 http://dx.doi.org/10.14500/aro.12814

The First Comprehensive Study of Stance Detection Modeling for the Sorani Kurdish Language

Payman S. Rostam[1†], Rebwar M. Nabi[2,3]

[1]Department of Information Technology, Computer Science Institute, Sulaimani Polytechnic University, Sulaimani, Kurdistan Region, Iraq

[2]Department of Information Technology, Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, Kurdistan Region, Iraq

[3]Department of Information Technology, Kurdistan Technical Institute, Sulaimani, Kurdistan Region, Iraq

***Abstract*—Stance detection has become a fundamental task in natural language processing (NLP), yet it remains under-explored for low-resource languages such as Sorani Kurdish. Building on the previously released Bochun dataset, the present work focuses exclusively on the comprehensive evaluation of stance detection models and provides the first benchmarking of classical machine learning, deep learning (DL), and transformers. Eight models are implemented: Support vector machine (SVM), logistic regression, random forest, extreme gradient boosting, convolutional neural network, bidirectional long short-term memory, and two frozen transformer encoders, Central Kurdish Bidirectional Encoder Representations from Transformers (BERT) and XLM-RoBERTa-base (XLM-R), each combined with a logistic-regression classification head. All models are trained and evaluated under four protocols (80/20 and 70/30 stratified splits and 5-fold and 10-fold cross-validation) with five random seeds. Class-imbalance handling is investigated through a dedicated ablation comparing no weighting, model-appropriate weighting, and Synthetic Minority Oversampling Technique oversampling, and pairwise statistical comparisons are conducted using Wilcoxon signed-rank tests complemented by Cohen's d effect sizes. With the chosen feature representations and dataset size, the SVM trained on term frequency–inverse document frequency features delivers the strongest performance (accuracy and weighted F1 of 74% on the 80/20 split and 72% under 10-fold cross-validation), outperforming the DL and frozen-transformer baselines. The findings provide a foundational reference point for future research on Kurdish NLP in general and stance detection specifically.**

***Index Terms*—Cross-lingual learning, Kurdish Sorani language, Low-resource languages, Natural language processing, Stance detection, Transformers.**

I. Introduction

Automated stance detection has gained significant attention in the domain of natural language processing (NLP) in recent years and remains one of the fundamental challenges that researchers strive to address. The task is to determine whether a piece of text supports, opposes, or remains neutral on a given target. The problem is vital for a wide range of applications, including social-media monitoring, political analysis, and the verification of misinformation. Considerable progress has been achieved for high-resource languages, but the problem remains poorly addressed for low-resource languages that lack annotated datasets and computational tools (Mets *et al.*, 2024).

Sorani Kurdish is one of the two major dialects of the Kurdish language and is spoken by millions of speakers across the Middle East, yet it has been historically under-represented in the NLP community. Sorani has rich morphology, split-ergative syntax, and considerable dialectal variation, all of which present standard challenges for computational modeling. The scarcity of annotated datasets, lexicons, and standardized tools for Kurdish further limits the development of robust language technologies for this linguistic family (Aziz, *et al.*, 2024). Kurdish, especially its Sorani dialect, is a clear example of an under-resourced language whose morpho-syntactic features make NLP tasks particularly difficult to solve (Azad, Ahmed and Saeed, 2025). Despite Kurdish's significant speaker base and growing presence in online media, few publicly available datasets exist to support automated stance detection for the language, limiting research and the development of practical applications for analyzing Kurdish text (Article and Badawi, 2023).

While deep learning (DL) and transformer-based models have driven substantial progress for high-resource languages (Aslam et al., 2025; Bharathi and Zubiaga, 2025), whether such gains transfer to low-resource Kurdish—where pretraining data and downstream task data are both limited—remains an open empirical question that the present work addresses directly.

Building on this background, the contribution of this paper must be carefully delineated against previously published work. The Bochun dataset was introduced and described in a separate publication (Rostam and Nabi, 2025), which

ARO-The Scientific Journal of Koya University
Vol. XIV, No.2 (2026), Article ID: ARO.12814. 10 pages
DOI: 10.14500/aro.12814
Received: 02 January 2026; Accepted: 19 June 2026
Regular research paper; Published: 15 August 2026
†Corresponding author's e-mail: payman.sabr.rostam@spu.edu.iq

addressed dataset creation, lexicon development (2,456 target keywords and 4,243 stance phrases), automatic labeling pipeline design, and human validation of label quality.

The present paper makes a strictly different and complementary contribution: It focuses on the rigorous evaluation, comparison, and benchmarking of stance-detection models on the established Bochun resource. Specifically, this work (i) Benchmarks six classical machine learning (ML) and DL baselines together with two frozen transformer encoders (Central Kurdish BERT **(**KuBERT) and Cross-lingual Language Model – RoBERTa(XLM-R)) under four evaluation protocols and five random seeds; (ii) provides a dedicated ablation study isolating the effect of class-imbalance handling on each model; (iii) reports pairwise Wilcoxon signed-rank tests and Cohen's d effect sizes to support claims about model differences statistically; and (iv) discusses the conditions under which classical linear classifiers remain competitive with neural and transformer baselines on a small, morphologically rich Kurdish dataset. The dataset construction and validation steps are not the contribution of this paper but are summarized in Section III for reproducibility and self-contentedness.

II. Literature Review

Research on stance detection has expanded considerably in recent years, exploring how textual data convey supportive, opposing, or neutral attitudes toward a given target. Early works such as SemEval-2016 (Mohammad *et al.*, 2016) established a basis for stance detection on English social media, in which both classical feature-based and early DL models (e.g., Convolutional neural network [CNN], recurrent neural network) reached comparable performance levels (F1 ≈ 67%).

In Arabic, several high-quality datasets, including MAWQIF (Alturayeif, Luqman and Ahmed, 2022) and AraStance (Alhindi, *et al.*, 2021), achieved an F1-score of about 79%. Likewise, the broad range of multilingual and cross-lingual resources in X-Stance (Vamvas and Sennrich, 2020) also have a moderate degree of success in transferring the knowledge of stance across languages (macro-F1 ≈ 76%). Follow-up work in Turkish, Persian, Hindi-English, and Czech broadened stance detection to morphologically rich and low-resource settings and found that even hybrid neural architectures significantly outperform traditional baselines for contextual understanding.

Nevertheless, low-resource languages such as Kurdish have minimal coverage in this area, lacking dedicated stance-detection datasets and consistent benchmarks. Table I provides an overview of representative stance-detection datasets and methods across languages.

III. Materials and Methods

A. Dataset Description

This study utilizes the Bochun dataset (Rostam and Nabi, 2025), a Sorani Kurdish stance detection dataset containing 2,174 unique economic and political articles collected from Rudaw between March 2024 and March 2025. While this paper builds on that empirical data, it omits the specific details regarding the dataset's collection, lexicon construction, and automated three-stage labeling pipeline, which are documented in the original publication. The final stance-label distribution, overall and by domain, is reported in Table II.

As detailed in Table II, the dataset exhibits moderate class imbalance – particularly within the economy subset (56.39% support vs. 11.22% neutral overall) – which motivates the use of stratified evaluation and the ablation studies in Section IV. To ensure label reliability, the original publication's automated pipeline was validated against human expert annotations (four native Sorani Kurdish speakers specializing in economics and politics). Inter-annotator agreement (IAA) on a sample of 700 articles yielded almost-perfect consensus (Cohen's $\kappa = 0.90$ for economy, $\kappa = 0.93$ for politics), outperforming comparable benchmarks like MAWQIF ($\kappa \approx 0.85$). For computational efficiency and focus, all models in this study utilize the lexically dense article titles rather than the full body text, a design choice evaluated further in Section IV.

B. Proposed Modeling Pipeline and Hyperparameter Configuration

This section presents the modelling pipeline of the present study: Language-specific preprocessing of the article titles, three feature-representation strategies, and eight classifiers spanning classical ML, DL, and two frozen transformer encoders. The complete hyperparameter configuration of all eight models is consolidated in Table III. The Sorani Kurdish stance detection framework follows a multi-stage pipeline that combines language-specific preprocessing, three feature representation strategies, and a heterogeneous set of classification architectures (classical, DL, and frozen transformer).

Data preprocessing

Article titles are preprocessed with the Kurdish Language Processing Toolkit (KLPT) (Ahmadi, 2020) before feature extraction. The pipeline comprises four steps: (i) Unicode normalization of alternative character forms and numerals to a single canonical form, to prevent encoding mismatches during tokenization; (ii) word-level tokenization; (iii) stop-word removal of common function words (e.g., لە ,و); and (iv) stemming, which collapses morphological variants to a common root (for example, دەنگدان "voting" and دەنگدەران "voters" both stem to دەنگدان). These steps homogenize the input and reduce surface-form variation, which is particularly important for a morphologically rich language such as Kurdish.

All preprocessing decisions affecting feature representation – the Term Frequency–Inverse Document Frequency (TF–IDF) vocabulary, IDF weights, vector-space dimension, and hyperparameter values – were applied strictly inside each training fold or training split. TF–IDF vectorizers were fit on the training portion only and then applied to the test portion, ensuring that no test-set information leaked into the

TABLE I
COMPARATIVE SUMMARY OF STANCE DETECTION DATASETS ACROSS LANGUAGES

Language	Study	Annotation type	Size	Methods used	Results (%)
English	SemEval-2016 Task 6: Detecting Stance in Tweets (Mohammad *et al.*, 2016)	Manual	4163 tweets	SVM, CNN, recurrent neural network	F1≈67
Arabic	MAWQIF: Multi-label Arabic Stance Detection (Alturayeif, Luqman and Ahmed, 2022)	Manual (crowdsourced)	4121 tweets	AraBERT, BERT	Macro-F1≈78.9
Arabic	AraStance: Multi-Country and Multi-Domain (Alhindi, *et al.*, 2021)	Manual	4063 articles	AraBERT, MARBERT, mBERT	Macro-F1≈78
Multilingual	X-stance: A Multilingual Multi-Target Dataset (Vamvas and Sennrich, 2020)	Manual	67271 sentences	mBERT	Macro-F1≈76
Catalan/Spanish	Multilingual Stance Detection: The Catalonia Independence Corpus (Zotova *et al.*, 2020)	Manual	20125 tweets	fastText, linear classifiers	Macro-F1≈70–72
Turkish	Stance Detection in Turkish Tweets (Küçük, 2017)	Manual	700 tweets	SVM	F1≈80–83
Persian	Stance detection on social media, case study: Persian sentences using deep learning architecture (Mohammadi, Farzi and Alavi, 2024)	Manual	3167 posts	BiLSTM+CNN+ BERT	Accuracy≈86.8
Hindi-English	Stance Detection in Code-Mixed Hindi-English Social Media Data using Multi-Task Learning (Sane *et al.*, 2021)	Manual	3545 tweets	Multi-task CNN	Accuracy≈63.2
Czech	Detecting Stance in Czech News Commentaries (Hercig *et al.*, 2017)	Manual	2638 comments	SVM, CNN	Accuracy≈65–70
Italian (Sardinian)	SardiStance@EVALITA2020: Overview of the task on stance detection in Italian tweets (Cignarella *et al.*, 2020)	Manual	3242 tweets	BiLSTM, CNN	F1≈68–74

CNN: Convolutional neural network, SVM: Support vector machine, BiLSTM: Bidirectional long short-term memory, BERT: Bidirectional Encoder Representations from Transformers, AraBERT: Arabic Transformer-based Model for Arabic Language Understanding (Antoun, Baly, and Hajj, 2020), MARBERT: Deep Bidirectional Transformers for Arabic (Abdul-Mageed, Elmadany, and Nagoudi, 2020)

TABLE II
STANCE LABEL DISTRIBUTION ACROSS THE DATASET AND BY DOMAIN CATEGORY

Stance	Overall count	Overall %	Economy count	Economy %	Politics count	Politics %
Support	1,226	56.39	860	60.99	366	47.91
Oppose	704	32.38	431	30.57	273	35.73
Neutral	244	11.22	119	8.44	125	16.36
Total	2,174	100.00	1,410	100.00	764	100.00

feature representation. The FastText embedding matrix used for the DL models was fixed before splitting and was not re-fit per fold; the frozen transformer encoders (KuBERT and XLM-R) were likewise held fixed across all folds, with only the logistic-regression head re-fit per training partition. Hyperparameter selection for the classical models used grid search nested inside the training partition, and early stopping for the DL models was driven by a validation subset drawn from the training partition only.

Feature representation strategies

Three feature-representation strategies were used. First, for the classical classifiers (logistic regression [LR], support vector machine (SVM), random forest [RF], extreme gradient boosting [XGBoost]), titles were converted into sparse TF–IDF vectors (Zhang, Yoshida and Tang, 2011), capturing lexical patterns indicative of stance. Second, for the DL classifiers (CNN, long short-term memory [BiLSTM]), titles were tokenized into integer sequences and mapped to dense embeddings initialized from pre-trained Kurdish FastText vectors (Saeed, 2024) with subword-level character n-gram information (n = 3–6). Third, for the transformer baselines, titles were tokenized using each model's native tokenizer and passed through the frozen encoder, with mean-pooled sequence representations extracted as features for a downstream classifier. The transformer methodology is detailed in Section (Transformer Baselines).

Training strategy and imbalance management

Once features had been extracted, both classical ML and DL models were trained to classify each article into one of three stance classes: Support, oppose, or neutral. To ensure robustness, models were evaluated under stratified train–test splits (80/20 and 70/30) and stratified k-fold cross-validation (5-fold and 10-fold), with results averaged over five random seeds (1, 42, 123, 2024, and 9999). To address class imbalance, model-appropriate weighting schemes were applied during training: SVM, LR, and RF used class_weight = 'balanced' (inverse-frequency weights), XGBoost used per-sample weights derived from inverse class frequencies, and CNN and BiLSTM used focal loss combined with class weights to give greater attention to harder, minority-class samples (Aljohani, Fayoumi and Hassan, 2023). Multiple regularization techniques mitigated overfitting in the DL models: Dropout (rate 0.3–0.5), L2 weight regularization, and early stopping on validation loss with a patience of three epochs. Random shuffling of training data each epoch helped improve generalization (Xie *et al.*, 2022). Section IV reports a dedicated ablation isolating the effect of these imbalance-handling choices.

TABLE III
HYPERPARAMETER CONFIGURATION FOR ALL MODELS

Model	Parameter	Value/Setting
Logistic regression	Solver	lbfgs
	Regularization	L2 (C=1.0)
	Class weight	balanced
Support vector machine	Kernel	Linear
	Regularization	C=1.0
	Max iterations	1,000
	Probability	True
	Class weight	balanced
RF	Estimators (trees)	100
	Class weight	balanced
XGBoost	Max depth	20
	Objective	multi: softprob
	Evaluation metric	mlogloss
	Class balance	Per-sample inverse-frequency weights
CNN and BiLSTM	Embedding dimension	300 (FastText cc.ckb. 300.bin)
	Vocabulary size	10,000
	Sequence length	150 tokens (truncated/padded)
	Optimizer	Adam (lr=1 × 10^{-3})
	Loss function	Focal loss (γ=2.0, α=0.25)
	Batch size	32
	Epochs	Up to 20
	Regularization	L2 (0.001), Dropout (0.5)
	Early stopping	Patience=3 (restore best weights)
CNN	Filters	128, kernel size 5, ReLU
	Pooling	Global max-pooling
	Dense head	64 ReLU+softmax (3)
BiLSTM	Hidden units (per direction)	128 (256 total)
	Return sequences	True (with global max-pooling)
	Dense head	64 ReLU+softmax (3)
KuBERT	Architecture	Frozen encoder+logistic regression head
	Tokenizer	Native KuBERT WordPiece
	Pooling	Mean over non-pad tokens
	Sequence length	128 subword tokens
XLM-R	Architecture	Frozen XLM-RoBERTa-base+logistic regression head
	Tokenizer	Native XLM-R SentencePiece
	Pooling	Mean over non-pad tokens
	Sequence length	128 subword tokens

BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, XGBoost: Extreme gradient boosting

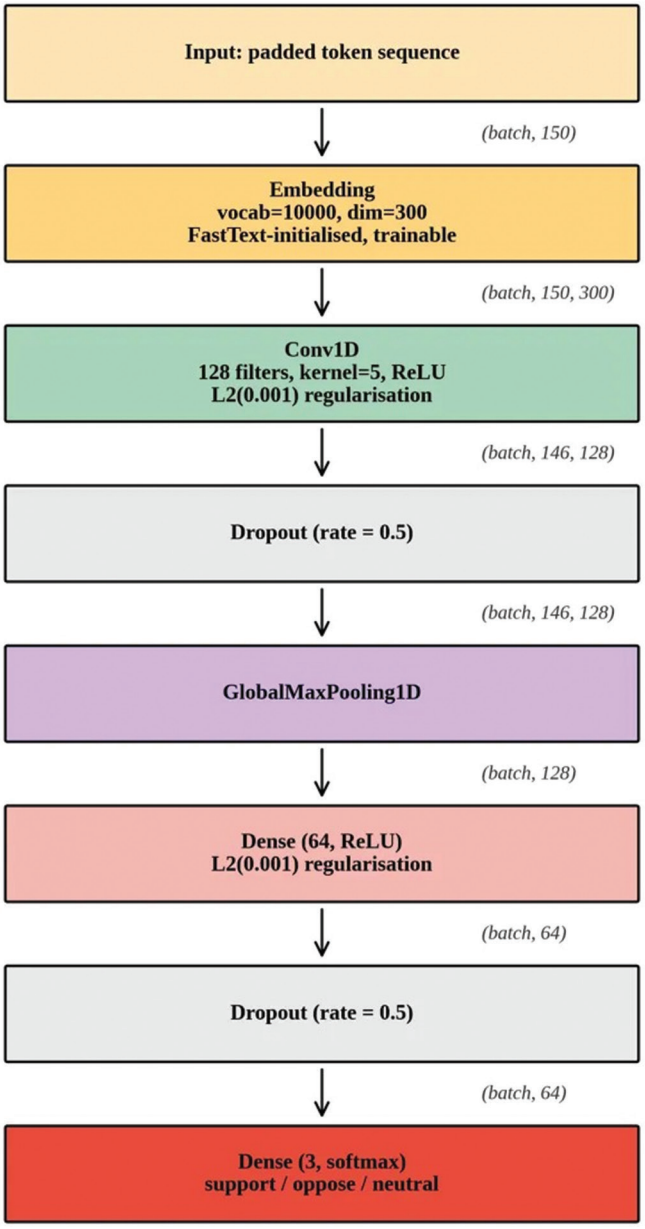

Fig. 1. Architecture of the convolutional neural network model adapted to Sorani Kurdish text classification. Annotations on the right indicate the tensor shape after each layer.

CNN and BiLSTM architectures

Both the CNN and BiLSTM models were trained from scratch (no transfer learning was used for the DL baselines) and were adapted to the Kurdish text classification setting through a 1D convolution over embeddings design. The shared input pipeline produces a fixed-length integer sequence of 150 tokens (longer titles are truncated; shorter titles are right-padded with a dedicated pad index). Each token is mapped through an embedding layer of dimension 300, initialized from FastText cc.ckb.300.bin and fine-tuned during training (vocabulary size 10,000).

The CNN architecture, illustrated in Fig. 1, comprises an embedding layer (vocabulary 10,000; embedding dimension 300; FastText-initialized; trainable), followed by a 1D convolutional layer with 128 filters of kernel size 5, ReLU activation, and L2 regularization (0.001). A dropout layer (rate 0.5) is applied to the convolutional feature map, followed by a global max-pooling layer that reduces the (batch, 146, 128) tensor to a 128-dimensional sentence vector. A dense layer (64 units, ReLU, L2 = 0.001), a second dropout layer (rate 0.5), and a final dense softmax layer (3 output units) complete the network. This 1D-conv-over-embeddings design is the standard adaptation of CNNs to text classification.

The BiLSTM architecture, illustrated in Fig. 2, comprises an identical embedding layer, followed by a single bidirectional LSTM layer with 128 hidden units per direction (256 total), return-sequences enabled, and L2 regularization (0.001). A dropout layer (rate 0.5) is applied to the recurrent output sequence, followed by a global max-pooling layer

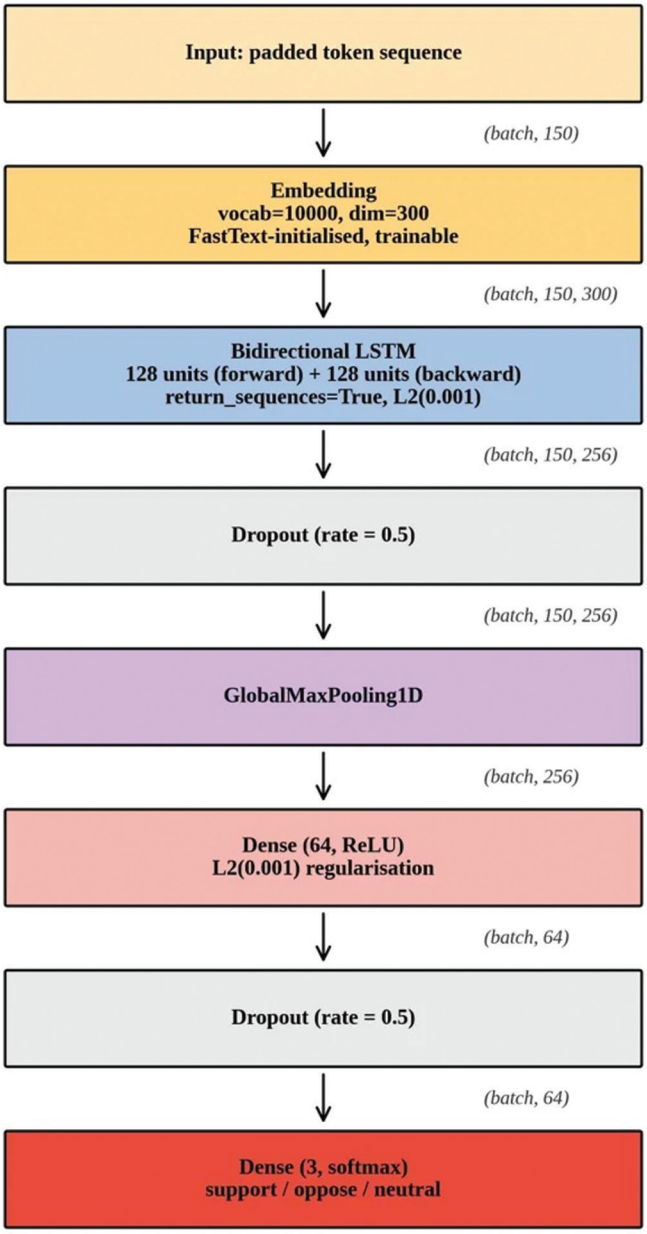

Fig. 2. Architecture of the bidirectional long short-term memory model adapted to Sorani Kurdish text classification. Annotations on the right indicate the tensor shape after each layer.

over the time dimension to yield a 256-dimensional sentence vector. A dense layer (64 units, ReLU, L2 = 0.001), a second dropout layer (rate 0.5), and a final dense softmax layer (3 output units) complete the network. Both networks are optimized with Adam (learning rate 1×10^{-3}), focal loss (γ = 2.0, α = 0.25), batch size 32, up to 20 epochs, and early stopping on validation loss (patience = 3 epochs, restoring best weights). Complete hyperparameter values are summarized in Table III.

Transformers baselines

The experimental design included modern transformer-based models; specifically, two encoders were implemented: (i) KuBERT (Abas, Veisi and Ali, 2025), a KuBERT model with approximately 81.9 million parameters and (ii) XLM-RoBERTa-base (Conneau *et al.*, 2020), a multilingual encoder with approximately 270 million parameters pretrained on 100 languages, including Kurdish. Both encoders were used in a frozen-encoder, linear-probe configuration, which is a standard fair-comparison protocol when downstream training data are limited. Specifically, each article title was tokenized with the model's native tokenizer (with truncation/padding to 128 subword tokens), passed through the frozen encoder, and the final hidden states were mean-pooled across all non-padding positions to produce a single fixed-length embedding (768 dimensions for both KuBERT and XLM-R-base). These embeddings were then fed into a logistic-regression head with class_weight = 'balanced' and L2 regularization (C = 1.0), trained per fold or per split using the same stratified evaluation protocols as the other models. Importantly, KLPT-based stemming was deliberately skipped for the transformer inputs to preserve the subword-tokenization advantages of the encoders' native tokenizers. This frozen-plus-linear-probe approach yields a controlled comparison: Any difference in performance between the transformer baselines and the classical or DL models reflects the quality of the pretrained representations rather than confounding fine-tuning hyperparameters.

IV. Results and Discussion

A. Evaluation Methods and Metrics

The stance detection framework was assessed under four evaluation protocols: Two stratified train–test splits (80/20 and 70/30) and two stratified k-fold cross-validation schemes (5-fold and 10-fold). Stratification preserved the original class proportions of support, oppose, and neutral in each split, mitigating bias due to class imbalance. All experiments used identical preprocessing and feature-extraction settings across model families to keep comparisons consistent. For classical models, hyperparameters were tuned via grid search inside each training partition; for DL models, early stopping with a held-out validation subset drawn from the training partition was used in lieu of grid search. To assess overall accuracy and per-class balance, the following metrics were computed: Accuracy, weighted precision, weighted recall, weighted F1, macro precision, macro recall, macro F1, and micro F1. Macro metrics treat each class equally, while weighted metrics account for class-frequency imbalance. All metrics use the multiclass functions of scikit-learn; scores are averaged over five random seeds, and standard deviations are reported alongside the means.

B. Experimental Results

The SVM model achieved the best accuracy (74%) and weighted F1 (74%) under the 80/20 stratified train–test split. The CNN model achieved slightly higher performance than its DL counterpart, the BiLSTM, suggesting that the CNN's parallel kernels capture localized semantic cues from short Kurdish news titles more effectively than the BiLSTM's

sequential gating. The two frozen transformer baselines achieved noticeably lower accuracies – KuBERT 60% and XLM-R 59% – despite their much larger parameter counts. This counterintuitive ordering is attributable to two factors examined in detail in Section IV.F: (i) The dataset is small (~2,200 titles), which is insufficient to leverage transformer capacity and (ii) the encoders are used in a frozen, linear-probe configuration rather than full fine-tuning, which limits their adaptation to the Kurdish stance task. Table IV summarizes the 80/20 results.

With smaller training data, the 70/30 split produced a slight performance decrease across all models. Even so, SVM (accuracy = 71%, weighted F1 = 72%) again led the field, suggesting strong robustness with limited data. LR achieved nearly identical performance, confirming the value of linear classifiers for stance detection in morphologically rich languages. Table V summarizes these outcomes.

Under 5-fold cross-validation, SVM again produced the strongest performance (accuracy = 72%, weighted F1 = 72%) with high robustness and balanced predictions across classes. Linear models continued to dominate, with LR closely behind (accuracy = 71%, weighted F1 = 71%). Most other models performed in a moderate range (CNN and RF at 70%, XGBoost at 69%, BiLSTM at 67%); the transformer baselines achieved 58–60%. TABLE VI summarizes these results.

Under 10-fold cross-validation, SVM again produced the best results (accuracy = 72%, weighted F1 = 72%), with high cross-fold stability. LR and CNN followed with accuracy in the 70–71% range. BiLSTM, RF, and XGBoost achieved accuracy of 68–70%; the transformer baselines remained at 58–60%. Table VII summarizes these results.

Across all four protocols, SVM achieved the highest performance; the DL models (CNN > BiLSTM) sit in a middle band, and the frozen transformer baselines are clearly weakest. The relative ordering is preserved across protocols, indicating that the conclusions are not artifacts of a single split or fold count.

C. Effect of Class Imbalance Handling

To address the request to demonstrate the effect of class-imbalance handling, each of the six classical and DL models was evaluated under three configurations: (i) No weighting, (ii) the model-appropriate weighting scheme used in the main experiments (class weighting for SVM, LR, RF; sample weighting for XGBoost; focal loss combined with class weighting for CNN, BiLSTM), and (iii) Synthetic Minority Oversampling Technique (SMOTE) applied to the training features (TF–IDF vectors for classical models; token-embedding outputs for DL models). All experiments used 10-fold stratified cross-validation with five random seeds. TABLE VIII reports the comparison.

Three patterns emerge from Table VIII. First, the effect of imbalance handling depends strongly on the model family. RF is highly sensitive: Without weighting, its macro F1 collapses to 0.51, and its neutral-class F1 to 0.32, indicating near-complete failure to recover the minority class; class weighting and SMOTE both restore RF's performance to macro F1 of 0.64 and 0.65, respectively. Second, the linear classifiers (SVM, LR) and XGBoost are comparatively

TABLE IV
Performance of ML, DL, and Transformer Models using 80/20 Stratified Train–Test Split

Algorithms\metrics	Accuracy % ±SD	Weighted precision %±SD	Weighted recall %±SD	Weighted F1%±SD	Macro precision %±SD	Macro recall %±SD	Macro F1%±SD	Micro F1%±SD
BiLSTM	67±0.01	69±0.01	67±0.01	68±0.01	61±0.01	63±0.03	62±0.02	67±0.01
CNN	69±0.02	70±0.02	69±0.02	69±0.02	63±0.02	64±0.03	63±0.02	69±0.02
KuBERT	60±0.02	62±0.02	60±0.02	61±0.02	54±0.02	56±0.02	55±0.02	60±0.02
Logistic regression	72±0.01	73±0.01	72±0.01	72±0.01	67±0.02	70±0.02	68±0.02	72±0.01
RF	71±0.02	71±0.03	71±0.02	70±0.02	**70±0.03**	63±0.03	65±0.03	71±0.02
SVM	**74±0.02**	**74±0.02**	**74±0.02**	**74±0.02**	69±0.02	**71±0.03**	**70±0.03**	**74±0.02**
XGBoost	70±0.01	70±0.01	70±0.01	70±0.01	67±0.01	67±0.02	67±0.02	70±0.01
XLM-R	59±0.01	64±0.01	59±0.01	60±0.01	54±0.01	58±0.02	54±0.01	59±0.01

SD: Standard deviation, BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, ML: machine learning, DL: Deep Learning, SVM: Support vector machine. Note: Bold values indicate the highest performance achieved across all models for each respective metric.

TABLE V
Performance of ML, DL, and Transformer Models using 70/30 Stratified Train–Test Split

Algorithms\Metrics	accuracy %±SD	Weighted precision %±SD	Weighted recall %±SD	Weighted F1%±SD	Macro Precision %±SD	Macro Recall %±SD	Macro F1%±SD	Micro F1%±SD
BiLSTM	64±0.04	65±0.03	64±0.04	64±0.04	58±0.04	59±0.03	58±0.04	64±0.04
CNN	68±0.01	68±0.01	68±0.01	68±0.01	62±0.02	63±0.01	63±0.01	68±0.01
KuBERT	60±0.00	61±0.01	60±0.00	60±0.01	53±0.00	56±0.01	54±0.00	60±0.00
logistic regression	**71±0.01**	**72±0.01**	**71±0.01**	71±0.01	65±0.01	**68±0.01**	66±0.01	**71±0.01**
RF	**71±0.01**	70±0.01	**71±0.01**	70±0.01	**69±0.02**	62±0.01	64±0.01	**71±0.01**
SVM	**71±0.01**	**72±0.01**	**71±0.01**	**72±0.01**	67±0.02	**68±0.02**	**67±0.02**	**71±0.01**
XGBoost	70±0.02	70±0.02	70±0.02	70±0.02	65±0.02	66±0.02	65±0.02	70±0.02
XLM-R	57±0.01	62±0.01	57±0.01	59±0.01	52±0.01	57±0.01	53±0.01	57±0.01

SD: Standard deviation, BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, ML: machine learning, DL: Deep Learning, SVM: Support vector machine. Note: Bold values indicate the highest performance achieved across all models for each respective metric.

 http://dx.doi.org/10.14500/aro.12814

TABLE VI
PERFORMANCE OF ML, DL, AND TRANSFORMER MODELS USING 5-FOLD CROSS-VALIDATION

Algorithms\Metrics	Accuracy %±SD	Weighted precision %±SD	Weighted recall %±SD	Weighted F1%±SD	Macro precision %±SD	Macro recall %±SD	Macro F1%±SD	Micro F1%±SD
BiLSTM	67±0.01	68±0.01	67±0.01	68±0.01	61±0.01	63±0.01	62±0.01	67±0.01
CNN	70±0.00	70±0.00	70±0.00	70±0.00	65±0.01	65±0.01	65±0.01	70±0.00
KuBERT	60±0.00	62±0.00	60±0.00	61±0.00	54±0.01	56±0.01	55±0.01	60±0.00
logistic regression	71±0.01	71±0.00	71±0.00	71±0.00	66±0.00	68±0.00	67±0.00	71±0.01
RF	70±0.00	70±0.00	70±0.00	69±0.00	**68±0.00**	62±0.00	64±0.00	70±0.00
SVM	**72±0.01**	**72±0.00**	**72±0.00**	**72±0.00**	67±0.00	**69±0.00**	**68±0.00**	**72±0.01**
XGBoost	69±0.01	69±0.00	69±0.00	69±0.00	65±0.00	65±0.00	65±0.00	69±0.01
XLM-R	58±0.01	63±0.01	58±0.01	59±0.01	53±0.01	58±0.01	54±0.01	58±0.01

SD: Standard deviation, BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, ML: machine learning, DL: Deep Learning, SVM: Support vector machine. Note: Bold values indicate the highest performance achieved across all models for each respective metric.

TABLE VII
PERFORMANCE OF ML, DL, AND TRANSFORMER MODELS USING 10-FOLD CROSS-VALIDATION

Algorithms\Metrics	Accuracy %±SD	Weighted precision %±SD	Weighted recall %±SD	Weighted F1%±SD	Macro precision %±SD	Macro recall %±SD	Macro F1%±SD	Micro F1%±SD
BiLSTM	68±0.01	69±0.01	68±0.01	68±0.01	62±0.01	64±0.01	63±0.01	68±0.01
CNN	70±0.01	70±0.01	70±0.01	70±0.01	65±0.01	66±0.01	65±0.01	70±0.01
KuBERT	60±0.01	62±0.01	60±0.01	61±0.01	54±0.00	57±0.00	55±0.00	60±0.01
Logistic regression	70±0.01	71±0.00	70±0.00	71±0.00	66±0.00	68±0.00	67±0.00	70±0.01
RF	70±0.00	70±0.00	70±0.00	69±0.00	**68±0.00**	62±0.00	64±0.00	70±0.00
SVM	**72±0.00**	**73±0.00**	**72±0.00**	**72±0.00**	**68±0.00**	**70±0.00**	**68±0.00**	**72±0.00**
XGBoost	70±0.00	70±0.00	70±0.00	70±0.00	66±0.00	66±0.00	65±0.00	70±0.00
XLM-R	58±0.00	63±0.00	58±0.00	60±0.00	53±0.01	59±0.01	54±0.01	58±0.00

SD: Standard deviation, BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, ML: machine learning, DL: Deep learning, SVM: Support vector machine. Note: Bold values indicate the highest performance achieved across all models for each respective metric.

TABLE VIII
EFFECT OF CLASS-IMBALANCE HANDLING ON MODEL PERFORMANCE UNDER 10-FOLD CROSS-VALIDATION

Model	Configuration	Accuracy %±SD	Weighted F1%±SD	Macro F1%±SD	Neutral F1%±SD
BiLSTM	No weighting	**71±0.00**	**71±0.00**	**65±0.01**	**54±0.02**
	Focal+Class weighting	68±0.01	68±0.01	63±0.01	51±0.01
	SMOTE	47±0.02	49±0.02	45±0.02	32±0.02
CNN	No weighting	**71±0.00**	**71±0.00**	**65±0.01**	53±0.01
	Focal+Class weighting	70±0.01	70±0.01	**65±0.01**	**54±0.01**
	SMOTE	51±0.02	53±0.02	48±0.01	34±0.02
Logistic regression	No weighting	**73±0.00**	**72±0.00**	65±0.00	53±0.01
	Class weighting	70±0.01	71±0.00	**67±0.00**	59±0.01
	SMOTE	71±0.00	71±0.00	**67±0.00**	**60±0.01**
RF	No weighting	67±0.00	61±0.00	51±0.00	32±0.02
	Class weighting	70±0.00	69±0.00	64±0.00	57±0.01
	SMOTE	**71±0.00**	**70±0.00**	**65±0.00**	**59±0.01**
SVM	No weighting	**74±0.00**	**73±0.00**	**68±0.00**	60±0.01
	Class weighting	72±0.00	72±0.00	**68±0.00**	**61±0.01**
	SMOTE	72±0.00	72±0.00	**68±0.00**	**61±0.01**
XGBoost	No weighting	**72±0.01**	**71±0.00**	66±0.00	57±0.01
	Sample weighting	70±0.00	70±0.00	65±0.00	58±0.01
	SMOTE	**72±0.00**	**71±0.00**	**67±0.00**	**60±0.01**

SD: Standard deviation, BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, SVM: Support vector machine, SMOTE: Synthetic minority oversampling technique, XGBoost: Extreme gradient boosting. Note: Bold values indicate the highest performance achieved across all models for each respective metric.

robust: Their accuracy varies by less than three percentage points across configurations, although class weighting and SMOTE both yield consistent gains in neutral-class F1. Third, SMOTE is highly destructive for the DL models, reducing accuracy by 21% points for BiLSTM and 19% points for CNN. This finding is consistent with prior reports that SMOTE applied at the embedding-feature level for sequence models produces interpolated samples lacking valid token-sequence semantics, degrading the representations the network has learned. Overall, the model-appropriate weighting schemes adopted in the main experiments are well justified: they recover minority-class performance for the models that need it most (RF in particular) while avoiding the destructive effects of SMOTE on the DL models.

D. Statistical Significance of Model Differences

To assess whether the performance differences observed across models are statistically meaningful, pairwise Wilcoxon signed-rank tests were performed on the per-seed mean macro F1 scores from 10-fold cross-validation. Nine comparisons were selected to cover the principal architectural contrasts of the study: SVM (the strongest baseline) against each of the other seven models, CNN against BiLSTM, and KuBERT against XLM-R. Results are reported in Table IX.

Two observations follow from Table IX. First, the Wilcoxon p-values are at or near the structural floor of the test for n = 5 paired samples (p = 0.0625 in eight of nine comparisons), and none reaches the Bonferroni-corrected threshold $\alpha = 0.05/9 \approx 0.0056$. This is a known limitation of the Wilcoxon test under small sample sizes and does not indicate statistical equivalence between the compared models. Second, the corresponding Cohen's d values are uniformly large or huge: Seven of the nine comparisons exceed d = 2.0, and the SVM-versus-transformer comparisons reach $d \approx 19$, indicating that the across-seed variation is small relative to the between-model performance gap. Taken together, the effect-size evidence supports the conclusion that the model ordering reported in Tables IV-VII reflects substantive differences between models rather than seed-level variation. Future work using a larger seed budget (e.g., 20 or more seeds) could provide formal Wilcoxon significance under standard thresholds.

E. Confusion Matrix Analysis

Beyond aggregate metrics, the confusion matrix reveals classification performance per stance class and exposes typical points of confusion between classes. To enable side-by-side comparison across all eight models, Fig. 3 reports row-normalized confusion matrices computed from the 10-fold cross-validation predictions averaged across the five random seeds. Row-normalization expresses each cell as the recall of the corresponding (true class → predicted class) transition, which removes the visual bias introduced by class imbalance and allows direct cross-model comparison.

Three patterns emerge from Fig. 3. First, the classical models (SVM, LR, XGBoost) and the DL models (CNN, BiLSTM) all exhibit strong diagonal dominance for the majority support class (recall 0.74–0.77), with SVM achieving the most balanced behavior across all three classes (support 0.77, oppose 0.65, neutral 0.68). RF achieves the highest support recall (0.84) but at the cost of the lowest oppose recall (0.53) among the classical models, indicating a bias toward the majority class. Second, the most informative discrimination errors lie between the support and oppose classes for all models (off-diagonal mass typically 0.16–0.43), reflecting the difficulty of separating closely related opinion expressions in Kurdish news headlines. Third, the transformer baselines display markedly different per-class behavior: KuBERT shows uniform confusion across all class pairs (support recall 0.64, oppose recall 0.57, and neutral recall 0.49), while XLM-R achieves the highest neutral recall in the study (0.62) but pays a substantial cost in support and oppose recall (0.59 and 0.55, respectively). These transformer-specific patterns are consistent with the calibration analysis discussed in the section below.

V. Discussion

Under the chosen experimental setup, classical ML models – specifically SVM and LR – achieved the strongest performance. SVM obtained the highest weighted F1 of 74% on the 80/20 split and 72% under 10-fold cross-validation, with LR producing near-identical results; however, these results do not imply that linear models are intrinsically superior in general. Among DL baselines, CNN outperformed BiLSTM because parallel kernels efficiently capture local n-gram dependencies in dense Kurdish news titles, while BiLSTM gains little from short sequences. RF and XGBoost were competitive but trailed slightly due to underweighting rare tokens in sparse spaces.

The transformer baselines exhibited distinct trade-offs. KuBERT achieved slightly higher hard-classification metrics (accuracy 0.60 vs. 0.58 and neutral F1 0.43 vs. 0.40 under 10-fold cross-validation) despite being 3.3× smaller than XLM-R (81.9M vs. 270M parameters), showing the benefit of Sorani-specific pretraining. Conversely, XLM-R produced better-calibrated probability estimates (log loss 0.89 vs. 1.23) and recovered more neutral instances (recall 0.62 vs. 0.49) due to broader multilingual pretraining. Both transformers underperformed relative to the TF–IDF SVM by 12–14 percentage points of weighted F1, which is explained by the small dataset size (~2,200 titles) and the frozen, linear-probe configuration.

TABLE IX
Pairwise Wilcoxon Signed-Rank Tests on Per-Seed Macro F1 Scores Under 10-Fold Cross-Validation

Model A	Model B	Mean A	Mean B	Δ (A−B)	Cohen's d	Wilcoxon p-value
SVM	Logistic regression	0.68	0.67	0.0147	1.55	0.0625
SVM	RF	0.68	0.65	0.0308	**10.70**	0.0625
SVM	XGBoost	0.68	0.65	0.0269	**2.95**	0.0625
SVM	CNN	0.68	0.65	0.0354	**8.33**	0.0625
SVM	BiLSTM	0.68	0.63	0.0514	**16.70**	0.0625
SVM	KuBERT	0.68	0.55	0.1288	**19.20**	0.0625
SVM	XLM-R	0.68	0.54	0.1416	**19.89**	0.0625
CNN	BiLSTM	0.65	0.63	0.0160	**2.50**	0.0625
KuBERT	XLM-R	0.55	0.54	0.0128	1.15	0.1250

BiLSTM: Bidirectional long short-term memory, CNN: Convolutional neural network, RF: Random Forest, SVM: Support vector machine. Note: Bold values indicate the highest performance achieved across all models for each respective metric.

http://dx.doi.org/10.14500/aro.12814

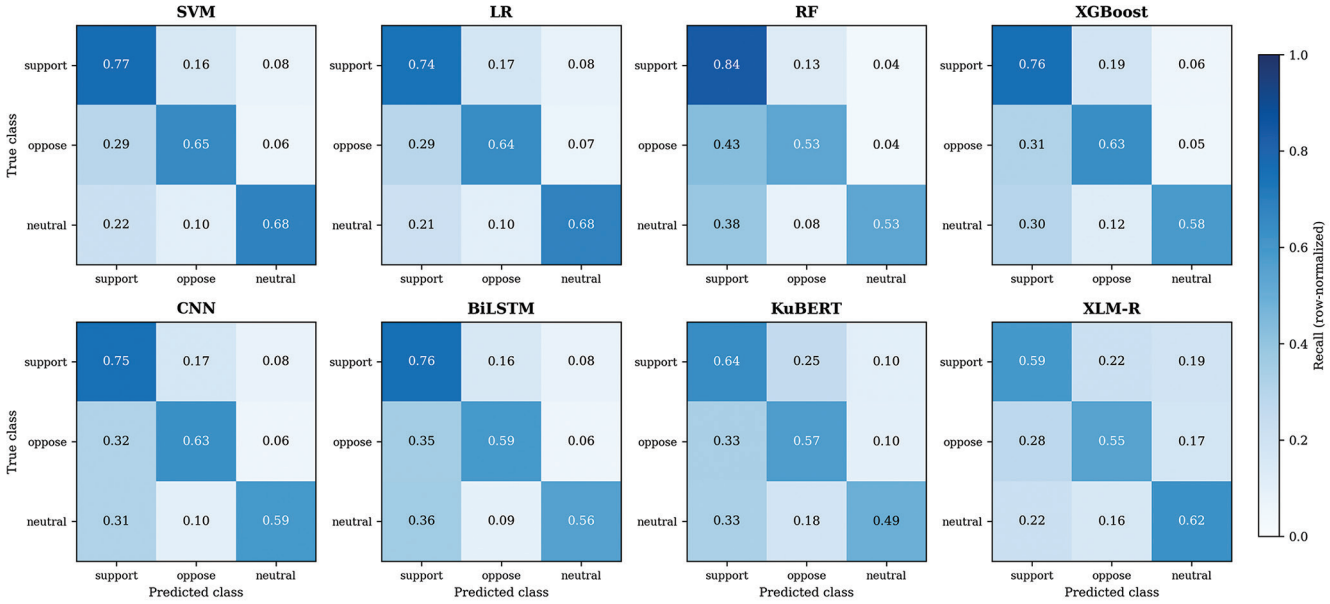

Fig. 3. Row-normalized confusion matrices of all eight models under 10-fold cross-validation, averaged across five random seeds. Top row: SVM, LR, RF, XGBoost. Bottom row: CNN, BiLSTM, KuBERT, XLM-R. Diagonal cells show per-class recall; off-diagonal cells show typical misclassification patterns. SVM: Support vector machine, RF: Random forest, LR: Logistic regression, XGBoost: Extreme gradient boosting.

Confusion-matrix analysis showed that most errors occur between support and oppose classes due to overlapping polarity expressions. Neutral articles were the hardest to identify because they lack direct sentiment cues and adopt a factual tone, reflecting implicit stance communication in Sorani Kurdish news writing. Stratified sampling, regularization, and early stopping successfully controlled overfitting. Finally, the experiments indicate models capture genuine stance signals rather than pipeline artifacts because: (i) All four classical models converge within a four-percentage-point band (70–74%) under 10-fold cross-validation despite heterogeneous decision functions and (ii) the pretrained transformer baselines recover comparable per-class confusion patterns (Fig. 3) without prior exposure to the annotation pipeline.

A. Limitations

Several key limitations warrant explicit acknowledgment. First, performing stance classification on article titles rather than full body text excludes distributed rhetorical and argumentative cues. Second, the modest scale of the Bochun dataset (~2,200 articles) limits the representational capacity of transformer models compared to high-resource benchmarks. Third, generalizability is restricted since the data cover only two domains (economy and politics) and a single dialect (Sorani). Methodologically, conducting the pairwise statistical significance analysis across five random seeds establishes a structural floor for Wilcoxon p-values ($p \geq 0.0625$ for $n = 5$), which accounts for the non-significant p-values reported in Table IX despite large Cohen's d effect sizes. Finally, because full fine-tuning was unfeasible at this dataset size, evaluation is limited to a frozen-encoder, linear-probe configuration.

VI. Conclusion and Future Work

This study presented the first systematic evaluation of stance detection for Sorani Kurdish using the Bochun dataset. Eight classifiers – spanning classical ML (SVM, LR, RF, XGBoost), DL (CNN, BiLSTM), and transformers (KuBERT, XLM-RoBERTa) – were benchmarked under four evaluation protocols and five random seeds, alongside a class-imbalance ablation and a pairwise statistical-significance analysis. Under the chosen setup, SVM on TF–IDF features achieved the strongest performance, with an accuracy and weighted F1 of 74% on the 80/20 stratified split and 72% under 10-fold cross-validation, outperforming the DL and frozen-transformer baselines. The ablation demonstrated that model-appropriate weighting strategies (class weighting for SVM, LR, RF; sample weighting for XGBoost; focal loss with class weighting for CNN, BiLSTM) reliably recovered minority-class performance, whereas SMOTE proved destructive for sequence model embeddings. Pairwise Wilcoxon tests and Cohen's d effect sizes confirmed these differences reflect substantive performance gaps rather than seed-level variation, establishing a robust reference benchmark for Sorani Kurdish.

Several future directions emerge from this work. First, extending the unit of analysis from titles to full body text would allow models to exploit distributed rhetorical cues. Second, the statistical analysis would benefit from a larger seed budget (≥20 seeds) to achieve formal significance. Third, expanding the dataset to include new thematic domains (sports, culture, and technology) and additional dialects (Kurmanji, Hawrami) would test external validity. Finally, replacing the frozen-encoder setup with full fine-tuning, parameter-efficient fine-tuning, or domain-adaptive pretraining represents the most plausible route to closing the transformer-versus-classical performance gap. Beyond methodology, this

work has practical applications for automated Kurdish-media monitoring, partisan-content detection, and misinformation analysis in low-resource settings.

References

Abas, A., Veisi, H., and Ali, H.M., 2025. *KurdSTS: The Kurdish Semantic Textual Similarity*. [Preprint].

Ahmadi, S., 2020. KLPT - Kurdish Language Processing Toolkit. Association for Computational Linguistics, Stroudsburg, pp.72-84.

Alhindi, T., Alabdulkarim, A., Alshehri, A., Abdul-Mageed, M., and Nakov, P., 2021. AraStance: A Multi-Country and Multi-Domain Dataset of Arabic Stance Detection for Fact Checking. In: *NLP4IF 2021 - NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, Proceedings of the 4th Workshop*, pp.57-65.

Aljohani, N.R., Fayoumi, A., and Hassan, S.U., 2023. A novel focal-loss and class-weight-aware convolutional neural network for the classification of in-text citations. *Journal of Information Science*, 49(1), pp.79-92.

Alturayeif, N., Luqman, H., and Ahmed, M., 2022. MAWQIF: A Multi-label Arabic Dataset for Target-specific Stance Detection. In: *WANLP 2022 - 7th Arabic Natural Language Processing - Proceedings of the Workshop*, pp.174-184.

Aslam, S., Arshad, S., Shabir, Z., Sohail, M., Ishaq, R., Hameed, H., Javed, S., Ahmed, A.H., and Ahmed, N.H., 2025. Global voices, local frames: Cross-lingual corpus analysis of stance and discourse in social media and news. Scholars Journal of Arts, Humanities and Social Sciences, 9493(9), pp.320-334.

Azad, R., Ahmed, M.S., and Saeed, S.A.B., 2025. KurdABSA : Kurdish aspect-based sentiment analysis dataset curation using few-shot learning. *Data in Brief*, 62, p.112012.

Aziz, K.O., Teimoor, R.A., Tofiq, T.A., and Abdulla, S., 2024. Kurdish sorani dialect morphology generation using a concatenative strategy. *UHD Journal of Science and Technology*, 8(1), pp.13-19.

Badawi, S., 2023. Data augmentation for sorani kurdish news head- line classification using back-translation and deep learning model. *Kurdistan Journal of Applied Research*, 8(1), pp.27-37.

Bharathi, A., and Zubiaga, A., 2025. Zero-shot cross-lingual stance detection via adversarial language adaptation. *PeerJ Computer Science*, 11, p.e2955

Cignarella, A.T., Lai, M., Bosco, C., Patti, V., and Rosso, P., 2020. SardiStance @ EVALITA2020: Overview of the task on stance detection in Italian tweets. *CEUR Workshop Proceedings*, 2765, pp.1-10.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán., F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V., 2020. Unsupervised Cross-Lingual Representation Learning at Scale. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp.8440-8451.

Hercig, T., Krejzl, P., Hourová, B., Steinberger, J., and Lenc, L., 2017. Detecting stance in Czech news commentaries.*CEUR Workshop Proceedings*, 1885, pp.176-180.

Küçük, D., 2017. Stance detection in Turkish tweets. *CEUR Workshop Proceedings*, 1914, pp.3-6.

Mets, M., Karjus, A., Ibrus, I., and Schich, M., 2024. Automated stance detection in complex topics and small languages: The challenging case of immigration in polarizing news media. *PLoS ONE*, 19(4), p.e0302380

Mohammad, S.M., Kiritchenko, S., Sobhani, P., Zhu, X., and Cherry, C., 2016. SemEval-2016 task 6: Detecting Stance in Tweets. In: *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, pp.31-41.

Mohammadi, S.M., Farzi, S., Alavi, S.M., and Joonaghany, G.H., 2024. Stance detection on social media, case study: Persian sentences using deep learning architecture. *Scientia Iranica*, 31(10), pp.764-773.

Rostam, P.S., and Nabi, R.M., 2025. Bochun: Automatically annotated stance detection dataset for Sorani Kurdish language. *Data in Brief*, 61, p.111839.

Saeed, A.M., 2024. An automated new approach in fast text classification: A case study for Kurdish text. *Science Journal of University of Zakho*, 12(3), pp.1-4.

Sane, S.R., Tripathi, S., Sane, K.R., and Mamidi, R., 2021. Stance Detection in Code-Mixed Hindi-English Social Media Data Using Multi-Task Learning. In: *WASSA@NAACL-HLT 2019 - 10th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Proceedings*, pp 1-5.

Vamvas, J., and Sennrich, R., 2020. X-Stance: A Multilingual Multi-Target Dataset for Stance Detection. In: *CEUR Workshop Proceedings*, p.2624.

Xie, X., Xie, M., Moshayedi, A.J., and Skandari, M.H., 2022. A hybrid improved neural networks algorithm based on L2 and dropout regularization. In: *Mathematical Problems in Engineering*. Wiley, Hoboken.

Zhang, W., Yoshida, T., and Tang, X., 2011. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, 38(3), pp.2758-2765.

Zotova, E., Agerri, R., Nuñez, M., and Rigau, G., 2020. Multilingual Stance Detection: The Catalonia Independence Corpus. In: *LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings*, pp.1368-1375.