(1), 22185.
34. Ingale, O., & Margaj, S. (2025). Comparative Analysis of Embedding Models for Hindi-English Code-Mixed University related queries. *The Voice of Creative Research*, *7*(2), 362-369.
35. Khanduja, N., Kumar, N., & Chauhan, A. (2026). Text Classification Techniques for Low-Resource Indian Languages: Challenges and Trends. *Technological Advancements in Engineering and Management Sciences*, 193-221.

36. Hossain, M. Z., & Goyal, S. (2024). Advancements in natural Language processing: Leveraging transformer models for multilingual text Generation. *Pacific Journal of Advanced Engineering Innovations*, *1*(1), 4-12.
37. Shen, L., Hao, T., He, T., Zhang, Y., Liu, P., Zhao, S., ... & Ding, G. (2025). Temporal modelling with frozen vision–language foundation models for parameter-efficient text–video retrieval. *IEEE Transactions on Neural Networks and Learning Systems*.
38. Cheng, P., Xia, M., Wang, D., Lin, H., & Zhao, Z. (2025). Transformer self-attention change detection network with frozen parameters. *Applied Sciences*, *15*(6), 3349.
39. Wang, Z., Li, J., Zhou, H., Weng, R., Wang, J., Huang, X., ... & Huang, S. (2025, July). Investigating and scaling up code-switching for multilingual language model pre-training. In *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 11032-11046).
40. Doddapaneni, S., Ramesh, G., Khapra, M., Kunchukuttan, A., & Kumar, P. (2025). A primer on pretrained multilingual language models. *ACM Computing Surveys*, *57*(9), 1-39.
41. Qu, H., Ning, L., An, R., Fan, W., Derr, T., Liu, H., ... & Li, Q. (2026). A survey of mamba. *ACM Transactions on Intelligent Systems and Technology*, *17*(5), 1-43.
42. Tang, J., Zhou, X., & Yan, Q. (2026). A Mamba State-Space Sequence Model for AI-Driven Dynamic Aggregation and Predictive Control of Electric Vehicle Clusters in Vehicle-to-Grid Energy Management. *Electronics*, *15*(11), 2380.
43. Rezaei Jafari, F., Montavon, G., Müller, K. R., & Eberle, O. (2024). Mambalrp: Explaining selective state space sequence models. *Advances in neural information processing systems*, *37*, 118540-118570.
44. Ileri, K. (2025). Comparative analysis of CatBoost, LightGBM, XGBoost, RF, and DT methods optimised with PSO to estimate the number of k-barriers for intrusion detection in wireless sensor networks. *International Journal of Machine Learning and Cybernetics*, *16*(9), 6937-6956.
45. Abdelmotaleb, H., Mcneile, C., & Wojtyś, M. (2025). A comparative study of word embedding techniques for classification of star ratings. *Expert Systems with Applications*, 129037.
46. Feng, K., Huang, L., Wang, K., Wei, W., & Zhang, R. (2024). Prompt-based learning framework for zero-shot cross-lingual text classification. *Engineering Applications of Artificial Intelligence*, *133*, 108481.
47. Liu, W., Xiang, M., & Ding, N. (2026). Active use of latent tree-structured sentence representation in humans and large language models. *Nature Human Behaviour*, *10*(2), 303-316.
48. Wang, S., Gai, K., Yu, J., Zhang, Z., & Zhu, L. (2025). PraVFed: Practical heterogeneous vertical federated learning via representation learning. *IEEE Transactions on Information Forensics and Security*, *20*, 2693-2705.
49. Sheshadri, S. K., Gupta, D., Paul, B., & Bhavani, J. S. (2025). A Novel Approach to Continual Knowledge Transfer in Multilingual Neural Machine Translation Using Autoregressive and Non-Autoregressive Models for Indic Languages. *IEEE Access*.
50. Miloudi, A., Bouhamed, M. M., Laouid, A., Boulesnane, A., Khelaifa, A., & Derdour, M. (2026, April). Algerian Arabic Dialects in Natural Language Processing: Challenges and Perspectives. In *2026 8th International Conference on Pattern Analysis and Intelligent Systems (PAIS)* (pp. 1-8). IEEE.
51. Pakray, P., Gelbukh, A., & Bandyopadhyay, S. (2025). Natural language processing applications for low-resource languages. *Natural Language Processing*, *31*(2), 183-197.
52. Raza, M. O., Mahoto, N. A., Shaikh, A., Pathan, N., Alshahrani, H., & Elmagzoub, M. A. (2025). A Machine Learning Approach of Text Classification for High-and Low-Resource Languages. *Computational Intelligence*, *41*(4), e70114.
53. Ivačič, N., Škrlj, B., Koloski, B., Pollak, S., Lavrač, N., & Purver, M. (2025). Extreme Multi-Label Text Classification for Less-Represented Languages and Low-Resource Environments: Advances and Lessons Learned. *Machine Learning and Knowledge Extraction*, *7*(4), 142.
54. Gedela, R. T., Baruah, U., & Soni, B. (2024). Deep contextualised text representation and learning for sarcasm detection. *Arabian Journal for Science and Engineering*, *49*(3), 3719-3734.
55. Mamtani, S. (2025). Integrating graph-based representations with deep contextual models for text classification. *Computer Science & Information Technology (CS & IT)*.
56. Muca Cirone, N., Orvieto, A., Walker, B., Salvi, C., & Lyons, T. (2024). Theoretical foundations of deep selective state-space models. *Advances in Neural Information Processing Systems*, *37*, 127226-127272.
57. Ping, W., Jiao, Y., Fan, H., & Zhang, X. (2026). Multimodal fraud detection in financial statements: A trimodal attention network with contrastive evidence chain construction. *IEEE Access*.
58. Dong, H., & Kotenko, I. (2026). A lightweight and robust approach to IoT intrusion detection based on ensemble deep learning. *International Journal of Parallel, Emergent and Distributed Systems*, 1-21.

59. Navya, D., Varshini, P. H., Deyar, D. U., & Gupta, D. (2026). Exploring OCR for the Low-Resource Dogri Language: A Deep Learning Approach Enabled by Comprehensive Dataset Creation and Systematic Evaluation. *Procedia Computer Science*, *282*, 3062-3072.
60. Song, B., Xu, Y., Liang, P., & Wu, Y. (2024). State space models based efficient long documents classification. *Journal of Intelligent Learning Systems and Applications*, *16*(3), 143-154.
61. Fayaz, S. A., Khaja, M. M., Bhat, A. S., Mansoor, D., Thapa, A., & Zaman, M. (2026). A baseline optical character recognition framework for printed Kashmiri Nastaliq text using deep learning. *Acadlore Transactions on AI and Machine Learning*, *5*(2), 119-137.
62. Gull, S., Ahmed, N., & Sheikh, S. A. (2025). Deep Learning-Driven OCR System for Brahui Printed Text: Bridging the Digital Gap in Low-Resource Language Processing. *Liberal Journal of Language & Literature Review*, *3*(3), 1174-1198.
63. Dao, T., & Gu, A. (2024). Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*.
64. Zhao, H., Sui, D., Wang, Y., Ma, L., & Wang, L. (2025). Privacy-preserving federated learning framework for multi-source electronic health records prognosis prediction. *Sensors*, *25*(8), 2374.
65. Saleem, H., Javed, M., & Khan, J. (2025). Hate Speech Identification in Formal and Informal social media Text Using RoBERTa-Base and XLM-RoBERTa-Base Models. *BRAIN: Broad Research in Artificial Intelligence & Neuroscience*, *16*(4).
66. Shahzad, D. K., Tariq, M. U., Akhtar, N., & Shah, A. A. (2025). Gradient Boosted and Ensemble-Led Enhancements for Hierarchical Intrusion Detection: Addressing Class Imbalance Using CatBoost, LightGBM, and Ensemble Methods. *LightGBM, and Ensemble Methods (December 16, 2025)*.