DRES: Fake news detection by dynamic representation and ensemble selection

Faramarz Farhangian[1], Leandro A. Ensina[2], George D. C. Cavalcanti[3], Rafael M. O. Cruz[1]

[1]École de Technologie Supérieure (ÉTS-Montréal), Canada
[2]Universidade Tecnológica Federal do Paraná (UTFPR), Brazil
[3]Universidade Federal de Pernambuco (UFPE), Brazil

faramarz.farhangian.1@ens.etsmtl.ca, leandroa@utfpr.edu.br, gdcc@cin.ufpe.br, rafael.menelau-cruz@etsmtl.ca

Abstract

The rapid spread of information via social media has made text-based fake news detection critically important due to its societal impact. This paper presents a novel detection method called Dynamic Representation and Ensemble Selection (DRES) for identifying fake news based solely on text. DRES leverages instance hardness measures to estimate the classification difficulty for each news article across multiple textual feature representations. By dynamically selecting the textual representation and the most competent ensemble of classifiers for each instance, DRES significantly enhances prediction accuracy. Extensive experiments show that DRES achieves notable improvements over state-of-the-art methods, confirming the effectiveness of representation selection based on instance hardness and dynamic ensemble selection in boosting performance. Codes and data are available at: https://github.com/FFarhangian/FakeNewsDetection_DRES.

1 Introduction

Detecting fake news is an increasingly important task in today's world as false news spreads significantly faster and deeper than true news (Vosoughi et al., 2018). This problem is further exacerbated by generative AI, which amplifies misinformation by creating highly persuasive but fabricated content (Loth et al., 2024). Traditional text-based models frequently struggle with context sensitivity and generalization, especially when processing ambiguous text or domain-shifted inputs (Wang, 2017; Reddy et al., 2020). Addressing these issues is essential to support the credibility of information in online platforms and public communication.

While large language models (LLMs) such as Mistral (Jiang et al., 2023) have improved performance in many text classification tasks, relying on a single feature representation may be insufficient for fake news detection. Different representations

Figure 1: Instance hardness heat map for a few instances taken from the Liar dataset.

capture complementary aspects of the input, such as surface-level statistics or contextual semantics. Recent work (Farhangian et al., 2024) shows that using multiple representations, even with the same classifier, can reduce misclassification errors by exploiting this diversity. However, attempts to naively combine all representations (e.g., (Pereira et al., 2025)) often yield suboptimal results due to increased noise failing to account for input-specific representation effectiveness.

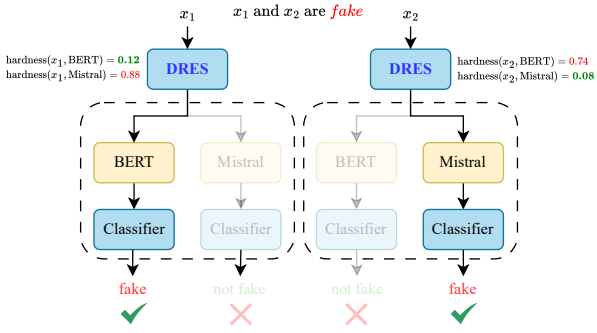


Figure 2: Overview of DRES that dynamically selects the most suitable text representation based on the test-time instance hardness.

In this paper, we argue that robust detection requires dynamic selection of both representations and classifiers based on input characteristics. We

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 20023–20041
November 4-9, 2025

formalize this via instance hardness (Smith et al., 2014), which measures how likely a sample is to be misclassified by any learning algorithm under a given representation. Figure 1 illustrates this, showing hardness scores for Liar dataset (Wang, 2017) instances across three representations. A news item may have low hardness in ELMO's space but high hardness in Mistral's, with the reverse holding for other inputs. Thus, naively combining all representations or classifiers may hurt performance.

To bridge this gap, we propose a two-stage framework called **DRES** (Dynamic Representation and Ensemble Selection). An overview is shown in Figure 2. First, given an input, DRES estimates instance hardness across multiple text representations using a test-time variant of the k-Disagreeing Neighbors (kDN) metric (Smith et al., 2014). Based on this estimate, the representation with the lowest hardness is selected.

However, even within the chosen representation, classifiers trained on that space can still exhibit distinct local error patterns, especially for harder instances. To address this, DRES applies dynamic ensemble selection (Cruz et al., 2018) as a second stage, selecting only the most competent classifiers in the neighborhood of the query rather than averaging over the entire pool. This ensures that predictions are adapted to the local region of the input, improving robustness on difficult cases. The final prediction is obtained by majority voting over this selected subset. While prior work (Cook et al., 2025) used hardness metrics to guide training or data sampling, we extend their use to test time, showing that instance hardness can also support inference-time adaptation.

Our approach introduces a novel synergy between representation selection and classifier specialization. By first selecting the representation with the lowest estimated instance hardness, the framework maps the problem to a space where classifiers are more likely to agree. A second stage then refines this by dynamically selecting the most locally competent classifiers in that space per test instance. To the best of our knowledge, this is the first work to apply dynamic representation and ensemble selection jointly, and to repurpose complexity-based metrics to guide inference-time decisions.

The contributions of this paper are as follows: 1) We propose a dynamic multi-view ensemble framework for fake news detection that adaptively selects both the most effective text representation and the most competent classifiers for each instance. 2) We introduce a novel test-time estimation approach for instance hardness using a supervised metric (kDN) to guide representation selection decisions during inference. 3) We empirically demonstrate that combining diverse text representations with dynamic classifier selection leads to consistent performance improvements across multiple fake news datasets while having plenty of potential for future improvements.

2 Related work

Fake news detection has been approached from diverse perspectives, including image-based analysis (Qi et al., 2019) and social-context methods (i.e., modeling network structure and propagation patterns) with geometric deep learning (Monti et al., 2019). As this work is dedicated to textual content, our review focuses on text-only models and hybrid text-augmented methods.

Text-Only Fake News Detection. The earliest text-only methods relied on shallow lexical and stylometric features such as TF-IDF representations (Patwa et al., 2021), n-grams, punctuation patterns, and readability metrics (Agudelo et al., 2018) fed into classifiers like Logistic Regression, SVM, or Random Forest (Shu et al., 2017). Pérez-Rosas et al. (2018) expanded on these by adding psycholinguistic and rhetorical cues to capture deceptive language, while Potthast et al. (2018) showed that pure stylometric features alone can separate hyperpartisan and fake news from genuine articles.

Later work moved to dense representations. Word-embedding–based models include BiLSTM over GloVe (Sastrawan et al., 2022), and CNNs on Word2Vec (Girgis et al., 2018). These were followed by transformer models like FakeBERT, which pairs BERT embeddings with 1D-CNN filters (Kaliyar et al., 2021), ScrutNet's Bi-LSTM + CNN fusion (Verma et al., 2025), and by attention-driven architectures such as 3HAN, a three-level hierarchical network that attends separately to words, sentences, and headlines for interpretable classification (Singhania et al., 2017). Finally, end-to-end transformer classifiers (e.g., BERT, LLaMA) now serve as strong baselines (Liu and Chen, 2023; Farhangian et al., 2024). These unimodal approaches all use fixed representations or model combinations and cannot adjust to per-instance variations in text complexity, making them vulnerable to domain shifts and adversarial inputs.

Multiple Text Representation Techniques. Some

approaches combine different textual representations of the same input to enrich the feature space. Essa et al. (2023) merge several layers of BERT embeddings and use the result in a LightGBM classifier, while Gautam et al. (2021) integrate XLNet embeddings with LDA topics to capture both contextual and thematic cues. MisRoBÆRTa (Truică and Apostol, 2022) fuses BART and RoBERTa encodings into an ensemble architecture. MVAE (Pereira et al., 2025) aligns heterogeneous features into a shared latent space with Multi-View Auto Encoder.

Nevertheless, these methods fuse all representations uniformly, without accounting for input-specific suitability. This can lead to inefficiencies and misclassification, as shown in (Pereira et al., 2025), where joint representations from MVAE reduce performance on certain instances. Moreover, Peng et al. (2024) further highlight that news samples vary widely in structure, supporting the need for context-aware embedding selection. In contrast, the proposed DRES works by first estimating the hardness of each instance and dynamically selecting the most suitable representation before applying ensemble selection within that space. Thus, the representation and classifiers are adapted to the specific characteristics of each input.

Hybrid text-augmented methods. To harness richer signals, some methods integrate metadata (e.g., publication source, timestamps, user profiles) or fuse multiple data modalities. Social-context models encode user engagement and propagation patterns via graph neural networks (Shu et al., 2019). Multimodal frameworks process text alongside images using attention or contrastive learning—SpotFake aligns text and image embeddings (Singhal et al., 2019), while MCOT applies optimal transport for cross-modal fusion (Shen et al., 2024). Hybrid systems further incorporate temporal or social features into neural classifiers (Ruchansky et al., 2017) to improve classification.

Multimodal approaches that integrate textual and social context features show consistent improvements in fake news detection accuracy (Huang et al., 2019), with late fusion methods combining text, image, and social signals providing further gains (Nguyen et al., 2020). DANES (Truică et al., 2024) combines RNN-based text encoding with RNN-CNN social context encoding, concatenating both into a unified embedding for final classification via a dense layer. Metadata and multimodal inputs are often dataset-specific and may be missing or noisy in real-world settings, risking biased predictions and limiting generalization beyond curated benchmarks.

Dynamic Ensemble Selection. Ensemble methods improve performance by combining multiple base learners to reduce bias and variance compared to any single model (Dietterich, 2000). Static ensembles use fixed strategies such as majority voting (Kittler et al., 1998) or stacked generalization (Wolpert, 1992). Dynamic ensemble selection (DES) instead adapts to each query by (1) defining a Region of Competence (RoC) around the sample (e.g., via clustering or similarity search), (2) measuring each classifier's competence within that RoC, and (3) combining the outputs of the most competent models. DES techniques mainly differ in the heuristics used to estimate classifier competence within the Region of Competence. KNORA-U/E use local accuracy oracles (Ko et al., 2008), META-DES predicts competence via a meta-learner trained on multiple meta-features (Cruz et al., 2015), DES-P selects classifiers that perform locally better than a random classifier (Woloszynski et al., 2012), and recent methods leverage graph neural networks (Souza et al., 2024) or fuzzy neural networks (Davtalab et al., 2024) to model more flexible competence patterns.

The key assumption in DES is that a classifier that succeeds on samples similar to the query will also succeed on the query itself (Woods et al., 1997). By focusing on these local competence estimates, DES handles hard cases, such as samples near decision boundaries, more effectively than static models (Cruz et al., 2018). However, to the best of our knowledge, DES work assumes a single feature space. DRES addresses this gap with a two-stage framework that first picks the best representation per sample based on hardness measures, then applies competence estimation heuristics and classifier selection from DES techniques within the selected representation space.

3 Dynamic Representation and Ensemble Selection (DRES)

DRES tackles the limitations of fixed fusion and static ensembles by adapting both representation and classifier selection to each input. During training, we compute an instance hardness profile for each sample under every text representation. At inference, we introduce a novel test-time instance hardness estimation by relating a new article to

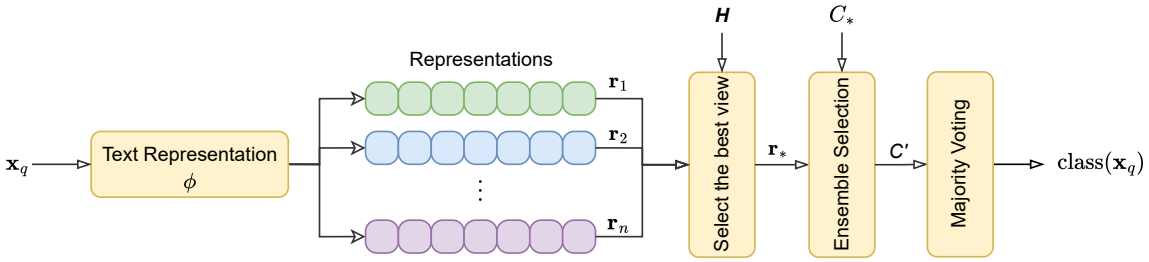


Figure 3: Dynamic Representation and Ensemble Selection (DRES) generalization phase. $\mathbf{x}_q$ is an unknown sample, $\mathbf{r}_j = \phi_j(\mathbf{x}_q)$ is the text representation for $\mathbf{x}_q$, $\mathbf{H}$ is the instance hardness matrix, $\mathbf{r}_*$ is the best text representation associated with the lowest hardness score, C_* is a pool of classifiers trained using the representation $\mathbf{R}_*$, and $C' \subset C_*$ is the most competent subset of classifiers to predict the class of $\mathbf{x}_q$.

its neighbors' training-time hardness to choose the easiest representation, then dynamically select the most competent classifiers in that space to make the final decision.

3.1 Training Phase

In the training phase, we embed the training set $\mathcal{T} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{|\mathcal{T}|}\}$ into n representation matrices $\mathcal{R} = \{\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_n\}$ using text representation algorithms $\phi = \{\phi_1, \phi_2, \ldots, \phi_n\}$. That is, $\mathbf{R}_j = \phi_j(\mathcal{T})$, and each $\mathbf{R}_j \in \mathbb{R}^{|\mathcal{T}| \times d_j}$ has its own dimensionality d_j (e.g., 768 for BERT and 4096 for Mistral). Afterward, two processes occur: instance hardness calculation and classifier training.

Instance Hardness Calculation. The instance hardness (IH) of each sample in $\mathcal{T}$ is calculated to compose $\mathbf{H} \in \mathbb{R}^{|\mathcal{T}| \times n}$. Each element $h_{ij} \in \mathbf{H}$ denotes the hardness of instance $\mathbf{x}_i$ when encoded using the $\mathbf{R}_j$ representation. Among the instance hardness metrics, we selected the kDN metric due to its high correlation with classification errors (Smith et al., 2014; Paiva et al., 2022). Equation 1 shows how h_{ij} is calculated.

$$h_{ij} = \text{kDN}(\mathbf{x}_i, \mathbf{R}_j, k) = \frac{|\{(\mathbf{x}_l, \mathbf{y}_l) \in \mathcal{N}_k(\mathbf{x}_i; \mathbf{R}_j) : \mathbf{y}_l \neq \mathbf{y}_i\}|}{k} \quad (1)$$

where, $\mathcal{N}_k(\mathbf{x}_i; \mathbf{R}_j)$ denotes the set of the k-nearest neighbors of $\mathbf{x}_i$ in the feature space defined by the representation $\mathbf{R}_j$. In other words, kDN counts the number of neighboring instances whose labels differ from that of $\mathbf{x}_i$. Thus, a higher h value indicates that the instance resides in a region of class overlap and is, therefore, difficult to classify correctly.

Classifier Training. For each representation $\mathbf{R}_j \in \mathcal{R}$, we generate a set $C_j = \{c_{j1}, c_{j2}, \ldots, c_{jm}\}$ of m classifiers, each trained with a different learning algorithm. This results in n pool of classifiers $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, where each C_j corresponds to models trained on the same representation $\mathbf{R}_j$. In total, the framework produces $n \times m$ classifiers spanning all combinations of representation and learning algorithms.

3.2 Generalization Phase

Given a new news article as input ($\mathbf{x}_q$), the generalization phase consists of three main steps (Figure 3):

Text Representation. $\mathbf{x}_q$ is transformed into n different representations, denoted as $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$, where $\mathbf{r}_j = \phi_j(\mathbf{x}_q)$. Our proposed DRES framework is agnostic regarding the number and type of text representations used.

Representation Selection. We estimate the difficulty of predicting the sample query $\mathbf{x}_q$ across different feature spaces ($\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n\}$, as shown in Figure 3). The goal is to determine how hard it is to predict $\mathbf{x}_q$ based on each representation and use this information to guide the selection of the optimal feature space. The central hypothesis is that focusing on the classifiers trained using the easiest representation will lead to better prediction outcomes for $\mathbf{x}_q$.

Since kDN is a supervised metric and we lack access to labels during generalization, we propose an unsupervised estimation of the instance hardness score for the new query sample, called *test-time instance hardness*. It works as follows: for each representation $\mathbf{r}_j$, we identify its k-nearest neighbors in its corresponding training data embeddings matrix $\mathbf{R}_j$ using the k-Nearest Neighbors algorithm. We then estimate its test-time instance hardness score $\hat{h}_j$ by averaging the precomputed instance hardness scores $h_{lj} \in \mathbf{H}$ of these neighbors:

$$\hat{h}_j = \frac{1}{k} \sum_{l=1}^{k} h_{lj} \quad (2)$$

where h_{lj} is the precomputed hardness score of the l-th nearest neighbor in the representation $\mathbf{R}_j$ (Eq. 1). This process is shown in the upper part of Figure 4. As an example, for the representation 1, we first select the seven neighbors of $\mathbf{x}_q$ in $\mathcal{T}$ (as shown in the leftmost figure), and their instance values are retrieved from the matrix $\mathbf{H}$ (these values are shown close to each instance). After, the hardness of $\mathbf{x}_q$ is calculated using Eq. 2; so, $\hat{h}_1 = 0.26$. The same procedure is performed for all representations.

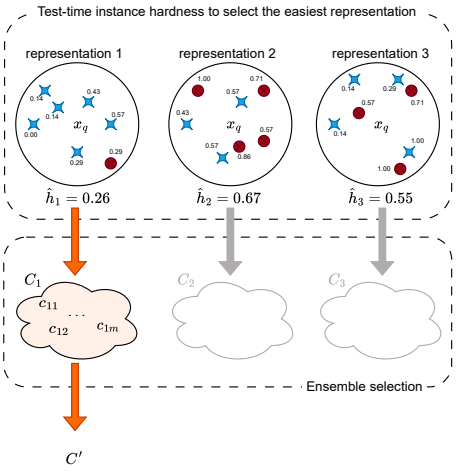


Figure 4: Test-time instance hardness calculation and Ensemble selection. The input text $\mathbf{x}_q$ and its neighbors in three different representation spaces. Instances from two classes (blue and red) are shown associated with their instance hardness values extracted from $\mathbf{H}$. $\hat{h}_j$ represents the instance hardness of $\mathbf{x}_q$ under representation $\mathbf{R}_j$. C_j is the pool of classifiers trained with the same representation and $C' \subset C_1$ is the most competent subset of classifiers to classify $\mathbf{x}_q$.

Collecting these estimated hardness scores produces a vector $\hat{\mathbf{h}} = \{\hat{h}_1, \hat{h}_2, \ldots, \hat{h}_n\}$, where each element corresponds to the test-time instance hardness for a specific text representation. Then, the text representation with the lowest estimated instance hardness score (Eq. 3) is selected. When multiple representations share the same minimal hardness score $\hat{h}_j$, we select the one with the lowest average instance hardness across the entire training set.

$$\mathbf{r}_* = \arg\min_{\mathbf{r}_j} \hat{h}_j. \tag{3}$$

In the example (Figure 4), representation 1 is claimed as the best feature space to classify $\mathbf{x}_q$ since it lies in a region where its neighbors are more prone to be correctly classified given their kDN. So, $\mathbf{r}_*$ is defined as representation 1 ($\mathbf{r}_1$).

Ensemble Selection. After the easiest representation $\mathbf{r}_*$ is chosen for query $\mathbf{x}_q$, the method proceeds to select classifiers from the pool C_* trained on $\mathbf{r}_*$. A dynamic selection mechanism identifies the subset of C_* most competent for $\mathbf{x}_q$, and the final prediction is produced via majority vote.

Thus, as shown in the lower part of Figure 4, $C_1 \subset \mathcal{C}$ is selected since its classifiers were trained on the text representation previously chosen (representation 1). Given that not all classifiers $c \in C_1$ are competent to classify $\mathbf{x}_q$, only a subset of classifiers $C' \subset C_1$ is selected. This subset selection is performed by dynamic ensemble selection methods (Cruz et al., 2018), which have the advantage of addressing the classification task in a customized way since the subset C' depends on the query instance under evaluation. The proposed framework is flexible and can incorporate any dynamic selection algorithms, such as META-DES (Cruz et al., 2015) or KNORA-E (Ko et al., 2008). The predictions of the selected ensemble (C') are then combined using the Majority Vote rule to produce the final classification.

4 Experimental Setting

Datasets. We evaluate DRES on three standard fake news detection benchmarks: (1) Liar (Wang, 2017) (12.8K statement collected from PolitiFact, 6 truthfulness categories), (2) COVID (Patwa et al., 2021) (10.7K fact-checked tweets about the pandemic), and (3) GM (McIntire, 2017) (14.1K news articles from reputable sources like the New York Times and Wall Street Journal). These datasets represent diverse fake news scenarios, from political claims to health misinformation. Detailed information on these datasets can be found in Table 1.

Models. We employ a comprehensive set of $m = 10$ classification algorithms spanning traditional machine learning (Support Vector Machines, Logistic Regression, K-Nearest Neighbors, Naive Bayes, Multi-layer Perceptron, Random Forest, AdaBoost, XGBoost) and deep learning approaches (Convolutional Neural Networks, Bidirectional LSTMs). These are combined with $n = 14$ feature representations, including context-independent models (Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), FastText (Bojanowski et al., 2017)) and context-dependent transformers (BERT (Devlin et al., 2018), DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019), BART (Lewis et al., 2019), ELECTRA

Table 1: Main characteristics of the dataset used in this study.

Dataset	Domain	Media	Fact-checking	Size	No.Class	Class distribution
Liar (Wang, 2017)	Politics	Mainstream media	Editors & journalists	12836	6	(1050, 2511, 2108, 2638, 2466, 2063)
George McIntire (McIntire, 2017)	Business,Technology, etc.	Mainstream media	journalists	11000	2	(3151, 3159)
Covid (Patwa et al., 2021)	Covid-19 & Health	Twitter	Fact-checking websites	10700	2	(5100, 5600)

(Clark et al., 2020), XLNet (Yang et al., 2019), Falcon (Almazrouei et al., 2023), ELMo (Peters et al., 2018), LLaMA3 (Grattafiori et al., 2024), and Mistral (Jiang et al., 2023)). Detailed information on these representations can be found in Appendix A.

We use 5-fold cross-validation with hyperparameter tuning (see Appendices B and C), considering F1 scores to account for class imbalance.

Baselines. We evaluate DRES against three static ensemble baselines representing distinct static selection approaches in order to test our hypothesis that dynamic selection of both representations and classifiers outperforms static ensemble strategies:

- **Group A:** 10 pools of the same classifier type across all feature representations (10 single classifiers $\times$ 14 representations).
- **Group B:** 14 pools of diverse classifiers per text representation (10 classifiers $\times$ 14 single representation)
- **Group C:** Full combination of all 140 classifiers (10 classifiers $\times$ 14 representations).

The classifiers in each group are combined using stacked generalization (Wolpert, 1992) with logistic regression as the meta-classifier, as in (Cruz et al., 2022).

In the DRES method, instance hardness scores were calculated using kNN ($k = 5$) to define the region of competence (RoC), i.e., the local neighborhood of the test instance in the validation dataset. A detailed analysis of different k-values is provided in Appendix D. We adopted three dynamic ensemble selection techniques, KNORA-E, DES-P, and META-DES, which were adapted from the DESlib library (Cruz et al., 2020) to support multiple text representations. These methods were chosen because they rely on neighborhood-based competence estimation, aligning with our test-time instance hardness approach.

5 Results

5.1 Comparison with static ensemble

Table 2 reports DRES's performance against the baseline Groups A, B, and C using macro F1. In this table, only the best results for Groups A and B are reported; the complete results and additional metrics are in Appendices E and F.

Table 2: F1-score per dataset for DRES and baseline models. The absolute best results per dataset are in bold, and the top methods that are statistically equivalent are marked with an asterisk.

	Dataset		
Method	**Liar**	**COVID**	**GM**
MLP (Best Group A)	0.250 (0.002)	0.950 (0.002)	0.950 (0.003)
Mistral (Best Group B)	0.260 (0.002)	0.943 (0.003)	0.951 (0.002)
Group C	0.243 (0.003)	0.941 (0.003)	0.950 (0.002)
DRES (KNORA-E)	0.371 (0.003)	**0.973 (0.002)***	**0.986 (0.002)***
DRES (META-DES)	0.367 (0.002)	0.972 (0.002)*	0.980 (0.005)
DRES (DES-P)	**0.385 (0.003)***	0.972 (0.003)*	0.984 (0.002)*

DRES outperforms static strategies by a clear margin for all datasets, independent of the dynamic selection classifier used, as statistically corroborated by the repeated measures ANOVA with Tukey's post-hoc test with a 95% confidence interval. The tests for all datasets revealed statistically significant differences (p-value < 0.0001), demonstrating the superiority of our method. In Table 2, the methods with the absolute best results are highlighted in bold, and the top methods that are statistically equivalent are marked with an asterisk for easier identification. For the Liar dataset, DRES with DES-P presented the best performance, while for the COVID dataset, the performance differences among DRES methods (KNORA-E, META-DES, and DES-P) were not statistically significant. In turn, DRES with KNORA-E and DES-P presented better results for the GM dataset, with no significant difference between them.

Group B, a homogeneous ensemble that uses only one text representation for all classifiers, fails when the representation is unsuitable for specific instances. Group A, limited to a single classifier learning algorithm, cannot adapt across heterogeneous views. Even combining all models and representations (Group C) does not overcome these fundamental shortcomings, yielding consistently lower scores. DRES addresses these issues via a dynamic two-stage selection. First, it selects the most appropriate feature representation (i.e., the one deemed easiest for classification for the given instance), and then it dynamically identifies locally competent classifiers trained over the selected text

representation. This adaptive mechanism is essential for robust performance in complex fake news detection tasks.

5.2 Comparison with state-of-the-art models

Table 3: Liar dataset: DRES versus state-of-the-art methods. The best results are in bold.

Method	F1-Score
LSTM (GloVe) (Rashkin et al., 2017)	0.210
Hybrid CNN (Word2vec) (Wang, 2017)	0.274
Logistic Regression (GloVe) (Alhindi et al., 2018)	0.250
CNN (Word2Vec) (Girgis et al., 2018)	0.270
MMFD (LSTM + Word2vec) (Karimi et al., 2018)	0.290
Multi-view autoencoder (Pereira et al., 2025)	0.253
DRES (KNORA-E)	**0.371**
DRES (META-DES)	**0.367**
DRES (DES-P)	**0.385**

Table 4: COVID dataset: DRES versus state-of-the-art methods. The best results are in bold.

Method	F1-Score
BiGRU-Attention (BERT) (Karnyoto et al., 2022)	0.919
SVM (TFIDF) (Patwa et al., 2021)	0.933
BERT (End-to-End) (Liu and Chen, 2023)	0.944
DRES (KNORA-E)	**0.973**
DRES (META-DES)	**0.972**
DRES (DES-P)	**0.972**

Table 5: GM dataset: DRES versus state-of-the-art methods. The best results are in bold.

Method	F1-Score
HDSF (Word2vec) (Karimi and Tang, 2019)	0.822
XGBoost (Word2vec) (Reddy et al., 2020)	0.860
XGBoost (GloVe) (Bali et al., 2019)	0.873
Naive Bayes (TFIDF) (Agudelo et al., 2018)	0.881
Ensemble Learning (Elsaeed et al., 2021)	0.946
BiLSTM (GloVe) (Sastrawan et al., 2022)	0.948
DRES (KNORA-E)	**0.986**
DRES (META-DES)	**0.980**
DRES (DES-P)	**0.984**

Tables 3, 4, and 5 compare DRES with state-of-the-art models for the Liar, COVID, and GM datasets, respectively, focusing on text-based fake news detection. DRES achieves the highest macro F1 score on the LIAR dataset (Table 3) using only textual features. It is important to note that the Liar dataset contains additional metadata about the speaker's profile and justification in addition to the news statements that were not used in this analysis, as these auxiliary features are often unavailable or inconsistent across datasets.

For the COVID (Table 4) and GM (Table 5) datasets, the proposed DRES model achieved the top-ranking with an F1-score of 0.971 (with DES-P) and 0.982 (with KNORA-E), respectively. These results highlight our model's capability in another domain (medical information), where reliability is of the highest importance. In all three scenarios, DRES obtained better results than SOTA, regardless of the selected DES algorithm used. Additional experiments with fine-tuned large LLMs are reported in Appendix G, confirming that DRES remains competitive even against stronger contextual baselines.

5.3 Ablation Study

In this section, we evaluate the contribution of view selection and classifier selection phases within the DRES framework by comparing three variants: (i) only view selection and majority voting over classifiers, (ii) only classifier selection on a fixed view (Mistral), and (iii) the DRES system (Table 6).

The proposed dynamic view selection scheme alone significantly improves the baseline Group C, with gains of +10.2, +1.4, and +2.1 percentage points on the Liar, Covid, and GM datasets, respectively. These improvements show that dynamically selecting representations helps address textual ambiguity, particularly for the liar dataset (6 classes). Moreover, when combined with KNORA-E's dynamic classifier selection, DRES consistently achieves further improvements across all datasets. This improvement is more evident for the Liar dataset (+5.6 percentage points), which can be explained by the fact that this is a harder dataset (full hardness heatmap for all datasets are presented in Appendix J). These results highlight the effectiveness of a two-stage selection approach, where dynamic view and classifier selection work together to improve overall performance, particularly in the presence of residual ambiguity (i.e., classification uncertainty that persists after representation selection due to overlap).

The last two lines of Table 6 show the Oracle results for i) the selection of the best representation and ii) the selection of both best representation and best classifier per new instance. The Oracle represents a theoretical upper bound since it always selects the best representation and classifier per query instance. First, perfect representation selection could lead to a +58.9 percentage points improvement over the baseline on Liar, pointing to significant room for improvement in representation selection. Second, achieving nearly perfect accuracy with complete Oracle settings (99.5–100%) validates DRES's architecture when both phases work optimally. Hence, the gap with the Oracle

Table 6: Impact of DRES components. Blue values show improvements (in percentage points, pp) over the baseline (Group C). Oracle configurations (bottom) represent the theoretical upper bounds.

Method	Liar	COVID	GM
Group C (no selection)	0.243	0.941	0.950
Dynamic Ensemble selection only	0.299 (+5.6 pp)	0.948 (+0.7 pp)	0.963 (+1.3 pp)
Representation selection only	0.345 (+10.2 pp)	0.955 (+1.4 pp)	0.971 (+2.1 pp)
DRES (KNORA-E)	**0.371** (+12.8 pp)	**0.961** (+2.0 pp)	**0.982** (+3.2 pp)
DRES + Oracle (representation)	0.832	0.996	0.999
DRES + Oracle (representation + classifier)	0.995	0.999	1.000

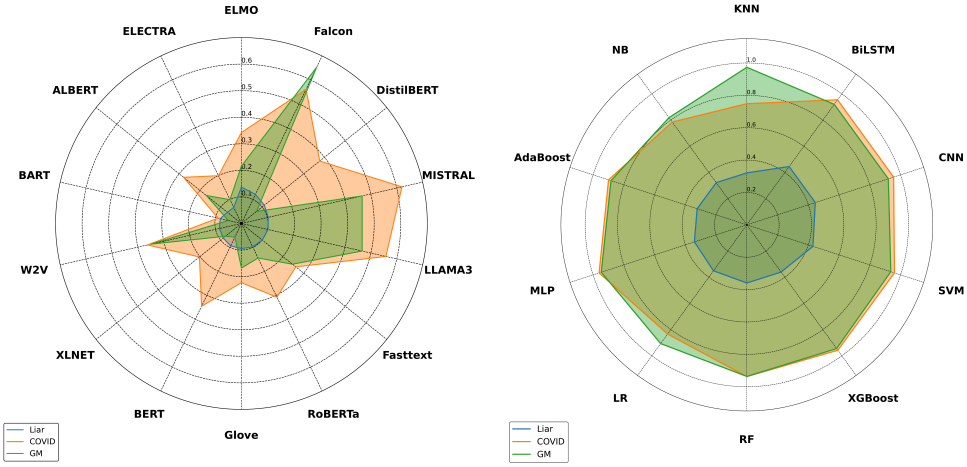


Figure 5: Frequency of selected views (left) and classifiers (right) for each dataset using the KNORA-E dynamic ensemble selection method.

suggests that better instance hardness estimators should be investigated in order to improve the representation selection stage.

5.4 Selection Analysis

DRES framework's instance-specific selection of classifier-view pairs is analyzed in Figure 5 through complementary radar plots showing i) representation selection (left) and ii) classifier selection frequencies (right) across datasets, using KNORA-E for dynamic ensemble selection. The results reveal significant diversity among the chosen views and classifiers. While significant diversity in selected combinations confirms the value of the multi-stage dynamic selection approach, the underutilization of specific components may suggest they could be pruned without affecting overall performance.

5.5 Instance hardness analysis

We investigate how classification difficulty varies across representations by analyzing instance-level hardness scores on the GM dataset. Figure 6 shows the range of hardness scores across views for each instance, computed as the difference between the maximum and minimum values. Over 50% of instances show a range above 0.5 and 25% exceed 0.7, indicating substantial disagreement between representations. Figure 7 further confirms that this variation is widespread and not limited to a few outliers. These findings highlight the limitations of relying on a single representation or a fixed set, as they may perform inconsistently across different inputs. Our results reinforce the motivation for a dynamic selection strategy based on test-time hardness estimation. Full results, including standard deviation and coefficient of variation analyses, as

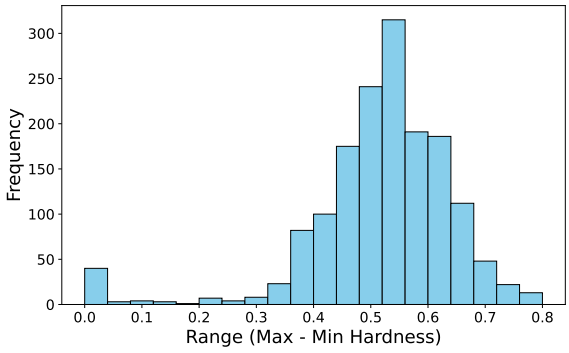


Figure 6: Distribution of the range (max-min) instance hardness computed for the GM dataset.

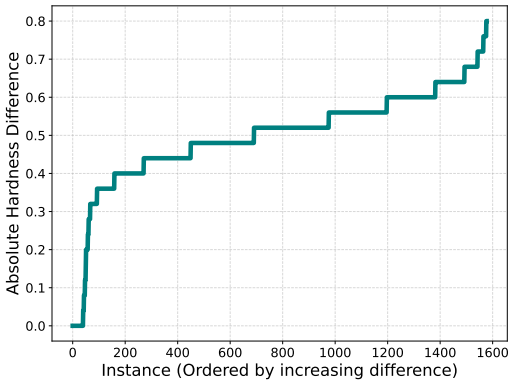


Figure 7: Cumulative distribution of the difference between maximum and minimum instance hardness values for the GM dataset.

well as figures for the Liar and COVID datasets, are provided in Appendix I.

6 Conclusion

In this paper, we introduced DRES, a fake news detection model that dynamically selects the most suitable feature representations and classifier ensembles for each input text by leveraging instance hardness analysis and dynamic classifier selection. We evaluated our approach on three diverse datasets. We found that it outperformed state-of-the-art methods and baseline models, especially when only textual information was considered, which is the most prevalent and accessible form of information in fake news detection. In addition, several ablation studies demonstrate that one must consider a dynamic system in both view and classifier selection in order to obtain higher accuracy. Future work will involve adapting DRES to integrate multiple information sources, such as network propagation and user behavior data, as well as investigating new mechanisms for test-time hardness calculation.

7 Limitations

While DRES advances text-based fake news detection through dynamic representation and classifier selection, three key limitations merit discussion. First, our evaluation focuses on English datasets (Liar, COVID, GM), which limits insights into low-resource languages like Urdu or Arabic, which pose unique challenges (Harris et al., 2025; Albtoush et al., 2025). Furthermore, we did not test DRES on synthetic datasets containing LLM-generated misinformation, despite evidence that detectors often misclassify such content due to linguistic biases (Su et al., 2023). Future work should validate DRES against adversarial LLM outputs to assess its robustness in diverse scenarios.

Second, DRES currently selects only one representation per instance, potentially overlooking complementary signals from views with similar hardness levels. Additionally, our hardness estimation relies solely on the kDN metric, while alternative measures (Ethayarajh et al., 2022) might capture different aspects of instance difficulty. Future work should explore multi-representation fusion strategies and hardness metric ensembles to better handle borderline cases where multiple representations appear equally viable.

Third, while DRES dynamically selects representations and classifiers, its training phase incurs upfront costs from computing instance hardness across multiple representations. As shown in Figure 5, certain representations like BART are selected in <10% of cases across datasets, suggesting potential redundancy. Pruning such infrequently used representations guided by data complexity (Cook et al., 2025) or embedding diversity could simplify the system without compromising performance.

Lastly, it is important to acknowledge the possibility of training data contamination for recently released language models such as LLaMA3 and Mistral. These models were made available after the release of the datasets used in this study (LIAR, COVID, GM), suggesting that portions of these datasets may have been included in their pretraining corpora. This raises concerns about potential memorization effects, particularly in fine-tuning scenario. However, this issue does not compromise the core findings of our work that is: instance-level hardness varies across representation, and dynamic selection of both representation and classifier improves performance.

References

Gerardo Ernesto Rolong Agudelo, Octavio José Salcedo Parra, and Julio Barón Velandia. 2018. Raising a model for fake news detection using machine learning in python. In *Conference on e-Business, e-Services and e-Society*, pages 596–604.

Eman Salamah Albtoush, Keng Hoon Gan, and Saif A Ahmad Alrababa. 2025. Fake news detection: state-of-the-art review and advances with attention to arabic language aspects. *PeerJ Computer Science*, 11:e2693.

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: Improving fact-checking by justification modeling. In *Workshop on Fact Extraction and Verification*, pages 85–90.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Arvinder Pal Singh Bali, Mexson Fernandes, Sourabh Choubey, and Mahima Goel. 2019. Comparative performance of machine learning algorithms for fake news detection. In *Advances in Computing and Data Sciences*, pages 420–430. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Ryan A Cook, John P Lalor, and Ahmed Abbasi. 2025. No simple answer to data complexity: An examination of instance-level complexity metrics for classification tasks. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2553–2573.

Rafael MO Cruz, Woshington V de Sousa, and George DC Cavalcanti. 2022. Selecting and combining complementary feature representations and classifiers for hate speech detection. *Online Social Networks and Media*, 28:100194.

Rafael MO Cruz, Luiz G Hafemann, Robert Sabourin, and George DC Cavalcanti. 2020. DESlib: A Dynamic ensemble selection library in Python. *Journal of Machine Learning Research*, 21(8):1–5.

Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. 2018. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216.

Rafael MO Cruz, Robert Sabourin, George DC Cavalcanti, and Tsang Ing Ren. 2015. META-DES: A dynamic ensemble selection framework using meta-learning. *Pattern Recognition*, 48(5):1925–1935.

Rafael MO Cruz, Hiba H Zakane, Robert Sabourin, and George DC Cavalcanti. 2017. Dynamic ensemble selection vs k-nn: why and when dynamic selection obtains higher classification performance? In *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE.

Reza Davtalab, Rafael MO Cruz, and Robert Sabourin. 2024. A scalable dynamic ensemble selection using fuzzy hyperboxes. *Information Fusion*, 102:102036.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15.

Eman Elsaeed, Osama Ouda, Mohammed M Elmogy, Ahmed Atwan, and Eman El-Daydamony. 2021. Detecting fake news in social media using voting classifier. *IEEE Access*, 9:161909–161925.

Ehab Essa, Karima Omar, and Ali Alqahtani. 2023. Fake news detection based on a hybrid bert and lightgbm models. *Complex & Intelligent Systems*, 9(6):6581–6592.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pages 5988–6008.

Faramarz Farhangian, Rafael MO Cruz, and George DC Cavalcanti. 2024. Fake news detection: Taxonomy and comparative study. *Information Fusion*, 103:102140.

Akansha Gautam, V Venktesh, and Sarah Masud. 2021. Fake news detection system using xlnet model with topic distributions: Constraint@aaai2021 shared task. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, pages 189–200.

Sherry Girgis, Eslam Amer, and Mahmoud Gadallah. 2018. Deep learning algorithms for detecting fake news in online text. In *International Conference on Computer Engineering and Systems*, pages 93–97.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Sheetal Harris, Jinshuo Liu, Hassan Jalil Hadi, Naveed Ahmad, and Mohammed Ali Alshara. 2025. Benchmarking hook and bait urdu news dataset for domain-agnostic and multilingual fake news detection using large language models. *Scientific Reports*, 15(1):1–14.

Feiran Huang, Xiaoming Zhang, Jie Xu, Zhonghua Zhao, and Zhoujun Li. 2019. Multimodal learning of social image representation by exploiting social relations. *IEEE Transactions on Cybernetics*, 51(3):1506–1518.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix,

and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia Tools and Applications*, 80(8):11765–11788.

Hamid Karimi, Proteek Roy, Sari Saba-Sadiya, and Jiliang Tang. 2018. Multi-source multi-class fake news detection. In *International Conference on Computational Linguistics*, pages 1546–1557.

Hamid Karimi and Jiliang Tang. 2019. Learning hierarchical discourse-level structure for fake news detection. *arXiv preprint arXiv:1903.07389*.

Andrea Stevens Karnyoto, Chengjie Sun, Bingquan Liu, and Xiaolong Wang. 2022. Transfer learning and gru-crf augmentation for covid-19 fake news detection. *Computer Science and Information Systems*, 19(2):639–658.

Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Albert HR Ko, Robert Sabourin, and Alceu Souza Britto Jr. 2008. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Junjie Liu and Min Chen. 2023. Covid-19 fake news detector. In *International Conference on Computing, Networking and Communications*, pages 463–467.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Alexander Loth, Martin Kappes, and Marc-Oliver Pahl. 2024. Blessing or curse? a survey on the impact of generative ai on fake news. *arXiv preprint arXiv:2404.03021*.

George McIntire. 2017. Mcintire fake news dataset. Available online at: https://github.com/lutzhamel/fake-news, last accessed on 15.09.2025.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Federico Monti, Fabrizio Frasca, Davide Eynard, Damon Mannion, and Michael M Bronstein. 2019. Fake news detection on social media using geometric deep learning. *arXiv preprint arXiv:1902.06673*.

V. Nguyen, K. Sugiyama, P. Nakov, and M. Y. Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *ACM International Conference on Information & Knowledge Management*, page 1165–1174.

Pedro Yuri Arbs Paiva, Camila Castro Moreno, Kate Smith-Miles, Maria Gabriela Valeriano, and Ana Carolina Lorena. 2022. Relating instance hardness to classification performance in a dataset: a visual approach. *Machine Learning*, 111(8):3085–3123.

Parth Patwa, Shivam Sharma, Srinivas Pykl, Vineeth Guptha, Gitanjali Kumari, Md Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. 2021. Fighting an infodemic: Covid-19 fake news dataset. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation*, pages 21–29.

Liwen Peng, Songlei Jian, Zhigang Kan, Linbo Qiao, and Dongsheng Li. 2024. Not all fake news is semantically similar: Contextual semantic representation learning for multimodal fake news detection. *Information Processing & Management*, 61(1):103564.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Ingryd VST Pereira, George DC Cavalcanti, and Rafael MO Cruz. 2025. Multi-view autoencoders for fake news detection. In *IEEE Symposium on Computational Intelligence in Natural Language Processing and Social Media (CI-NLPSoMe)*.

Verónica Pérez-Rosas, Bennett Kleinberg, Anna Lefevre, and Rada Mihalcea. 2018. Automatic detection of fake news. In *International Conference on Computational Linguistics*, pages 3391–3401.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint*.

M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein. 2018. A stylometric inquiry into hyperpartisan and fake news. In *Annual Meeting of the Association for Computational Linguistics*, pages 231–240.

Peng Qi, Juan Cao, Tianyun Yang, Junbo Guo, and Jintao Li. 2019. Exploiting multi-domain visual information for fake news detection. In *IEEE International Conference on Data Mining*, pages 518–527.

Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937.

Harita Reddy, Namratha Raj, Manali Gala, and Annappa Basava. 2020. Text-mining-based fake news detection using ensemble methods. *International Journal of Automation and Computing*, 17(2):210–221.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. CSI: A Hybrid Deep Model for Fake News Detection. In *ACM on Conference on Information and Knowledge Management*, pages 797–806.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

I Kadek Sastrawan, IPA Bayupati, and Dewa Made Sri Arsa. 2022. Detection of fake news using deep learning cnn–rnn based methods. *ICT Express*, 8(3):396–408.

Xiaorong Shen, Maowei Huang, Zheng Hu, Shimin Cai, and Tao Zhou. 2024. Multimodal fake news detection with contrastive learning and optimal transport. *Frontiers in Computer Science*, 6:1473457.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36.

Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond news contents: The role of social context for fake news detection. In *ACM International Conference on Web Search and Data Mining*, pages 312–320.

Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. 2019. Spotfake: A multi-modal framework for fake news detection. In *International Conference on Multimedia Big Data*, pages 39–47.

Sneha Singhania, Nigel Fernandez, and Shrisha Rao. 2017. 3HAN: A deep neural network for fake news detection. In *International Conference on Neural Information Processing*, pages 572–581.

Michael R Smith, Tony Martinez, and Christophe Giraud-Carrier. 2014. An instance level analysis of data complexity. *Machine Learning*, 95:225–256.

Mariana de Araujo Souza, Robert Sabourin, George DC Cavalcanti, and Rafael MO Cruz. 2024. A dynamic multiple classifier system using graph neural network for high dimensional overlapped data. *Information Fusion*, 103:102145.

Jinyan Su, Terry Yue Zhuo, Jonibek Mansurov, Di Wang, and Preslav Nakov. 2023. Fake news detectors are biased against texts generated by large language models. *arXiv preprint arXiv:2309.08674*.

Ciprian-Octavian Truică, Elena-Simona Apostol, and Panagiotis Karras. 2024. DANES: Deep neural network ensemble architecture for social and textual context-aware fake news detection. *Knowledge-Based Systems*, 294:111715.

Ciprian-Octavian Truică and Elena-Simona Apostol. 2022. MisrobÆrta: Transformers versus misinformation. *Mathematics*, 10(4).

Aryan Verma, P Priyanka, Tayyab Khan, Karan Singh, Lawal O Yesufu, Mazeyanti Mohd Ariffin, and Ali Ahmadian. 2025. Scrutnet: a deep ensemble network for detecting fake news in online text. *Social Network Analysis and Mining*, 15(1):21.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Annual Meeting of the Association for Computational Linguistics*, pages 422–426.

Tomasz Woloszynski, Marek Kurzynski, Pawel Podsiadlo, and Gwidon W Stachowiak. 2012. A measure of competence based on random classification for dynamic ensemble selection. *Information Fusion*, 13(3):207–213.

David H Wolpert. 1992. Stacked generalization. *Neural Networks*, 5(2):241–259.

Kevin Woods, W. Philip Kegelmeyer, and Kevin Bowyer. 1997. Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):405–410.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32.

A Supplementary details on representations

Table 7 provides a comprehensive list of the models, their implementation sources (e.g., HuggingFace, Zeugma, AllenNLP), and embedding dimensions used in our experiments. This includes both classical word embedding methods (e.g., Word2Vec, GloVe) and modern pretrained language models (e.g., BERT, LLaMA, Mistral).

Table 7: Text representation techniques used in this study.

Model	Version/Name	Source	Dimension
Word2Vec	word2vec	Zeugma	300
GloVe	glove	Zeugma	300
FastText	fasttext	Zeugma	300
ELMO	allenai/elmo	AllenNLP	1024
BERT	bert-base-uncased	HuggingFace	768
DistilBERT	distilbert-base-uncased	HuggingFace	768
ALBERT	albert-base-v2	HuggingFace	768
RoBERTa	roberta-base	HuggingFace	768
BART	facebook/bart-base	HuggingFace	768
ELECTRA	google/electra-base-discriminator	HuggingFace	768
XLNet	xlnet-base-cased	HuggingFace	768
LLaMA	CodeLlama-7b	HuggingFace	4096
Falcon	falcon	HuggingFace	2048
LLaMA3	Llama3.2-1B	HuggingFace	2048
Mistral	mistral-7B-v0.1	HuggingFace	4096

B Hyperparameters for classical and ensemble models

Table 8: Hyperparameters considered for machine learning and ensemble models.

Method	Hyperparameters
SVM	Kernel:['rbf'] Gamma: [1, 0.1, 0.01, 0.001, 0.0001] C: [0.1, 1, 10, 100, 1000]
LR	solver: ['liblinear'] penalty: ['none', 'l1', 'l2', 'elasticnet'] C: [0.01, 0.1, 1, 10, 100]
NB	alpha: [0.1, 0.5, 1] fit_prior: [False, True]
KNN	n_neighbors: [1 - 20]
RF	bootstrap: [True, False] max_depth: [5,10, 20, 30, 40, 50] max_features: ['auto', 'sqrt', 'log2'] min_samples_leaf: [1, 2, 4] min_samples_split: [2, 5, 10] n_estimators: [200, 400, 600, 800, 1000] criterion: ['gini', 'entropy']
AdaBoost	n_estimators: [10, 50, 100, 200, 300, 400, 500, 1000] learning_rate: [0.001, 0.01, 0.1, 0.2, 0.5]
XGBoost	n_estimators: [200,300,400,500] max_features: ['sqrt', 'log2'] max_depth: [4,5,6,7,8] criterion: ['gini', 'entropy'] random_state: [18]
MLP	Activation function: [ReLU, logistic] solver: [Adam, lbfgs]

Table 8 outlines the grid search ranges used to tune classical and ensemble models. For SVM, we tuned the kernel coefficient (gamma) and regularization strength (C), which are known to have the most significant impact on model performance. Logistic regression was configured across solvers and regularization types. Naive Bayes used different smoothing values and prior settings. For KNN, we adjusted the number of neighbors. Random Forest, AdaBoost, and XGBoost were tuned by varying tree depths, number of estimators, and split criteria. MLP models were tested with different activation functions and solvers.

C Hyperparameters for deep learning models

Table 9 shows the tuning setup for CNN and BiLSTM models. Both were trained using the Adam optimizer with a fixed learning rate and dropout. We varied activation functions, batch sizes, and the number of epochs. For BiLSTM, we also included hidden size as a tuning parameter.

Table 9: Hyperparameters considered for all deep learning models.

Method	Setup
CNN	Activation function: [sigmoid, ReLU] Batch size: [64, 128, 512] Number of epochs: [5,20,100] Optimizer: [Adam] Learning rate: 0.001 Dropout: 0.2
BiLSTM	Activation function: [sigmoid, ReLU] Batch size: [64, 128, 512] Number of epochs: [5,20,100] Optimizer: [Adam] Learning rate: 0.001 Hidden size: 128 Dropout: 0.2

D The impact of the k-hardness hyperparameter

We conducted additional experiments to assess the impact of the number of neighbors (k) used in test-time instance hardness estimation. We evaluated DRES with $k \in \{3, 5, 7, 9, 11, 13\}$. The results are presented in Table 10.

The results indicate that DRES is robust to the choice of k, with only marginal differences in F1-scores across values. While $k = 5$ slightly outperforms other settings, the variation remains within a narrow range, suggesting that the method does not rely heavily on fine-tuning this parameter.

Table 10: Impact of k on DRES performance across datasets.

Method	k	COVID	GM	Liar
DRES (KNORA-E)	3	0.964 (0.002)	0.981 (0.002)	0.366 (0.003)
DRES (KNORA-E)	5	0.973 (0.002)	0.986 (0.002)	0.371 (0.003)
DRES (KNORA-E)	7	0.968 (0.007)	0.984 (0.005)	0.366 (0.005)
DRES (KNORA-E)	9	0.97 (0.001)	0.982 (0.004)	0.365 (0.004)
DRES (KNORA-E)	11	0.965 (0.002)	0.981 (0.006)	0.367 (0.003)
DRES (KNORA-E)	13	0.971 (0.002)	0.979 (0.001)	0.366 (0.002)
DRES (META-DES)	3	0.969 (0.001)	0.971 (0.001)	0.362 (0.007)
DRES (META-DES)	5	0.972 (0.002)	0.98 (0.005)	0.367 (0.002)
DRES (META-DES)	7	0.97 (0.009)	0.975 (0.002)	0.363 (0.002)
DRES (META-DES)	9	0.964 (0.002)	0.974 (0.003)	0.361 (0.001)
DRES (META-DES)	11	0.97 (0.001)	0.977 (0.001)	0.362 (0.002)
DRES (META-DES)	13	0.966 (0.001)	0.979 (0.006)	0.363 (0.001)
DRES (DES-P)	3	0.968 (0.001)	0.976 (0.003)	0.381 (0.004)
DRES (DES-P)	5	0.972 (0.003)	0.984 (0.002)	0.385 (0.003)
DRES (DES-P)	7	0.969 (0.001)	0.981 (0.001)	0.381 (0.003)
DRES (DES-P)	9	0.971 (0.004)	0.977 (0.002)	0.383 (0.002)
DRES (DES-P)	11	0.968 (0.001)	0.976 (0.007)	0.38 (0.008)
DRES (DES-P)	13	0.966 (0.001)	0.976 (0.003)	0.377 (0.009)

E Performance for other metrics

Table 11 reports DRES's performance for the precision results, while Table 12 reports the recall results.

Table 11: Precision performance of DRES models across datasets.

	Dataset		
Method	Liar	COVID	GM
DRES (KNORA-E)	0.380 (0.003)	0.970 (0.002)	0.985 (0.002)
DRES (META-DES)	0.377 (0.002)	0.968 (0.002)	0.978 (0.004)
DRES (DES-P)	0.390 (0.003)	0.970 (0.003)	0.983 (0.002)

Table 12: Recall performance of DRES models across datasets.

	Dataset		
Method	Liar	COVID	GM
DRES (KNORA-E)	0.365 (0.003)	0.975 (0.002)	0.988 (0.002)
DRES (META-DES)	0.358 (0.002)	0.975 (0.002)	0.982 (0.006)
DRES (DES-P)	0.380 (0.003)	0.974 (0.003)	0.985 (0.002)

F Static ensemble results

Table 13 expands upon the main manuscript's findings by detailing the performance of individual models within Groups A, B, and C across the Liar, COVID, and GM datasets. The results reveal that within each group, ensemble models exhibit similar performance levels, indicating that static combination methods, even when combined with a meta-classifier with stacked generalization (i.e., a learned combination scheme).

In contrast, the DRES framework, when integrated with dynamic ensemble selection methods such as KNORA-E, META-DES, and DES-P, consistently outperforms the baseline groups across all datasets. Notably, the choice among these dynamic selection techniques results in marginal performance differences, suggesting that the primary advantage arises from the dynamic selection mechanism itself rather than the specific competence estimation heuristics and classifier selection approaches they employ (Cruz et al., 2018).

Table 13: Results of F1-Score per dataset for DRES and baseline models (Groups A, B, and C).

	Dataset		
Method	Liar	COVID	GM
LR (Group A)	**0.260**	0.940	**0.950**
SVM (Group A)	**0.260**	0.940	0.940
KNN (Group A)	0.230	0.930	0.890
NB (Group A)	0.220	0.920	0.860
XGBoost (Group A)	0.250	0.940	**0.950**
RF (Group A)	**0.260**	0.920	0.920
AdaBoost (Group A)	0.190	0.910	0.920
BiLSTM (Group A)	0.250	0.930	0.920
CNN (Group A)	0.250	0.940	0.920
MLP (Group A)	0.250	**0.950**	**0.950**
TF (Group B)	0.250	0.930	0.940
TFIDF (Group B)	0.230	0.940	0.950
W2V (Group B)	0.230	0.910	0.920
GloVe (Group B)	0.220	0.860	0.840
FastText (Group B)	0.240	0.910	0.900
ELMO (Group B)	0.240	0.910	0.920
BERT (Group B)	0.230	0.910	0.820
DistilBERT (Group B)	0.220	0.920	0.830
RoBERTa (Group B)	0.230	0.910	0.850
ALBERT (Group B)	0.240	0.900	0.840
BART (Group B)	0.220	0.880	0.850
ELECTRA (Group B)	0.230	0.910	0.850
XLNET (Group B)	0.220	0.900	0.860
Falcon (Group B)	0.250	0.940	0.950
LLaMA3 (Group B)	**0.261**	0.940	0.950
Mistral (Group B)	0.260	**0.943**	**0.951**
Group C	0.243	0.941	0.950
DRES + KNORA-E	0.371	**0.973**	**0.986**
DRES + META-DES	0.367	0.972	0.980
DRES + DES-P	**0.385**	0.972	0.984

G End-to-end LLM fine-tuned results

As shown in Table 14, DRES consistently outperforms both end-to-end fine-tuned LLMs and static ensemble strategies across all datasets. While recent models such as LLaMA3 and Mistral achieve strong F1 scores, particularly on the GM and COVID datasets, the DRES variants surpass them by a clear margin, with gains of up to 11 F1 points on the LIAR dataset and around 2 to 4 points on the others. In contrast, static ensemble baselines

Table 14: End-to-end fine-tuned model results (F1-score (std)) across the LIAR, COVID, and GM datasets. The absolute best results per dataset are in bold, and the top DRES methods that are statistically equivalent are marked with an asterisk.

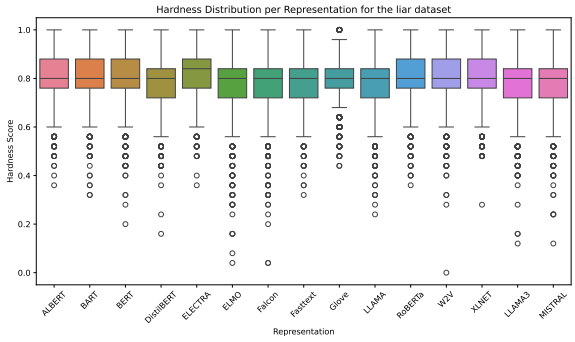
	Dataset		
Model	**Liar**	**COVID**	**GM**
ELMO	0.210 (0.001)	0.935 (0.001)	0.928 (0.005)
BERT	0.210 (0.003)	0.880 (0.002)	0.880 (0.006)
DistilBERT	0.220 (0.008)	0.940 (0.006)	0.890 (0.003)
ALBERT	0.180 (0.000)	0.840 (0.001)	0.870 (0.001)
BART	0.200 (0.001)	0.890 (0.002)	0.940 (0.001)
RoBERTa	0.190 (0.010)	0.850 (0.009)	0.850 (0.009)
ELECTRA	0.190 (0.002)	0.860 (0.004)	0.890 (0.001)
XLNET	0.210 (0.003)	0.860 (0.003)	0.890 (0.011)
LLaMA	0.261 (0.003)	0.941 (0.002)	0.992 (0.003)
Falcon	0.258 (0.001)	0.948 (0.001)	0.993 (0.000)
Mistral	0.268 (0.002)	0.950 (0.002)	0.994 (0.001)
LLaMA3	0.270 (0.003)	0.953 (0.002)	**0.995 (0.001)**
MLP (Group A)	0.250 (0.002)	0.950 (0.002)	0.950 (0.003)
Mistral (Group B)	0.260 (0.002)	0.943 (0.003)	0.951 (0.002)
Group C	0.243 (0.003)	0.941 (0.003)	0.950 (0.002)
DRES + KNORA-E	0.371 (0.003)	**0.973 (0.002)***	0.986 (0.002)*
DRES + META-DES	0.367 (0.002)	0.972 (0.002)*	0.980 (0.005)
DRES + DES-P	**0.385 (0.003)***	0.972 (0.003)*	0.984 (0.002)*

(Groups A, B, and C) do not outperform the best fine-tuned LLMs, indicating that simple model aggregation provides limited benefits in this setting. These results emphasize the effectiveness of DRES's dynamic selection strategy in addressing instance-specific challenges.

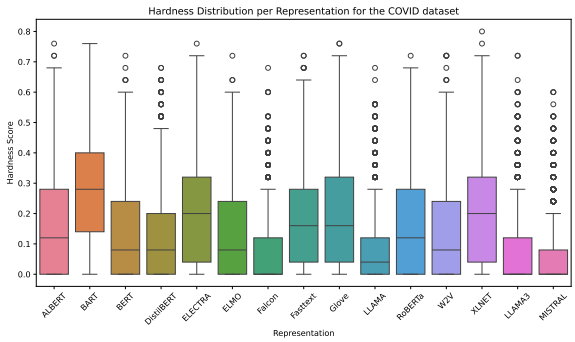
It is important to acknowledge the possibility of training data contamination for recently released language models such as LLaMA3 and Mistral. These models were made available after the release of the datasets used in this study (LIAR, COVID, GM), suggesting that portions of these datasets may have been included in their pretraining corpora. This raises concerns about potential memorization effects, particularly in fine-tuned or zero-shot scenarios.

H Instance Hardness Distribution Analysis

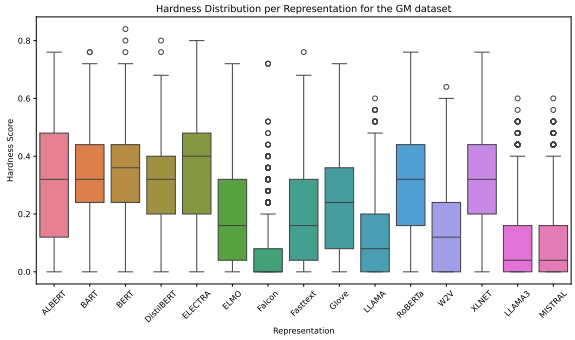
This section presents the distribution of instance hardness (IH) across various text representations and datasets. As shown in Figure 8, IH scores vary not only between datasets but also across representations within the same dataset. These distributions reflect differences in data complexity. For instance, GM exhibits overall lower hardness values than LIAR, while COVID displays greater variability depending on the chosen feature space. Such variation helps explain why static combination strategies often fall short—some representations may work well globally, but others introduce noise or redundancy when applied uniformly.



(a) Liar Dataset



(b) COVID Dataset



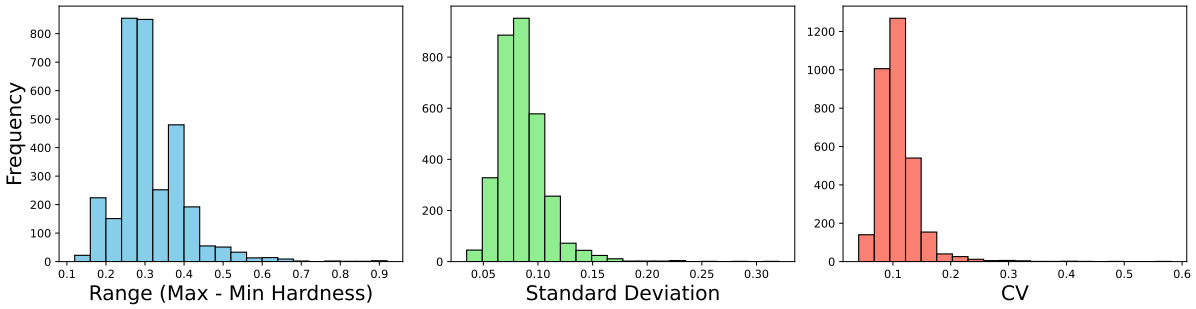
(c) GM Dataset

Figure 8: Boxplot showing the hardness distribution for the Liar, COVID and GM datasets.

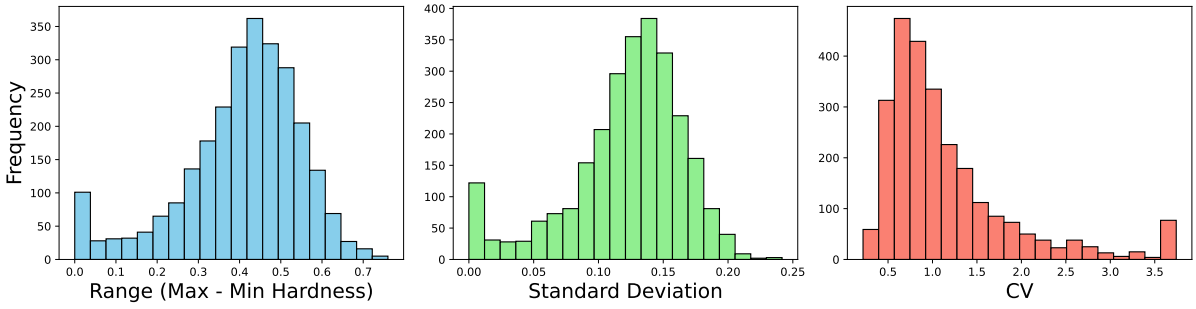
These plots offer a global view of how separable or difficult samples are under each representation, supporting the idea that representation quality is highly data- and model-dependent. However, while useful for assessing overall trends, these distributions do not capture per-instance variation. That is, a representation with good average performance might still perform poorly on specific inputs. These analyses are conducted in the following section.

I Analysis of Hardness Variation Across Representations

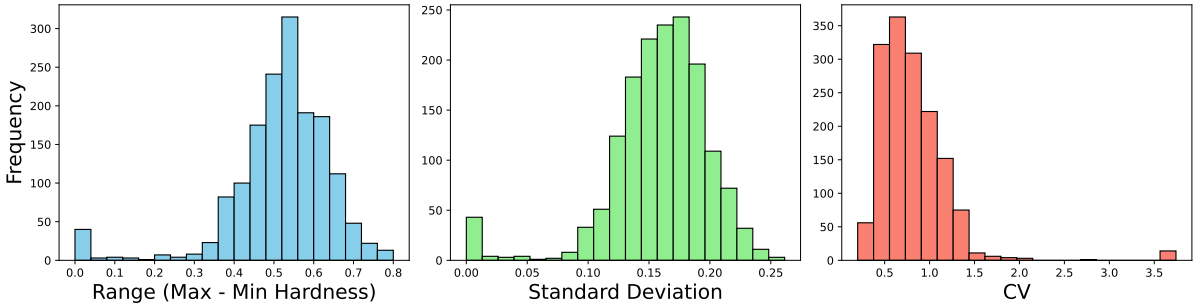
We analyze how instance hardness varies across different text representations using two complementary perspectives. First, we compute range, vari-



(a) Liar dataset



(b) COVID dataset



(c) GM dataset

Figure 9: Descriptive statistics showing (left) range, (center) standard deviation, and (right) coefficient of variation for each dataset.

ance, and coefficient of variation (CV) as summary statistics to capture the dispersion of hardness values across representations for each instance. Then, we present a sorted hardness gap profile to visualize the differences in maximum and minimum hardness values across views.

As shown in Figure 9, the range is defined as the difference between the maximum and minimum hardness scores per instance. The LIAR dataset exhibits a broader spread, with most values under 0.5 but a long tail approaching 0.9. The GM and COVID datasets show tighter distributions, with most instances higher than 0.4, indicating that there is a significant hardness difference across representations for the majority of cases.

In addition to summary statistics, Figure 10 shows instance-wise hardness profiles sorted in ascending order of the gap between the maximum and minimum hardness scores. These plots reveal substantial disagreement across views. For instance, in the GM dataset, over 50% of instances show a gap of at least 0.5, and 25% exceed 0.7. The COVID and LIAR datasets follow similar patterns, with significant fractions of instances showing gaps larger than 0.4.

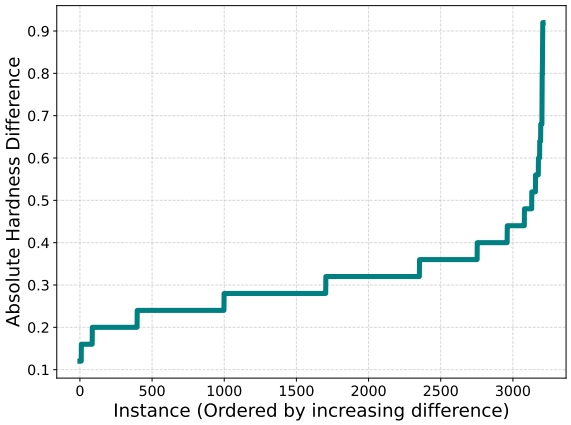
These findings further support the motivation behind our dynamic instance hardness representation selection approach. Instead of relying on a fixed or fused representation, DRES uses test-time hardness estimation to adaptively select the most appropriate view. The observed variation in hardness supports this strategy, as many instances would be suboptimally handled by any single representation or by combining all representations.

J Instance Hardness Heatmaps

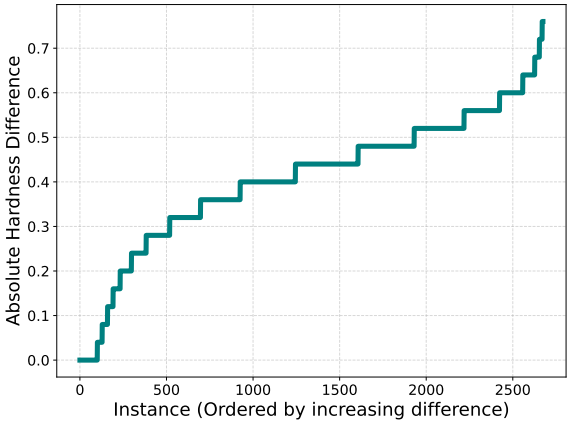
Figure 11 shows heatmaps of instance hardness scores across all representations for the LIAR, COVID, and GM datasets. Each row corresponds to a representation, and each column to a sample. The LIAR dataset displays consistently higher hardness values across representations, with many samples appearing difficult to classify regardless of the embedding space. In contrast, COVID and GM show larger regions of low hardness, indicating more apparent class separation and more consistent behavior across representations. These patterns help explain the results in Table 6, where LIAR shows the most significant performance gain (+5.6 percentage points) when using dynamic ensemble selection (DES) alone. This improvement can be explained by the DES's ability to deal with high disagreement between classifiers and better handle harder instances (Cruz et al., 2017).

K Usage of AI Assistants

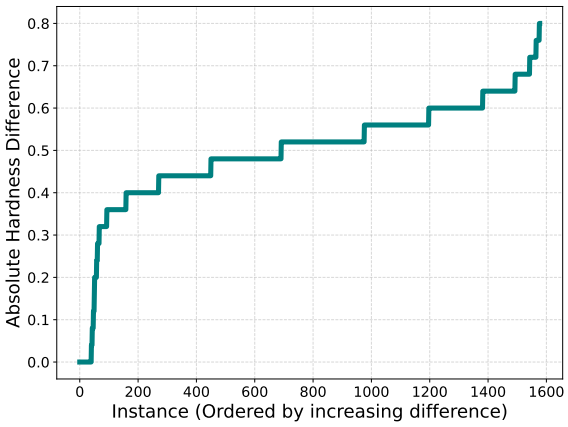
The authors used AI tools (ChatGPT and Grammarly) to help the manuscript writing process, specifically for revising grammar, improving clarity, and checking mathematical notation consistency. AI was also used to assist with coding tasks for data analysis and visualizations (e.g., heatmaps, radar plots, and cumulative distributions). All content and code were reviewed, verified, and finalized by the authors.



(a) Liar Dataset

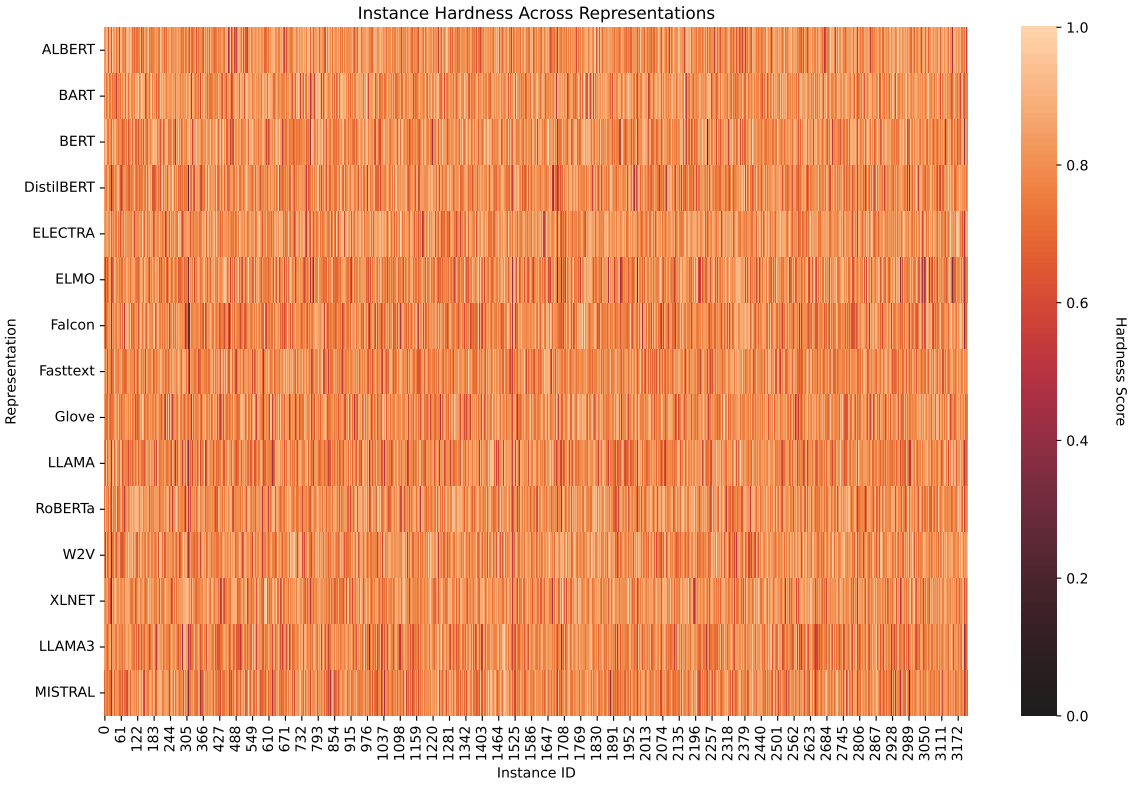


(b) COVID Dataset

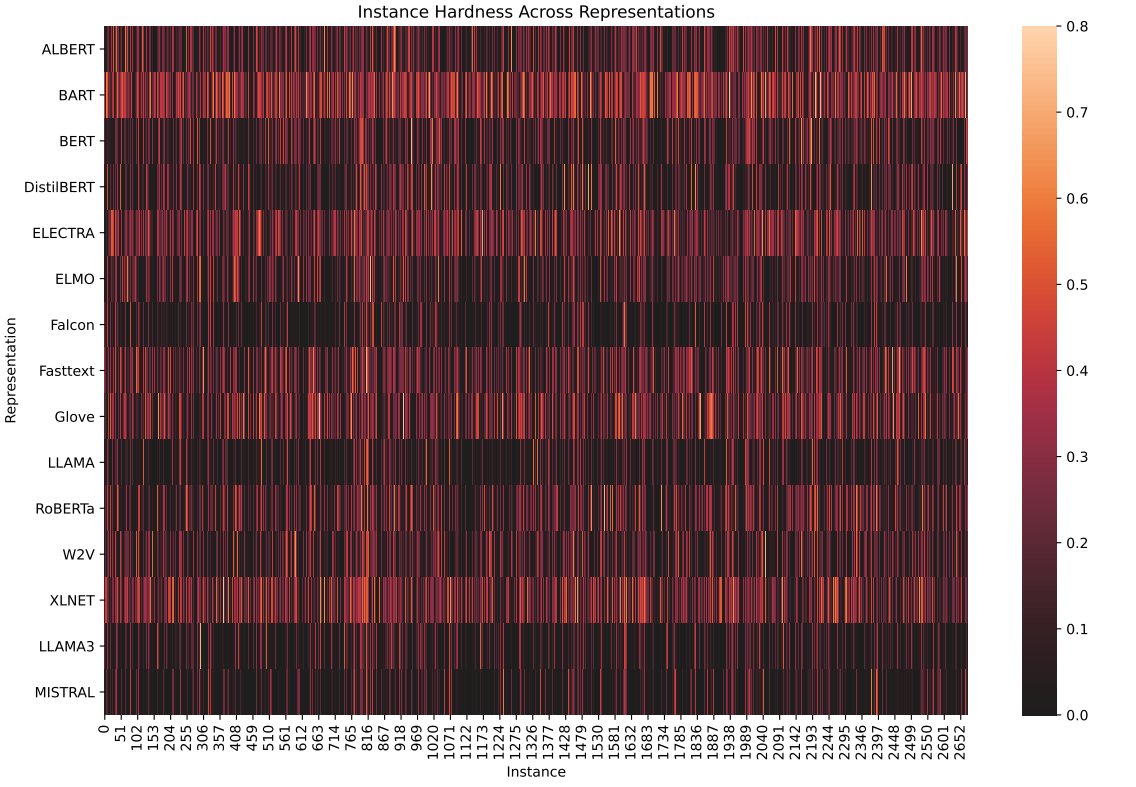


(c) GM Dataset

Figure 10: Cumulative distribution of the difference between maximum and minimum IH values across datasets.

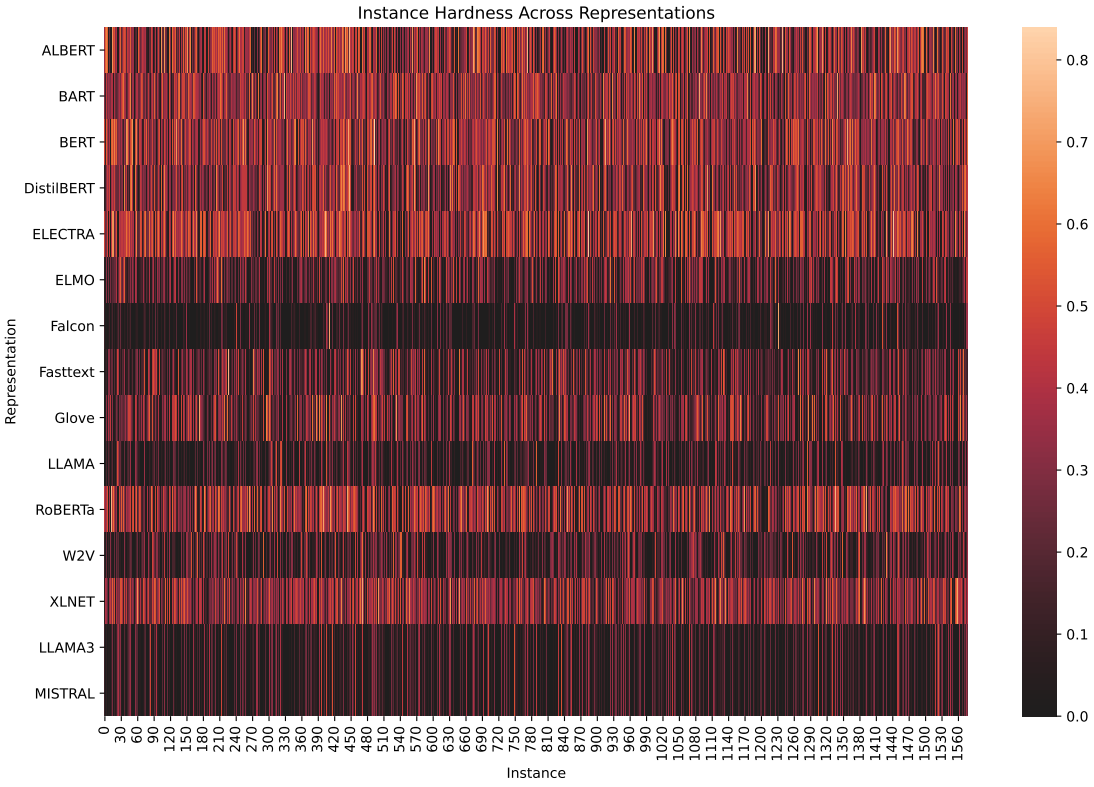


(a) Liar dataset



(b) COVID dataset

Figure 11: Instance hardness heatmaps across datasets (a)–(b).



(c) GM dataset

Figure 11: (Continued) Instance hardness heatmaps across datasets (c).