Optimized Statistical Ranking is All You Need for Robust Coreset Selection in Efficient Transformer-Based Spam Detection

Aisha Hamad Hassan, Tushar Shinde
MIDAS (Multimedia Intelligence, Data Analysis and compreSsion) Lab
Indian Institute of Technology Madras, Zanzibar, Tanzania
shinde@iitmz.ac.in

Abstract

Spam detection, particularly in resource-constrained environments, remains a challenging task due to issues like class imbalance, noisy text, and large-scale data requirements. Transformer-based models have demonstrated state-of-the-art performance in text classification tasks; however, their reliance on large datasets makes training computationally expensive and often impractical for real-world applications. To address this, we introduce a novel coreset selection strategy for efficient spam detection, leveraging a unified Uncertainty-Diversity Ranking (UDR) framework. Our method combines uncertainty-based entropy measures with diversity-driven techniques, ensuring that high-uncertainty samples are prioritized for training while promoting diversity within the selected coreset. The proposed approach supports multiple coreset selection strategies, including Top-K, Bottom-K, and adaptive schemes, making it flexible across various use cases. Moreover, our method implicitly addresses class imbalance by balancing uncertain samples across different classes, ensuring that minority classes are adequately represented. We evaluate the effectiveness of our approach on several benchmark spam detection datasets, including UCI SMS, UTKML Twitter, and LingSpam. Experimental results show that our method achieves competitive performance in terms of accuracy, precision, and recall, while significantly reducing the size of the training data. This results in faster training times and lower computational costs, making our approach particularly suitable for mobile devices, low-power communication systems, and other resource-limited environments.

1 Introduction and Related Work

Spam detection, especially in the context of mobile SMS filtering, has gained significant attention in recent years due to its critical importance in ensuring user satisfaction and mitigating financial losses for service providers Abdulhamid et al. [2017], Liu et al. [2021], Al Saidat et al. [2024]. The rise of spam is largely attributed to the increasing number of mobile users and the low cost associated with sending messages. Despite considerable advancements in the field, spam detection remains a challenging task in natural language processing (NLP), primarily due to factors such as class imbalance, noisy text, unstructured message formats, and the absence of large-scale, real-world annotated datasets Oyeyemi and Ojo [2024], Xia and Chen [2020]. Furthermore, service providers must constantly improve their spam detection technologies to combat increasingly sophisticated spam tactics Xia and Chen [2020]. Transformer-based models, such as BERT and RoBERTa, have revolutionized text classification tasks, achieving state-of-the-art performance Devlin et al. [2019], Pal et al. [2025], Zhang et al. [2025]. However, these models often require large-scale

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Optimization for Machine Learning.

datasets, making their training computationally expensive and often unnecessary for achieving optimal performance Abdulhamid et al. [2017].

To mitigate the computational burden, coreset selection has emerged as an effective strategy. This technique involves selecting a small, yet highly informative, subset of the training data that reduces computational cost without compromising model performance Xia et al. [2023], Shinde and Madabhushi, Shinde, Shinde et al. [2025], Shinde and Sharma. Existing coreset selection strategies include random sampling, uncertainty-based methods such as entropy, margin, and gradient-based approaches, as well as diversity-driven techniques based on clustering Guo et al. [2022], Chai et al. [2023]. However, these methods often struggle to simultaneously balance uncertainty and coverage, especially under conditions of class imbalance, which is a common issue in spam detection tasks.

The remainder of the paper is organized as follows: Section 2 presents the proposed architecture and modeling pipeline, Section 3 describes the datasets, preprocessing, and training setup, Section 4 reports the experimental results, and Section 5 concludes with future directions.

Our Contributions. In this paper, we propose a novel Entropy + Density Ranking strategy for coreset selection that addresses key challenges in spam detection:

- It combines entropy-based and density-based uncertainty to prioritize ambiguous and informative samples.
- It supports multiple coreset selection strategies, including Top-K, Bottom-K, and novel adaptive schemes.
- It ensures the diversity of the selected coreset, mitigating the risk of overfitting.
- It addresses the challenge of class imbalance by implicitly balancing uncertain samples across different classes.

We demonstrate the effectiveness of our approach through extensive experiments on benchmark SMS, email, and social media spam datasets, showing that our method achieves high performance while significantly reducing the required training data.

2 Method

In this work, we propose two methods for coreset selection and ranking: Entropy-based Uncertainty and Class Balanced Uncertainty Density Ranking (CBUDR). These methods are designed to identify the most informative and uncertain samples for efficient model training. Below, we describe the components and methodology behind each approach. We introduce a coreset ranking scheme that combines uncertainty based on entropy with density-based uncertainty. This combined approach allows us to prioritize both uncertain and representative samples, leading to efficient model training.

2.1 Entropy-based Ranking

To begin, each input sample x_i is first converted into a dense embedding e_i using a pre-trained text encoder, such as Sentence BERT (SBERT). The embedding represents the input data in a high-dimensional continuous space. Next, a proxy classifier predicts the class probabilities for each sample. Let $\mathbf{p}(x_i) = [p_1, p_2, \ldots, p_C]$ represent the predicted probabilities for C classes (e.g., spam vs ham). The uncertainty of the prediction is then measured using Shannon entropy, which quantifies the unpredictability or uncertainty associated with a given class prediction. The entropy is defined as:

$$U(x_i) = -\sum_{c=1}^{C} p_c \log p_c \tag{1}$$

where p_c is the predicted probability for class c. High entropy indicates higher uncertainty, meaning the model is less confident in its prediction, which is useful for selecting ambiguous samples for further training.

2.2 Class-Balanced Uncertainty-Density Ranking (CBUDR)

In the CBUDR approach, we combine uncertainty based on entropy with density-based ranking to select samples that are both uncertain and representative Hassan and Shinde [2025]. This combination

ensures that we prioritize samples that are difficult for the model to classify and those that are well-distributed across the data space. We first compute the class-normalized uncertainty for each sample. The uncertainty $U(x_i)$ is normalized across all samples in the same class C_c to account for the class distribution. The class-normalized uncertainty is given by:

$$U_c(x_i) = \frac{U(x_i)}{\max_{x \in C_c} U(x)} \tag{2}$$

where C_c is the set of samples in class c, ensuring that uncertainty is evaluated relative to the class-wise distribution. The next step is to compute the density score $D(x_i)$, which measures how representative a sample is within its local neighborhood in the embedding space. This score is calculated using cosine similarity between the embedding of the sample e_i and the embeddings of its k-nearest neighbors N_i. The density score is defined as:

$$D(x_i) = 1 - \frac{1}{|N_i|} \sum_{x_j \in N_i} \text{sim}(e_i, e_j) \tag{3}$$

where $\text{sim}(e_i, e_j)$ is the cosine similarity between the embeddings e_i and e_j, and $|N_i|$ is the number of nearest neighbors. The density score helps identify samples that are representative of their local regions in the feature space. The final CBUDR score combines both the class-normalized uncertainty and the density score. The combination is controlled by two hyperparameters, α and β, such that $\alpha + \beta = 1$:

$$\text{CBUDR}(x_i) = \alpha \cdot U_c(x_i) + \beta \cdot D(x_i) \tag{4}$$

By adjusting α and β, we can control the relative weight of each component, uncertainty and density, in the final score.

Once the CBUDR scores are computed, all samples are ranked in descending order of their scores. This ranking helps us identify the most informative samples, which can then be selected to form the coreset. For coreset selection, we choose the top-K% of samples based on their CBUDR scores. Additionally, adaptive or batch strategies can be used to dynamically select samples during training.

2.3 Entropy + CBUDR Combination

To further enhance the coreset selection, we combine entropy and CBUDR scores. This allows us to leverage both entropy-based uncertainty and density-based uncertainty for a more comprehensive ranking. The combined uncertainty score $U'(x_i)$ is calculated by linearly weighting the entropy score and the CBUDR score as follows:

$$U'(x_i) = \lambda \cdot U(x_i) + (1 - \lambda) \cdot \text{CBUDR}(x_i) \tag{5}$$

where $\lambda \in [0, 1]$ is a hyperparameter that controls the relative importance of the entropy and CBUDR scores. Finally, samples are ranked in descending order of the combined uncertainty score $U'(x_i)$. This ensures that the most informative samples, considering both entropy and density, are selected for the coreset.

3 Experimental Setup

The experiment was conducted on the Kaggle platform using an NVIDIA Tesla P100 GPU. The primary aim of the experiment was to evaluate model performance through various coreset selection strategies, which are designed to improve model efficiency while handling unreliable and biased data distributions.

3.1 Datasets

We evaluate our method on several benchmark spam detection datasets, chosen for their relevance to spam classification tasks across different domains. The **UTKML Twitter Spam** dataset consists of 11,968 tweets, with 5,815 labeled as spam and 6,153 labeled as ham. It offers real-world, social media data with noisy labeling. The **UCI SMS Spam Collection** includes 5,572 messages, 747 of which are labeled spam and 4,825 are labeled ham. This dataset is widely used in research and is known for its class imbalance. The **Ling-Spam** dataset comprises 2,893 email messages, 481 of which are spam and 2,412 are ham. This dataset presents a challenge for spam detection, using a smaller email corpus.

3.2 Preprocessing

The preprocessing steps are as follows: first, we tokenize the messages, convert them to lowercase, and remove punctuation and stopwords to reduce noise. Following this, we encode the messages into embeddings using Sentence-BERT (SBERT) Reimers and Gurevych [2019], a pretrained Transformer model for sentence embeddings. SBERT is used to generate contextual sentence embeddings, preserving the semantic meaning of the messages and enhancing model performance.

3.3 Coreset Selection

The coreset selection process consists of several key steps. First, we train a lightweight proxy classifier, Logistic Regression, to obtain class probabilities for all samples. These predictions serve as the basis for calculating uncertainty for each sample. Specifically, we compute two types of uncertainty: entropy and density uncertainty. Entropy measures the uncertainty in class prediction, while density uncertainty captures the sample's position in the data space.

Next, we rank the samples using three criteria: Entropy, Class-Balanced Uncertainty Density Ranking (CBUDR), and the combined Entropy-CBUDR scores. Entropy reflects the uncertainty of classification, while CBUDR balances uncertainty with the distribution of the classes within the data. The combined score uses both measures to ensure that selected samples are both uncertain and diverse in terms of class distribution. Based on these computed ranking scores, we select coresets using several strategies. First, in **Random Selection**, we randomly choose K% of the samples without considering any ranking criteria. In **Top-K Selection**, we choose the top K% of samples based on the highest combined ranking scores. **Bottom-K Selection** involves selecting the bottom K% of samples with the lowest ranking scores. For class balance, **Class-wise Top-K%** selects the top K% per class or per cluster, ensuring diversity in the coreset. Similarly, **Classwise Bottom-K%** dynamically selects the bottom K% per class or per cluster to maintain balance.

3.4 Model Training

For training, we fine-tune the BERT model on each selected coreset for a fixed number of epochs. We use 75% of the dataset for training, 15% for validation, and 15% for testing. Fine-tuning is conducted on each coreset to evaluate how performance scales when trained on smaller, strategically selected datasets. For comparison, we also fine-tune the BERT model on the full dataset as an upper-bound baseline.

3.5 Evaluation Metrics

The model performance is evaluated using standard classification metrics. These include accuracy (%), which measures the overall classification correctness, F1-score (%), which is the harmonic mean of precision and recall and is robust to class imbalance, precision (%), which is the fraction of correctly predicted spam samples, and recall (%), which measures the fraction of true positive spam samples correctly detected. These metrics are crucial for assessing the model's performance, especially in imbalanced datasets where class distribution significantly influences results.

3.6 Ablation Studies Setup

In the ablation studies, we evaluate the performance of different coreset selection strategies at various coreset sizes, specifically for $k = 5\%, 10\%, 25\%$. This analysis helps us understand how the size of the coreset impacts model performance. Additionally, we examine the effect of combining entropy and density uncertainty ranking versus using only entropy or only density uncertainty. This comparison enables us to assess how each selection strategy contributes to the robustness and efficiency of the model in different contexts.

4 Results and Discussion

In this section, we analyze the performance of different coreset selection strategies and ranking methods across the three benchmark datasets: UTKML Twitter Spam, UCI SMS Spam Collection, and Ling-Spam. We compare Entropy, CBUDR, and their combination under multiple coreset

Table 1: Ablation Study: Performance of Different Coreset Selection Strategies and Ranking Methods on UtkMl Twitter Spam Dataset

Coreset Strategy	Ranking Method	5%				10%				25%			
		Acc (%)	F1 (%)	Prec (%)	Rec (%)	Acc (%)	F1 (%)	Prec (%)	Rec (%)	Acc (%)	F1 (%)	Prec (%)	Rec (%)
Random		94.44	93.98	100.00	88.64	94.44	94.12	97.56	90.91	95.55	95.41	98.11	92.86
Top-K	Entropy	71.11	74.51	66.67	84.44	87.22	86.55	93.67	80.43	88.86	88.89	86.58	91.32
Top-K	CBUDR	67.78	60.27	70.97	52.38	85.56	85.56	85.56	85.56	89.31	88.84	90.95	86.82
Top-K	Entropy+CBUDR	78.89	73.24	96.30	59.09	83.33	84.69	80.58	89.25	91.76	91.90	88.98	95.02
Class-wise Top-K	Entropy	63.33	67.33	59.65	77.27	83.33	81.01	90.14	73.56	90.42	90.02	91.08	88.99
Class-wise Top-K	CBUDR	73.33	68.42	81.25	59.09	89.44	89.14	88.64	89.66	88.20	88.40	84.52	92.66
Class-wise Top-K	Entropy+CBUDR	78.89	76.54	83.78	70.45	82.22	78.67	93.65	67.82	89.76	89.55	88.74	90.37
Bottom-K	Entropy	97.78	98.11	96.30	100.00	98.33	98.52	99.01	98.04	98.22	98.33	97.51	99.16
Bottom-K	CBUDR	98.89	98.41	100.00	96.88	98.89	98.78	98.78	98.78	98.89	98.92	99.14	98.71
Bottom-K	Entropy+CBUDR	98.89	98.80	100.00	97.62	97.78	97.85	96.81	98.91	99.33	99.36	99.15	99.57
Class-wise Bottom-K	Entropy	98.89	98.85	100.00	97.73	98.33	98.27	98.84	97.70	98.44	98.38	99.07	97.71
Class-wise Bottom-K	CBUDR	100.00	100.00	100.00	100.00	99.44	99.42	100.00	98.85	99.33	99.31	100.00	98.62
Class-wise Bottom-K	Entropy+CBUDR	98.89	98.88	97.78	100.00	98.33	98.25	100.00	96.55	98.66	98.61	99.53	97.71
Baseline	All (100%)									96.49	96.41	95.92	96.91

strategies (Top-K, Bottom-K, Class-wise) and coreset sizes (5%, 10%, 25%). The discussion highlights dataset-specific behaviors, the impact of class imbalance, and insights into effective coreset design.

4.1 UTKML Twitter Spam (Balanced Dataset)

Table 1 presents the performance results for the UTKML Twitter Spam dataset, which is characterized by a balanced distribution of classes.

Bottom-K consistently outperforms Top-K: The Bottom-K strategy with entropy achieved an F1-score of 98.33% at 25%, which surpassed the baseline full-data performance of 96.41%. This highlights the advantage of reinforcing "easy" samples, leading to better generalization and improved spam detection.

Entropy vs. CBUDR: Entropy-based Top-K showed higher recall (84.44% at 5%) but lower precision (66.67%), indicating that it prioritized uncertain samples, often misclassifying ham as spam. In contrast, CBUDR provided more balanced trade-offs, with Top-K at 25% achieving precision of 90.95% and recall of 86.82%.

Combined ranking improves robustness: The combination of entropy and CBUDR achieved superior performance, with an F1-score of 91.90% at 25% coreset size, outperforming entropy (88.89%) or CBUDR (88.84%) alone. This suggests that combining uncertainty with density-based ranking improves model robustness.

Class-wise selection ensures balance: Ensuring balanced selection across spam and ham samples improved recall stability, although performance was slightly lower than the Bottom-K strategy.

Overall, UTKML demonstrates that carefully chosen coresets (especially Bottom-K) can surpass full-data baselines, even with a balanced dataset.

4.2 UCI SMS Spam Collection (Imbalanced Dataset)

Table 2 shows the results for the UCI SMS dataset, which is highly imbalanced with only 13% spam samples.

Entropy struggles under imbalance: With Top-K entropy at 5%, F1-score dropped to 72.73%, and recall was 80.00

CBUDR mitigates imbalance: By incorporating density, CBUDR maintained more balanced results. At 10%, CBUDR Top-K achieved recall of 90.91% and precision of 62.50%, outperforming entropy-based selection under imbalance.

Bottom-K dominates in imbalanced settings: Both Bottom-K entropy and CBUDR achieved perfect F1-scores (100.0%) at 5% and 10%, and maintained 100.0% precision and recall even at 25%, outperforming other strategies.

Class-wise selection stabilizes recall: Class-wise Top-K improved recall stability by ensuring that both ham and spam classes were represented, although performance was still lower than Bottom-K.

The UCI results reinforce that Bottom-K selection is particularly effective for imbalanced datasets, as it prevents overfitting to noisy majority-class samples and strengthens decision boundaries.

Table 2: Ablation Study (with handling imbalance): Performance of Different Coreset Selection Strategies and Ranking Methods on UCI Dataset

Coreset Strategy	Ranking Method	5%				10%				25%			
		Acc (%)	F1 (%)	Prec (%)	Rec (%)	Acc (%)	F1 (%)	Prec (%)	Rec (%)	Acc (%)	F1 (%)	Prec (%)	Rec (%)
Random	None	100.00	100.00	100.00	100.00	97.62	91.67	84.62	100.00	98.09	93.10	90.00	96.43
Top-K	Entropy	92.86	72.73	66.67	80.00	90.48	63.64	63.64	63.64	99.04	96.30	100.00	92.86
Top-K	CBUDR	90.48	33.33	100.00	20.00	91.67	74.07	62.50	90.91	97.61	90.91	92.59	89.29
Top-K	Combined	90.48	33.33	100.00	20.00	90.48	69.23	60.00	81.82	97.13	88.00	100.00	78.57
Class-wise Top-K	Entropy	90.48	60.00	60.00	60.00	90.48	63.64	63.64	63.64	99.04	96.30	100.00	92.86
Class-wise Top-K	CBUDR	90.48	33.33	100.00	20.00	91.67	74.07	62.50	90.91	97.61	90.91	92.59	89.29
Class-wise Top-K	Combined	90.48	33.33	100.00	20.00	90.48	69.23	60.00	81.82	97.13	88.00	100.00	78.57
Bottom-K	Entropy	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.52	98.18	100.00	96.43
Bottom-K	CBUDR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Bottom-K	Combined	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.52	98.18	100.00	96.43
Class-wise Bottom-K	Entropy	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.52	98.18	100.00	96.43
Class-wise Bottom-K	CBUDR	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Class-wise Bottom-K	Combined	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.52	98.18	100.00	96.43
Baseline	All (100%)									99.52	98.18	100.00	96.43

Table 3: Ablation Study (with handling imbalance): Performance of Different Coreset Selection Strategies and Ranking Methods on LingSpam Dataset

Coreset Strategy	Ranking Method	5%				10%				25%			
		Acc (%)	F1 (%)	Prec (%)	Rec (%)	Acc (%)	F1 (%)	Prec (%)	Rec (%)	Acc (%)	F1 (%)	Prec (%)	Rec (%)
Random	None	90.91	50.00	100.00	33.33	88.64	70.59	60.00	85.71	99.08	97.30	94.74	100.00
Top-K	Entropy	86.36	0.00	0.00	0.00	81.82	55.56	45.45	71.43	87.16	61.11	61.11	61.11
Top-K	CBUDR	90.91	50.00	100.00	33.33	79.55	40.00	37.50	42.86	93.58	78.79	86.67	72.22
Top-K	Combined	81.82	0.00	0.00	0.00	77.27	37.50	33.33	42.86	95.41	83.87	100.00	72.22
Class-wise Top-K	Entropy	86.36	0.00	0.00	0.00	81.82	55.56	45.45	71.43	87.16	61.11	61.11	61.11
Class-wise Top-K	CBUDR	90.91	50.00	100.00	33.33	79.55	40.00	37.50	42.86	93.58	78.79	86.67	72.22
Class-wise Top-K	Combined	81.82	0.00	0.00	0.00	77.27	37.50	33.33	42.86	95.41	83.87	100.00	72.22
Bottom-K	Entropy	100.00	100.00	100.00	100.00	97.73	93.33	87.50	100.00	100.00	100.00	100.00	100.00
Bottom-K	CBUDR	100.00	100.00	100.00	100.00	97.73	92.31	100.00	85.71	100.00	100.00	100.00	100.00
Bottom-K	Combined	100.00	100.00	100.00	100.00	97.73	92.31	100.00	85.71	100.00	100.00	100.00	100.00
Class-wise Bottom-K	Entropy	100.00	100.00	100.00	100.00	97.73	93.33	87.50	100.00	100.00	100.00	100.00	100.00
Class-wise Bottom-K	CBUDR	100.00	100.00	100.00	100.00	97.73	92.31	100.00	85.71	100.00	100.00	100.00	100.00
Class-wise Bottom-K	Combined	100.00	100.00	100.00	100.00	97.73	92.31	100.00	85.71	100.00	100.00	100.00	100.00
Baseline	All (100%)									99.54	98.61	98.61	98.61

4.3 Ling-Spam (Highly Imbalanced Dataset)

Table 3 presents the results for Ling-Spam, a dataset with severe imbalance. Key insights include:

Top-K fails at small coresets: Entropy-based Top-K at 5% collapsed entirely (F1=0.0), selecting primarily ham due to the skewed uncertainty distribution. Even combined ranking underperformed at low percentages.

CBUDR and larger coresets show improvement: At 25%, CBUDR Top-K achieved precision of 86.67% and recall of 72.22%, while combined ranking reached an F1-score of 83.87%. This demonstrates that density-based ranking mitigates imbalance effects as the coreset size increases.

Bottom-K maintains perfect scores: At all coreset sizes, Bottom-K entropy and CBUDR achieved near-perfect accuracy, precision, and recall, matching or exceeding the baseline performance (98.61% F1).

Class-wise strategies stabilize performance: Class-wise Bottom-K strategies maintained performance across coreset sizes while preventing class neglect. Ling-Spam illustrates the limitations of uncertainty-only methods under imbalance, while confirming that Bottom-K and CBUDR are effective strategies for overcoming this challenge.

Across all datasets, several consistent patterns emerge. Bottom-K strategies, especially when combined with CBUDR, consistently outperform other strategies, often surpassing the performance achieved with full-data training. This suggests that Bottom-K selection, particularly when density is considered, reinforces "easy" samples and improves model generalization. Entropy-based methods, on the other hand, prove insufficient under imbalanced conditions, as they tend to over-select ambiguous majority-class samples, leading to suboptimal performance, particularly for the minority class. CBUDR, by emphasizing representativeness and density, provides complementary benefits, resulting in more balanced and reliable performance. When combined, entropy and CBUDR offer enhanced robustness, particularly at larger coreset sizes, as they balance both uncertainty and representativeness in the selected samples. Finally, class-wise strategies ensure the inclusion of minority classes, which is particularly crucial under heavy class imbalance, preventing neglect of rare classes.

These findings suggest that carefully designed coreset selection strategies can significantly reduce dataset size, by up to 75%, while maintaining or even improving model performance compared to training on the full dataset. This highlights the potential for efficient and reliable machine learning in scenarios with unreliable or imbalanced data.

5 Conclusion

In this work, we presented a novel coreset selection strategy for efficient spam detection, leveraging the Entropy + Density Uncertainty Ranking scheme. By prioritizing highly informative and ambiguous samples, this approach enables Transformer-based models to achieve strong predictive performance while relying on significantly smaller subsets of the training data. Our method strategically identifies the most representative and uncertain samples, balancing the trade-off between model accuracy and data efficiency. Experimental results across benchmark spam datasets demonstrate that our approach can substantially reduce training costs, by up to 75%, without compromising accuracy, precision, or recall. This makes the method particularly suitable for real-world, resource-constrained environments, such as mobile devices or low-power communication systems, where computational and memory resources are limited.

Future work could extend our coreset selection framework by integrating advanced active learning techniques, such as query-by-committee or uncertainty sampling, to iteratively refine the coreset during training. Additionally, exploring adaptive sampling strategies or hybrid uncertainty measures that combine entropy, density, and model uncertainty could further enhance coreset efficiency. Expanding the application of this method to other domains, such as fraud detection, phishing prevention, and misinformation filtering, would validate the generalizability of our approach across different types of unreliable and imbalanced data, further solidifying its robustness and practical applicability. Moreover, investigating the use of our framework in adversarial settings could contribute to the growing need for reliable machine learning in the presence of distribution shifts or manipulated data.

References

Shafi'I Muhammad Abdulhamid, Muhammad Shafie Abd Latiff, Haruna Chiroma, Oluwafemi Osho, Gaddafi Abdul-Salaam, Adamu I Abubakar, and Tutut Herawan. A review on mobile sms spam filtering techniques. *IEEE Access*, 5:15650–15666, 2017.

Mohammed Rasol Al Saidat, Suleiman Y Yerima, and Khaled Shaalan. Advancements of sms spam detection: A comprehensive survey of nlp and ml techniques. *Procedia Computer Science*, 244:248–259, 2024.

Chengliang Chai, Jiayi Wang, Nan Tang, Ye Yuan, Jiabin Liu, Yuhao Deng, and Guoren Wang. Efficient coreset selection with cluster-based methods. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 167–178, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022.

Aisha Hamad Hassan and Tushar Shinde. Efficient spam detection with sentence-bert using adaptive uncertainty-diversity ranking coresets. In *Women in Machine Learning Workshop@ NeurIPS 2025*, 2025.

Xiaoxu Liu, Haoye Lu, and Amiya Nayak. A spam transformer model for sms spam detection. *IEEE Access*, 9: 80253–80263, 2021.

Dare Azeez Oyeyemi and Adebola K Ojo. Sms spam detection and classification to combat abuse in telephone networks using natural language processing. *arXiv preprint arXiv:2406.06578*, 2024.

Ankit Abhijit Pal, Sudin Mondal, C Ashok Kumar, and C Jothi Kumar. A transformer-based approach for fake news and spam detection in social media using roberta. In *2025 International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI)*, pages 1256–1263. IEEE, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Tushar Shinde. High-performance lightweight vision models for land cover classification with coresets and compression. In *TerraBytes-ICML 2025 workshop*.

Tushar Shinde and Manasa Madabhushi. Data-efficient and robust coreset selection via sparse adversarial perturbations. In *NeurIPS 2025 Workshop: Reliable ML from Unreliable Data*.

Tushar Shinde and Avinash Kumar Sharma. Scalable and efficient multi-weather classification for autonomous driving with coresets, pruning, and resolution scaling. In *ICLR 2025 Workshop on Machine Learning Multiscale Processes*.

Tushar Shinde, Avinash Kumar Sharma, Shivam Bhardwaj, and Ahmed Silima Vuai. Navigating coreset selection and model compression for efficient maritime image classification. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pages 1608–1616, 2025.

Tian Xia and Xuemin Chen. A discrete hidden markov model for sms spam detection. *Applied Sciences*, 10(14): 5011, 2020.

Xiaobo Xia, Jiale Liu, Shaokun Zhang, Qingyun Wu, Hongxin Wei, and Tongliang Liu. Refined coreset selection: Towards minimal coreset size under model performance constraints. *arXiv preprint arXiv:2311.08675*, 2023.

Haoran Zhang, Yong Liu, Yunzhong Qiu, Haixuan Liu, Zhongyi Pei, Jianmin Wang, and Mingsheng Long. Timesbert: A bert-style foundation model for time series understanding. *arXiv preprint arXiv:2502.21245*, 2025.