PSO-Weighted Ensemble of Bags-of-Word and N-Gram Classifiers for YouTube Spam Detection

Nor Azman Mat Ariff[1*], Mohd Zaki Mas'ud[2], Mohd Najwan Md Khambari[1], Mohd Fairuz Iskandar Othman[1], Taqwan Thamrin[3]

[1] Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Malaysia

[2] Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, Malaysia

[3] Fakultas Ilmu Komputer, Universitas Bandar Lampung, Indonesia

DOI: https://doi.org/10.47772/IJRISS.2025.91100545

Received: 09 December 2025; Accepted: 16 December 2025; Published: 23 December 2025

ABSTRACT

YouTube spam comments degrade user experience as well as increasing security and monetization risks, highlights the need for resilient automated detection system. The YouTube spam detection system has progressed from relying on single classifiers to incorporating ensemble-based system. However, current YouTube spam ensemble system typically train all base classifiers on homogeneous feature representations and rely on equal or fixed weighting schemes, which limits error diversity and prevents the ensemble from adapting to the varying strengths of individual models. This study proposed a Particle Swarm Optimization weighted ensemble that combined multiple n-gram and BoW classifiers to build spam detection models. Six single classifiers using 1-gram to 5-gram character features and BoW features were combined into ensemble configurations with equal weighting and PSO-optimized weighting, then evaluated on five YouTube spam datasets spanning Eminem, Katy Perry, LMFAO, Psy, and Shakira datasets. Results demonstrated that PSO-weighted ensembles consistently outperformed the best single classifier on every dataset, with improvements ranging from 1.0 to 1.5 percentage points and accuracies from 91.65% to 96.79%. The all n-grams plus BoW with PSO-optimized weights ensemble delivered robust performance across all datasets, with PSO gains over equal weighting of 0.2 to 0.7 percentage points. These findings confirmed that combining character n-gram and BoW features captured complementary spam patterns, and that PSO-based weighting provided an adaptive mechanism for classifier integration. The proposed approach offered a good, generalizable solution for automated spam detection across diverse YouTube comments and social media platforms without extensive manual tuning.

Keywords: Ensemble learning, N-gram features, Particle Swarm Optimization, Spam detection, Support Vector Machine

INTRODUCTION

Social media platforms, particularly YouTube, have become dominant ecosystems for entertainment, education and content creation and user engagement, facilitating billions of interactions daily across diverse communities and content domains. However, this exponential growth in user-generated content has simultaneously created an increasingly challenging environment for content moderation and community integrity. YouTube spam activities pose multiple threats, encompassing misleading comments, promotional links, phishing attempts, malware distribution, and fraudulent engagements, which collectively degrade user experience and create significant security risks (Chaudhary & Sureka, 2013; Tripathi et al., 2019; Sinhal & Maheshwari, 2022; Mohandas et al., 2024; Mukherjee et al., 2026). These spam manifestations represent a critical threat to platform sustainability and user satisfaction, necessitating the development of more sophisticated and adaptive detection mechanisms.

In response to these escalating threats, substantial research has been dedicated to the application of machine learning (ML) algorithms to mitigate the issue. Oh, (2022), Ansari et al., (2023), Shabadi et al., (2023), and Yadav et al., (2025) have collectively evaluated a broad spectrum of algorithms, including Support Vector Machine (SVM), Logistic Regression, Decision Tree, Random Forest, Naive Bayes, and K-Nearest Neighbors (KNN). Among the high-performing models, Yadav et al., (2025) highlighted the superior efficacy of the Support Vector Classifier (SVC) utilizing linear and Gaussian kernels. Similarly, R. Abinaya, (2020) demonstrated the strong reliability of Logistic Regression. Random Forest is also frequently cited as a robust classifier, often yielding performance metrics comparable to the leading models. While simpler in architecture, Naive Bayes was observed to be effective, particularly in contexts prioritizing computational efficiency. Conversely, Ansari et al., (2023) found the K-Nearest Neighbors (KNN) classifier to be the least effective, resulting in its exclusion from further experimental phases in their study.

While traditional classifiers have established a baseline for detection, recent studies has shifted toward advanced Deep Learning (DL) architectures to capture more intricate patterns. Sinhal et al., (2024) proposed two DL models, Feedforward Neural Networks(FNN) and Recurrent Neural Networks(RNN), both of which surpassed conventional machine-learning baselines for YouTube spam detection, with the RNN achieving the strongest overall performance. Similarly, Hasan et al. (2025) conducted a systematic comparison of deep learning and traditional machine-learning approaches, and reported that the RoBERTa-based deep learning model outperformed all evaluated ML baselines. In contrast, Yanto et al., (2025) compared classical ML models (Logistic Regression, Random Forest, SVM, FastText) with deep LSTM and GRU models on a balanced YouTube spam dataset and found that FastText, a non-DL classifier, achieved the highest accuracy (96.0%) while the best DL model (GRU) reached only 91.6% accuracy, clearly trailing the ML baselines. However, deep learning approaches are often criticized for their "black box" nature, which is hard to understand the decision making process. Furthermore, their high computational cost and implementation complexity often render them impractical for real-time deployment in resource-constrained environments.

Other than single classifier ML and DL, researchers also explored ensemble methods to improve robustness. however, current approaches suffer from a fundamental design limitation. Existing ensemble frameworks proposed by Oh, (2022), Sinhal & Maheshwari, (2022), Ansari et al., (2023) combine heterogeneous classifiers, such as SVMs, Decision Trees, and Gradient Boosting, but train all base models on identical word-level features (TF-IDF or Bag-of-Words). This configuration inherently limits diversity of errors: because all models see exactly the same input representation, they tend to make wrong predictions on the same samples and in similar ways, so the ensemble can add little over the best single model and may even underperform it when weaker but correlated members outvote a stronger classifier. Furthermore, these conventional ensembles typically assign equal or fixed weights to all constituent classifiers, neglecting the heterogeneous contributions of different models and leaving substantial performance gains unexploited.

To overcome the critical limitation of feature homogeneity and equal or fixed weighting, the primary objective of this research is to develop a robust YouTube spam detection ensemble that systematically integrates multi-granularity feature representations and adaptive weight optimization. Specifically, the study aims to: (1) construct a heterogeneous ensemble comprising character-level n-gram classifiers (1-gram to 5-gram) and Bags-of-Word (BoW) models to capture orthographic spam artifacts and semantic cues simultaneously; (2) implement optimization algorithm specifically Particle Swarm Optimization (PSO) to learn dataset-specific weight vectors that dynamically calibrate the contribution of each base classifier; and (3) evaluate the proposed framework across multiple YouTube spam datasets to demonstrate good generalizability and resilience against adversarial obfuscation compared to single classifiers and equal weight ensemble methods. This study introduces a novel hybrid framework combining character-level n-grams and BoW features to detect obfuscated spam tactics often missed by single classifier models. It is expected that employing Particle Swarm Optimization for adaptive weighting will generate a robust ensemble that outperforms single classifiers and equal weight ensemble in detection accuracy.

RELATED WORK

Feature Representation

Feature representation is crucial for the success of any machine learning-based system. This is because different feature representations extract different aspects of the data. Based on previous research, YouTube spam feature representations could be categorized into three types: content-based, metadata-based, and automatic features generated by deep learning.

Given that YouTube spam datasets consist of text-based comments, the main feature representation in most studies is content-based and they are constructed using text classification methods. This involves several critical preprocessing phases in text classification, including tokenization, which splits the text into individual words or tokens to facilitate text analysis; converting all text to a consistent letter case (typically upper or lower case) to reduce complexity; removal of special characters and stop words to reduce data dimensionality; and stemming or word truncation aimed at reducing the vocabulary size. These steps are applied in most of YouTube spam studies. Among the content-based feature representations that utilize text representation are bags-of-words, n-grams, and count vectorization as demonstrated in the studies by Aiyar & Shetty, (2018), Ansari et al., (2023), Shabadi et al., (2023), Hasan et al., (2025), and Yanto et al., (2025). Apart from text representations, linguistic features as studied by Chaudhary & Sureka, (2013), Tripathi et al., (2019), Ansari et al., (2023) and Yanto et al., (2025), as well as sentiment and semantic features as investigated in Chaudhary & Sureka, (2013), and Ansari et al., (2023), employ content-based features. Content-based features can effectively identify promotional keywords and detect obfuscation techniques that employ character substitution such as "fr33".

Metadata-based feature representation has received attention, with several studies demonstrating its effectiveness (Chaudhary & Sureka, 2013; Tripathi et al., 2019;Shabadi et al., 2023). These features extract behavioral signals from users rather than from the comment content itself. In contrast to content-based approaches that analyze what users comment on, metadata-based features analyze who commented, when they commented, and how the community responded. Study in Chaudhary & Sureka, (2013) achieved classification accuracy exceeding 80% using exclusively metadata features such as subscriber-to-view ratios, video duration patterns, and temporal upload sequences without analyzing any textual comments. Study by Oh, (2022) improved accuracy from 90% using text features alone to 95% by combining text and metadata features. This demonstrates that behavioral signals possess sufficient discriminative power to assist classifiers in making decisions.

Deep Learning have revolutionized feature representation by eliminating the need for manual feature engineering. Rather than depending on domain expertise, deep learning automatically discovers feature representations through end-to-end learning. Hasan et al., (2025) utilized automatically generated feature representations derived from BERT, RoBERTa, and DistilBERT. These three transfer learning algorithms were compared with six machine learning algorithms. RoBERTa achieved the best performance with 92.71% accuracy. Similarly, Yanto et al., (2025) compared deep learning models which are LSTM and GRU with classical machine learning models including Logistic Regression, Random Forest, and Support Vector Machine. However, the results found that machine learning models outperformed deep learning models.

Ensemble Method

Single-classifier approaches often struggle to identify spam accurately. Ensemble methods can enhance classification capabilities by leveraging multiple classifiers, compensating for the shortcomings of individual ones, and ultimately improving the overall performance of the classification system (Ostvar & Eftekhari Moghadam, 2020; Lin et al., 2021). Ensemble methods are utilized across various fields such as computer and network security (Lin et al., 2021; Oluwole Ogini et al., 2022), medicine (Zhang et al., 2019; Beeman et al., 2021; Bose et al., 2021) and business (Lahmiri et al., 2020; Titiani & Riana, 2022). By leveraging ensemble methods, each classifier analyzes different features of spam, thereby obtaining a more comprehensive and robust assessment. The three main classes of ensemble methods are bagging, boosting, and stacking. Bagging combines multiple classifiers trained separately in parallel. The final classification is obtained by combining the outputs of individual classifiers, typically using combination method such as majority voting, product rule, or mean rule. Boosting combines multiple weak classifiers into a strong classifier iteratively. Unlike bagging, each classifier attempts to correct the classification of the previous classifier by emphasizing samples that were misclassified. Stacking, known as stacked generalization, combines various base classifiers trained on the complete training set, and then a meta-classifier is trained on the output of the base classifiers. Here, the outputs of the original classifiers serve as features for the meta-classifier. Figure 1 shows how multiple classifiers are combined using bagging architecture.

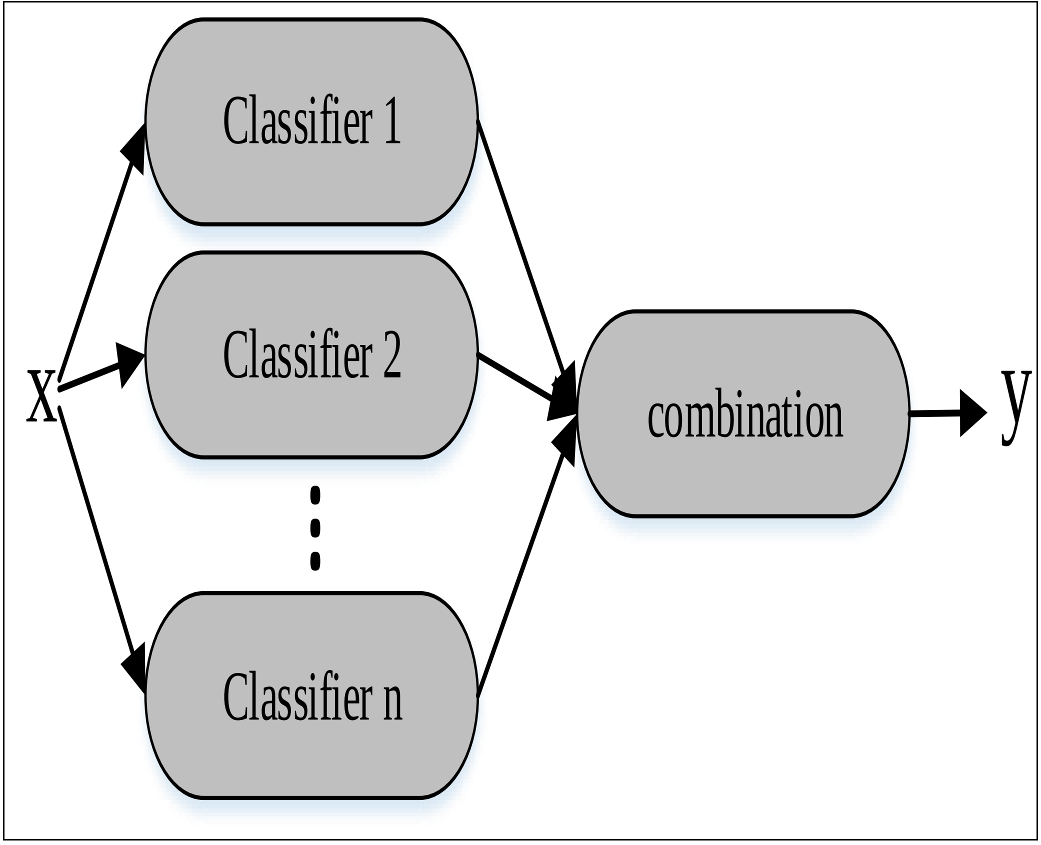

Figure 1 : Bagging Architecture

Previous studies demonstrate significant variation in weighted ensemble schemes applied to YouTube spam detection. Oh, (2022) compared hard voting (equal weights, majority rule) with soft voting (equal weights, probability averaging) and found that soft voting performed better because it incorporates classifier confidence into the final decision. In contrast, Chaudhary & Sureka, (2013) used a manual iterative procedure to adjust feature weights in a one-class classifier, assigning lower weights to more important, highly discriminative features. However, both approaches, whether based on fixed probability weights or manual iterative adjustment, rely on predetermined or manually-determined weight schemes that may not be optimal for diverse datasets and classifier combinations. Particle Swarm Optimization (PSO) provides another ensemble weighting scheme and is widely used. PSO dynamically adjusts the weights of the ensemble components, which allows it to adapt better to the data and improve prediction accuracy. This dynamic adjustment is more effective than static soft voting, which assigns equal or fixed weights to all classifiers (You et al., 2020; Ibrahim et al., 2025).

METHODOLOGY

The proposed PSO-weighted ensemble approach for YouTube spam detection has six stages namely data pre-processing, feature extraction, feature selection, base classifier training, weight optimization, and ensemble prediction. Figure 2 shows the block diagram of the proposed approach.

The data pre-processing stage involves four operations, namely removal of special characters, case normalization, stop word removal, and stemming. Removing special characters such as “/”, “>”, “ï”, “»”, “¿” and others is important because they introduce noise without any semantic value. The case normalization operation converts all text to lowercase to reduce the dimensionality of the feature vectors being developed, since high-dimensional feature vectors tend to be prone to overfitting. Stop word removal filters words such as "the," "is," and "a" that provide minimal discriminatory information for spam classification. Last, stemming reduces words to their root forms, consolidating variations of the same word such as "running", "runs", and "ran" all reduce to "run". These preprocessing operations are essential for reducing feature space complexity, improving computational efficiency, and enabling classifiers to focus on semantically meaningful patterns that distinguish spam from legitimate comments.

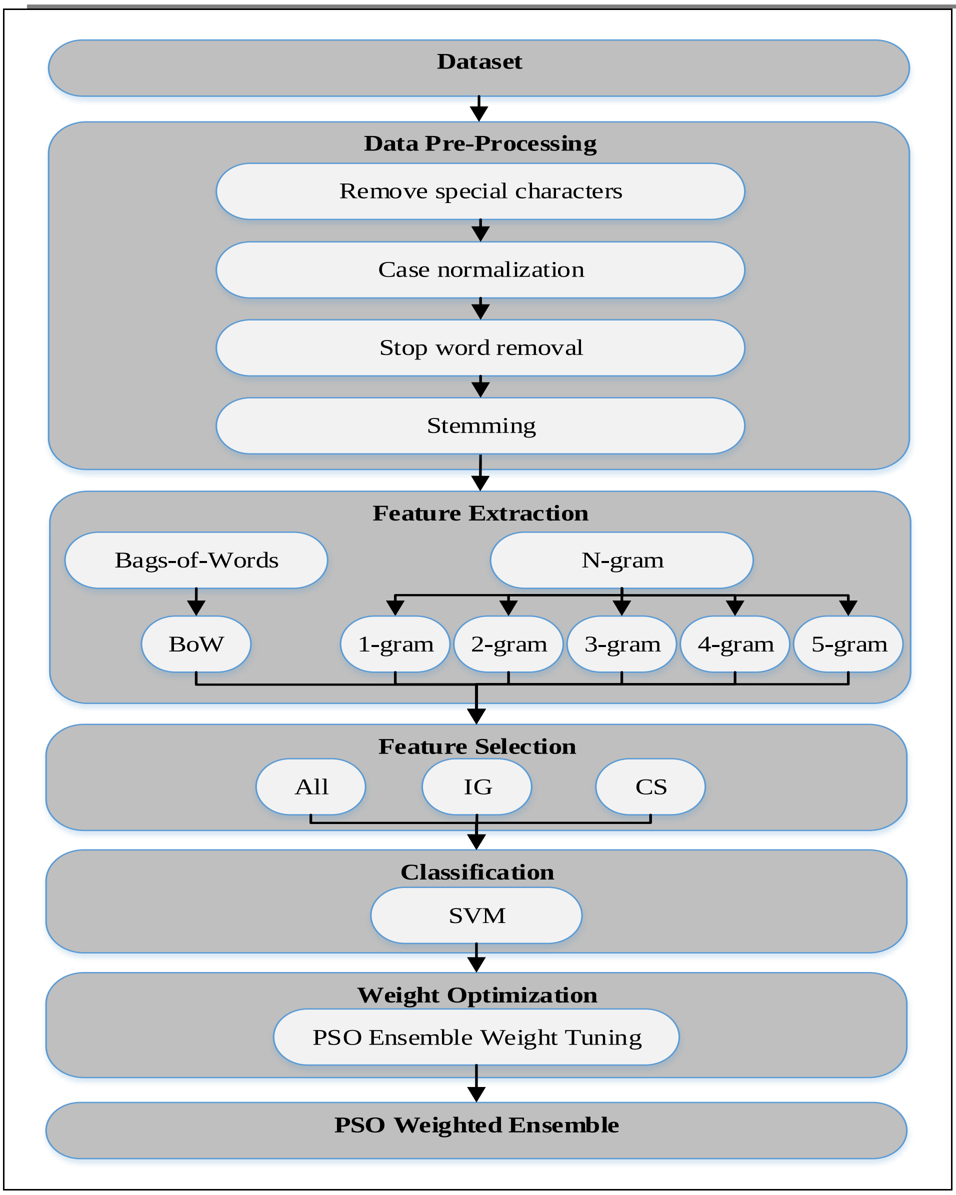

Figure 2 : PSO-Weighted Ensemble Framework for YouTube Spam Classification

Feature extraction stage transforms preprocessed text into numerical representations using two methods that capture different linguistic characteristics. BoW represents text as an unordered collection of word frequencies, where each unique word becomes a feature descriptor and elements represent word occurrence counts. This representation effectively capturing lexical information relevant to spam detection such as frequency of promotional keywords. N-gram extraction captures sequential word patterns by extracting contiguous sequences of n characters, from 1-gram to 5-gram. These N-gram features capture contextual and patterns that BoW misses. The combination of BoW and five N-gram variants produce six initial models that collectively encode diverse linguistic properties of comments.

Feature selection reduces dimensionality and improves classifier generalization by removing redundant and irrelevant features. Two feature selection techniques are applied namely Information Gain (IG) and Chi-Square (CS). Refer to Figure 2, “All” means that all BoW or N-gram features retains the complete feature set. Since each of the six extraction methods is paired with three selection strategies (All, IG, CS), this stage produces 18 distinct feature vector configurations (6 × 3 combinations) for the classification. This comprehensive exploration of feature engineering variations ensures that the methodology evaluates diverse combinations of feature representations and selection criteria, maximizing the likelihood of discovering optimal feature spaces for accurate spam detection.

Base classifiers are trained independently on each of the 18 feature vector configurations to generate diversity in the ensemble. Support Vector Machine (SVM) with Linear Kernel serves as the base classifier, selected for its proven effectiveness in high-dimensional feature spaces. The regularization parameter C is tuned via grid search to control the trade-off between maximizing the margin and penalizing misclassifications in a systematic manner. Each SVM classifier generates probability outputs for both classes, which serve as inputs to the weight optimization stage. This approach creates classifier diversity across two dimensions: (1) feature space diversity through different extraction methods and selection strategies, and (2) training diversity through independent model learning, both contributing to ensemble robustness and reducing overfitting risk.

Weight optimization using PSO will discovers optimal classifier weights that reflect actual classifier performance rather than assuming equal weight. Each particle representing a candidate weight vector $w = [w_1, w_2, w_3, \dots\ w_n]$ constrained to $\sum_{j=1}^{n} w_j$=100. At each iteration, particles update velocity and position according to personal best and global best solutions found by the swarm, balancing exploration of new weight combinations with exploitation of promising regions. The fitness function is accuracy on the training set, ensuring weights maximize classification performance on unseen data. The algorithm runs with 20 particles over 200 iterations. The final PSO-weighted ensemble produces predictions as $P_{ensemble}(spam) = \sum_{j=1}^{n} w_j \cdot P_j^{(spam)}$, classifying comments as spam if this probability exceeds 0.5. This approach addresses critical limitations of fixed probability-weighted voting where it discovers weights based on training set accuracy rather than assumption of equal weight and adapts weights to dataset characteristics without manual tuning.

RESULTS AND DISCUSSION

Dataset

The experimental dataset comprises YouTube comments publicly available from the UCI Machine Learning Repository. The dataset contains comments that were extracted from videos by five popular artists: Eminem, Katy Perry, LMFAO, Psy, and Shakira. The dataset contains 1,956 total comments distributed across the five music videos, with the following composition: Eminem (448 comments, 245 spam and 203 non-Spam), Katy Perry (350 comments, 175 spam and 175 non-Spam), LMFAO (438 comments, 236 spam and 202 non-Spam), Psy (350 comments, 175 spam and 175 non-Spam), and Shakira (370 comments, 174 spam and 196 non-Spam). There were 4 columns namely comment_id, author, date, and content,and pre-assigned binary labels indicating spam or non-Spam. In this study, only the textual content from the content field was utilized for analysis.

Experimental Setup

The experimental setup comprises repeated stratified random sub-sampling across 100 iterations with 80:20 training-testing split, maintaining 50% spam and 50% non-spam balance. Although the original datasets for Eminem, LMFAO, and Shakira exhibit class imbalance between spam and non-spam comments, stratified sampling ensures balanced 50% spam and 50% non-spam distribution in both training and testing sets. Preprocessed comments are transformed using Term Frequency (TF) weighting combined with six feature extraction methods (BoW, 1-gram through 5-gram), producing 18 feature configurations after applying IG and CS feature selection with 80% threshold criteria. Support Vector Machine with linear kernel serves as the base classifier, with regularization parameter C optimized via grid search and five-fold cross-validation. Particle Swarm Optimization executes 200 iterations as stopping criteria with 20 particles to discover optimal ensemble weights. Fitness is computed as accuracy. Final ensemble predictions aggregate SVM probability outputs using PSO-optimized weights. Performance is evaluated using accuracy across all 100 iteration.

Results

The experimental results across the five YouTube spam datasets reveal consistent patterns in both single-classifier performance and ensemble effectiveness. This section discusses the findings for each dataset individually before synthesizing the cross-dataset trends and their implications for robust spam detection.

Table 1 summarizes the classification performance of both single and ensemble models on the Eminem dataset. The first block reports six single classifiers built on 1-gram to 5-gram character features and BoW features, evaluated under three feature configurations. "All" indicates that all extracted features were used without feature selection. "IG" denotes feature selection using Information Gain and "CS" indicates feature selection using Chi-square. Across these single models, the performance differences between All, IG, and CS configurations are minimal for most classifiers, typically within 0.2 percentage points. The stability of results across All, IG, and CS configurations indicates that feature selection does not substantially alter classifier performance. Consequently, “All” features are used in the ensemble experiments. The second block presents six ensemble configurations that combine the baseline single classifiers. Three ensembles use equal weights. "All n-grams, equal weights" averages predictions from the five n-gram models. "All n-grams + BoW, equal weights" includes the BoW model alongside the n-gram models. "Best n-grams + BoW, equal weights" combines only the highest-performing n-gram model with the BoW model. The remaining three ensembles use Particle Swarm Optimization (PSO) to learn optimal weights for the same three combinations, resulting in "All n-grams, PSO-optimized weights," "All n-grams + BoW, PSO-optimized weights," and "Best n-grams + BoW, PSO-optimized weights.". Among the single classifiers, the 2-gram model achieves the highest accuracy of 92.50% using all features (indicated in bold), with IG yielding a marginal improvement to 92.59%. Equal-weight ensemble methods consistently surpass the best single model, with the "all n-grams" configuration reaching 93.52% and the "all n-grams + BoW" variant attaining 93.55%. PSO-optimized weighting further enhances accuracy, with the "all n-grams + BoW, PSO-optimized weights" ensemble delivering the best overall result of 93.84% (highlighted in bold), representing a 1.34% gain over the strongest single classifier.

Table 1 : Classification accuracy (%) of single and ensemble classifiers on the Eminem dataset

Model type	Model/feature setup	All	IG	CS
Single	1-gram	87.45	87.30	87.45
Single	2-gram	**92.50**	92.59	92.48
Single	3-gram	91.93	91.80	92.00
Single	4-gram	90.84	90.91	90.84
Single	5-gram	88.52	88.70	88.52
Single	BoW	89.41	89.39	89.39
Ensemble	All n-grams, equal weights	93.52		
Ensemble	All n-grams + BoW, equal weights	93.55		
Ensemble	Best n-grams + BoW, equal weights	93.14		
Ensemble	All n-grams, PSO-optimized weights	93.77		
Ensemble	All n-grams + BoW, PSO-optimized weights	**93.84**		
Ensemble	Best n-grams + BoW, PSO-optimized weights	93.27		

Note: PSO-optimized weights: All n-grams (w1=41, w2=84, w3=54, w4=57, w5=5), All n-grams + BoW (w1=70, w2=78, w3=66, w4=63, w5=7, & w6=75), Best n-gram + BoW (w1=48, & w2=34).

Table 2 presents results for the Katy Perry dataset, where the 3-gram classifier is the top-performing single model at 90.50% (highlighted in bold), whereas the BoW model lags substantially at 79.50%, suggesting that local character patterns dominate over BoW features for this dataset. Equal-weight ensembles improve upon the best single classifier, with the "best n-gram + BoW" ensemble reaching 90.94%. The PSO-weighted "all n-grams + BoW" ensemble achieves the highest accuracy of 91.65% (indicated in bold), demonstrating that adaptive weighting can effectively integrate contributions from all base classifiers even when one model is clearly dominant.

Table 2. Classification accuracy (%) of single and ensemble classifiers on the Katy Perry dataset

Model type	Model/feature setup	All	IG	CS
Single	1-gram	86.50	85.09	86.50
Single	2-gram	88.44	88.94	88.44
Single	3-gram	**90.50**	90.28	90.50
Single	4-gram	89.75	89.66	89.75
Single	5-gram	87.82	87.72	87.82
Single	BoW	79.50	79.31	79.50
Ensemble	All n-grams, equal weights	90.07		
Ensemble	All n-grams + BoW, equal weights	90.88		
Ensemble	Best n-grams + BoW, equal weights	90.94		
Ensemble	All n-grams, PSO-optimized weights	90.82		
Ensemble	All n-grams + BoW, PSO-optimized weights	**91.65**		
Ensemble	Best n-grams + BoW, PSO-optimized weights	91.37		

Note: PSO-optimized weights: All n-grams (w1=9, w2=14, w3=66, w4=0, w5=43), All n-grams + BoW (w1=32, w2=19, w3=10, w4=69, w5=50, & w6=96), Best n-gram + BoW (w1=53, & w2=27).

Table 3 reports results for the LMFAO dataset, which exhibits generally high single-classifier accuracies, with the 3-gram model attaining 96.50% (indicated in bold) and the 2-gram, 4-gram, and 5-gram models all exceeding 95%. Despite these strong baselines, ensemble methods still yield incremental improvements.The equal-weight "all n-grams + BoW" ensemble reaches 96.58%, and the PSO-weighted variant achieves 96.79% (highlighted in bold), the best result on this dataset. These findings indicate that even when individual models approach saturation, carefully weighted ensembles can extract complementary information across different feature granularities.

Table 3. Classification accuracy (%) of single and ensemble classifiers on the LMFAO dataset

Model type	Model/feature setup	All	IG	CS
Single	1-gram	87.61	88.20	87.61

Single	2-gram	95.19	95.28	95.20
Single	3-gram	**96.50**	96.59	96.50
Single	4-gram	96.09	96.00	96.09
Single	5-gram	95.60	95.63	95.61
Single	BoW	94.34	94.35	94.34
Ensemble	All n-grams, equal weights	96.35		
Ensemble	All n-grams + BoW, equal weights	96.58		
Ensemble	Best n-grams + BoW, equal weights	96.35		
Ensemble	All n-grams, PSO-optimized weights	96.53		
Ensemble	All n-grams + BoW, PSO-optimized weights	**96.79**		
Ensemble	Best n-grams + BoW, PSO-optimized weights	96.76		

Note: PSO-optimized weights: All n-grams (w1=15, w2=0, w3=85, w4=7, w5=0), All n-grams + BoW (w1=20, w2=75, w3=37, w4=20, w5=33, & w6=6), Best n-gram + BoW (w1=90, & w2=25).

Table 4 presents results for the Psy dataset. In contrast to other datasets, the BoW model is the strongest single classifier at 93.43% (indicated in bold), narrowly outperforming the 3-gram model at 93.19%. Equal-weight ensembles of "all n-grams" and "all n-grams + BoW" further enhance performance to 93.78% and 93.90%, respectively. PSO-optimized ensembles deliver the highest accuracy of 94.03% (highlighted in bold) for both the "all n-grams" and "all n-grams + BoW" configurations, indicating that optimized weights can effectively balance character-level and Bags-of-Word evidence in this dataset.

Table 4. Classification accuracy (%) of single and ensemble classifiers on the Psy dataset

Model type	Model/feature setup	All	IG	CS
Single	1-gram	83.66	82.71	83.63
Single	2-gram	92.40	92.96	92.40
Single	3-gram	93.19	93.19	93.19
Single	4-gram	92.47	92.43	92.49
Single	5-gram	91.54	91.63	91.54
Single	BoW	**93.43**	93.32	93.43
Ensemble	All n-grams, equal weights	93.78		
Ensemble	All n-grams + BoW, equal weights	93.90		

Ensemble	Best n-grams + BoW, equal weights	93.49		
Ensemble	All n-grams, PSO-optimized weights	**94.03**		
Ensemble	All n-grams + BoW, PSO-optimized weights	**94.03**		
Ensemble	Best n-grams + BoW, PSO-optimized weights	93.51		

Note: PSO-optimized weights: All n-grams (w1=29, w2=98, w3=10, w4=84, w5=26), All n-grams + BoW (w1=56, w2=83, w3=31, w4=18, w5=71, & w6=95), Best n-gram + BoW (w1=92, & w2=70).

Table 5 reports results for the Shakira dataset, where the 2-gram classifier emerges as the best single model at 92.71% (indicated in bold), with higher-order n-grams and the BoW model achieving slightly lower accuracies. Equal-weight ensembles of "all n-grams" reach 93.37%, and PSO-optimized weighting increases this to 93.60% (highlighted in bold), the top result for this dataset. Notably, the "all n-grams, PSO-optimized weights" ensemble outperforms configurations that include the BoW classifier, suggesting that the contribution of Bags-of-Word features is dataset dependent and that PSO effectively down-weights less informative models.

Table 5. Classification accuracy (%) of single and ensemble classifiers on the Psy dataset

Model type	Model/feature setup	All	IG	CS
Single	1-gram	86.38	86.50	86.37
Single	2-gram	**92.71**	92.82	92.71
Single	3-gram	92.26	92.53	92.25
Single	4-gram	91.13	91.25	91.13
Single	5-gram	89.38	89.56	89.38
Single	BoW	89.19	89.15	89.19
Ensemble	All n-grams, equal weights	93.37		
Ensemble	All n-grams + BoW, equal weights	93.27		
Ensemble	Best n-grams + BoW, equal weights	92.97		
Ensemble	All n-grams, PSO-optimized weights	**93.60**		
Ensemble	All n-grams + BoW, PSO-optimized weights	93.50		
Ensemble	Best n-grams + BoW, PSO-optimized weights	93.16		

Note: PSO-optimized weights: All n-grams (w1=18, w2=16, w3=77, w4=2, w5=17), All n-grams + BoW (w1=68, w2=92, w3=87, w4=28, w5=28, & w6=40), Best n-gram + BoW (w1=60, & w2=79).

Figure 3 visualizes these results across all five datasets, arranged as subfigures (a) through (e). Each subfigure displays the performance of single classifiers alongside ensemble configurations using equal weights and

PSO-optimized weights. The visualizations confirm the quantitative findings. Ensemble bars (E1 to E6) consistently exceed the tallest single-classifier bar in every dataset, and the PSO-optimized ensembles (E4 to E6) are always among the highest. For the Eminem dataset (Figure 1a), accuracy rises steadily from 1-gram to the 2-gram peak, then declines for higher-order n-grams, with PSO-optimized "all n-grams + BoW" visibly exceeding all singles at 93.84%. Figure 1b (Katy Perry) shows the BoW model performing substantially worse than n-gram models, with the 3-gram bar tallest among singles and PSO-optimized ensembles clearly separating at 91.65%. The LMFAO dataset (Figure 1c) demonstrates uniformly high performance across most models, with only slight visible gains from ensembles, peaking at 96.79% for the PSO-optimized variant. In Figure 1d (Psy), the BoW model is notably competitive, and PSO-optimized configurations clearly separate at 94.03%. Figure 1e (Shakira) shows the 2-gram model as the tallest single-classifier bar, with PSO-optimized "all n-grams" reaching the highest point at 93.60%, illustrating how PSO identifies when higher-order n-grams and word features contribute little value.

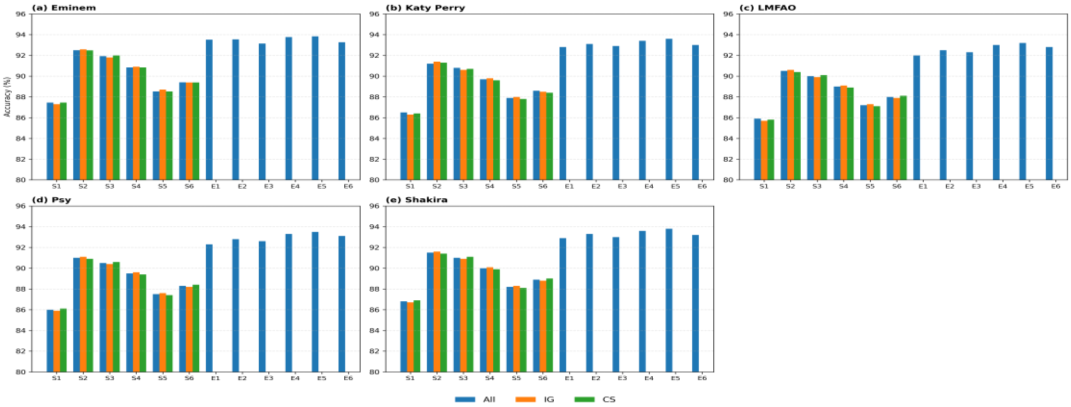

Figure 3 : Comparison of classification accuracy (%) for single classifiers (S1–S6: 1-gram, 2-gram, 3-gram, 4-gram, 5-gram, BoW) and ensemble classifiers (E1–E6: All n-grams – equal, All n-grams + BoW – equal, Best n-gram + BoW – equal, All n-grams – PSO, All n-grams + BoW – PSO, Best n-gram + BoW – PSO) across five YouTube spam datasets: (a) Eminem, (b) Katy Perry, (c) LMFAO, (d) Psy, and (e) Shakira. IG and CS values are shown only for single classifiers; ensemble rows display All accuracy.

DISCUSSION

Several consistent patterns emerge across the five YouTube datasets. First, ensemble methods systematically outperform the best single classifier for every dataset, with improvements ranging from approximately 1 to 1.5 percentage points. This confirms that combining multiple character n gram levels and BoW features captures complementary patterns in spam messages that no single feature representation can fully exploit. Second, while the optimal single model varies by dataset, with classifiers using 2 gram, 3 gram, or BoW features each proving strongest for different datasets, the PSO weighted ensemble that combines all n gram and BoW features is either the best or highly competitive on all datasets. This consistency highlights the advantage of retaining all feature granularities and allowing the optimization algorithm to assign dataset specific importance weights. Third, the gains from PSO over equal weighting are modest but stable, typically in the range of 0.2 to 0.7 percentage points. This suggests that PSO primarily fine tunes the contribution of each base classifier rather than inducing large structural changes in ensemble behavior.

The good performance of ensembles can be attributed to the complementary nature of character n-grams and BoW representations. Character n-grams capture local patterns such as obfuscation techniques, repeated characters, and morphological variations commonly used in spam, while BoW features encode semantic cues and whole-word repetitions. By combining these representations, the ensemble is better equipped to detect spam messages that employ diverse evasion strategies. The PSO-based weighting scheme extends this advantage by learning dataset-specific weights, automatically emphasizing the most informative base classifiers for each YouTube dataset without manual tuning. This adaptability is particularly valuable in practical applications, where spam characteristics can vary significantly across different social media platforms and content creators.

Furthermore, the consistent gains observed across all five datasets, despite their varying characteristics and baseline accuracies, suggest that the proposed ensemble approach is robust and generalizable. The PSO optimization, while computationally more intensive than simple averaging, remains tractable for real-world applications and provides a principled mechanism for integrating multiple information sources.

In summary, the experimental results demonstrate that PSO-weighted ensembles of n-gram and BoW classifiers offer a robust and effective solution for YouTube spam detection, consistently outperforming single models and equal-weight ensembles across all tested datasets. The findings underscore the importance of feature diversity and adaptive weighting in building accurate and generalizable spam detection systems.

CONCLUSION

This study proposes a PSO-weighted ensemble combining character n-grams (1-5) and BoW features for YouTube spam detection. Evaluated across five datasets, PSO-optimized ensembles consistently outperformed single classifiers by 1.0–1.5 percentage points, achieving 91.65%–96.79% accuracy. Complementary n-gram and BoW representations capture diverse spam patterns; PSO-learned weights adapt to dataset characteristics without manual tuning. Results demonstrate that adaptive ensemble weighting provides good classification performance and generalizability compared to equal-weight baselines, establishing an effective, practical approach for automated spam detection across social media platforms.

REFERENCES

1. Aiyar, S., & Shetty, N. P. (2018). N-Gram Assisted Youtube Spam Comment Detection. Procedia Computer Science, 132(Iccids), 174–182. https://doi.org/10.1016/j.procs.2018.05.181
2. Ansari, M. A., Prajapati, P., Dhotre, S., Kumar, S., & Chaudhari, S. (2023). Ensemble Learning based Efficient Spam Detection of YouTube Comments. 2023 6th International Conference on Advances in Science and Technology (ICAST), 448–453. https://doi.org/10.1109/ICAST59062.2023.10454921
3. Beeman, S. P., Morrison, A. M., Unnasch, T. R., & Unnasch, R. S. (2021). Ensemble ecological niche modeling of West Nile virus probability in Florida. PLOS ONE, 16(10), e0256868. https://doi.org/10.1371/journal.pone.0256868
4. Bose, S., Das, C., Banerjee, A., Ghosh, K., Chattopadhyay, M., Chattopadhyay, S., & Barik, A. (2021). An ensemble machine learning model based on multiple filtering and supervised attribute clustering algorithm for classifying cancer samples. PeerJ Computer Science, 7, e671. https://doi.org/10.7717/peerj-cs.671
5. Chaudhary, V., & Sureka, A. (2013). Contextual feature based one-class classifier approach for detecting video response spam on YouTube. 2013 Eleventh Annual Conference on Privacy, Security and Trust, 195–204. https://doi.org/10.1109/PST.2013.6596054
6. Hasan, M. N., Islam, M. M., Azim, R., & Biswas, J. (2025). YouTube Spam Comment Detection using Transfer Learning and Machine Learning algorithms. 2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), February, 1–6. https://doi.org/10.1109/ECCE64574.2025.11013288
7. Ibrahim, F., Mansour, K., Nasayreh, A., Samara, G., Bashkami, A., Smerat, A., & Nahar, K. M. O. (2025). Computer Methods and Programs in Biomedicine Update Optimized soft-voting CNN ensemble using particle swarm optimization for endometrial cancer histopathology classification. Computer Methods and Programs in Biomedicine Update, 8(August), 100217. https://doi.org/10.1016/j.cmpbup.2025.100217
8. Lahmiri, S., Bekiros, S., Giakoumelou, A., & Bezzina, F. (2020). Performance assessment of ensemble learning systems in financial data classification. Intelligent Systems in Accounting, Finance and Management, 27(1), 3–9. https://doi.org/10.1002/isaf.1460
9. Lin, H.-C., Wang, P., Chao, K.-M., Lin, W.-H., & Yang, Z.-Y. (2021). Ensemble Learning for Threat Classification in Network Intrusion Detection on a Security Monitoring System for Renewable Energy. Applied Sciences, 11(23), 11283. https://doi.org/10.3390/app112311283
10. Mohandas, R., Prasanna, D. S. J. D., Meenakshi, N., R, K., Nathiya, S., & Arivazhagan, N. (2024). An Intelligent Machine Learning Approach to Detect Spam in Social Media. 2024 5th International

Conference on Data Intelligence and Cognitive Informatics (ICDICI), 1149–1154. https://doi.org/10.1109/ICDICI62993.2024.10810782

11. Mukherjee, S., Dey, S., & Acharya, A. (2026). YouTube Spam Comment Detection System. In Lecture Notes in Electrical Engineering: Vol. 1026 LNEE (pp. 189–203). https://doi.org/10.1007/978-981-96-6537-2_13
12. Oh, H. (2022). A YouTube Spam Comments Detection Scheme Using Cascaded Ensemble Machine Learning Model. IEEE Access, 10, 40860–40860. https://doi.org/10.1109/ACCESS.2022.3166635
13. Oluwole Ogini, N., Adigwe, W., & Oghenefego Ogwara, N. (2022). Distributed Denial of Service Attack Detection and Prevention Model for IoT based Computing Environment using Ensemble Machine Learning Approach. International Journal of Network Security & Its Applications, 14(4), 39–53. https://doi.org/10.5121/ijnsa.2022.14403
14. Ostvar, N., & Eftekhari Moghadam, A. M. (2020). HDEC: A Heterogeneous Dynamic Ensemble Classifier for Binary Datasets. Computational Intelligence and Neuroscience, 2020, 1–11. https://doi.org/10.1155/2020/8826914
15. R. Abinaya, B. N. E. and P. N. (2020). Spam Detection On Social Media Platforms. 2020 7th International Conference on Smart Structures and Systems (ICSSS), 1–3. https://doi.org/10.1109/ICSSS49621.2020.9201948
16. Shabadi, L., Chaitra, Y. L., Srikanth, P., Vijay Kumar, L., & Kashyap, U. (2023). Youtube Spam Detection Scheme Using Stacked Ensemble Machine Learning Model. 2023 International Conference on Network, Multimedia and Information Technology, NMITCON 2023, 1–7. https://doi.org/10.1109/NMITCON58196.2023.10276002
17. Sinhal, A., Kumar, P., & Aggarwal, G. (2024). Enhancing YouTube Spam Filtration Efficiency Through Deep Learning Based Techniques. Proceedings - 4th International Conference on Technological Advancements in Computational Sciences, ICTACS 2024, 1893–1896. https://doi.org/10.1109/ICTACS62700.2024.10841327
18. Sinhal, A., & Maheshwari, M. (2022). YouTube: Spam Comments Filtration Using Hybrid Ensemble Machine Learning Models. International Journal of Emerging Technology and Advanced Engineering, 12(10), 169–183. https://doi.org/10.46338/ijetae1022_18
19. Titiani, F., & Riana, D. (2022). Ensemble Learning for the Prediction of Marketing Campaign Acceptance. International Journal of Software Engineering and Computer Systems, 8(2), 67–76. https://doi.org/10.15282/ijsecs.8.2.2022.7.0104
20. Tripathi, A., Bharti, K. K., & Ghosh, M. (2019). A Study on Characterizing the Ecosystem of Monetizing Video Spams on YouTube Platform. Proceedings of the 21st International Conference on Information Integration and Web-Based Applications & Services, 222–231. https://doi.org/10.1145/3366030.3366078
21. Yadav, S., Jena, J. J., Prakash Singh, J., Gourisaria, M. K., Jain, S., & Kumar, V. (2025). Spam Detection in YouTube Comments: A Machine Learning Approach. International Conference on Intelligent Systems and Computational Networks, ICISCN 2025, 1–7. https://doi.org/10.1109/ICISCN64258.2025.10934150
22. Yanto, J., Tandiono, R. D., Wulandari, L. A., & Nabiilah, G. Z. (2025). Spam Detection on YouTube Comment Section: Comparison Between Deep Learning and Machine Learning Methods. Proceedings of the 2025 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology, IAICT 2025, 753–759. https://doi.org/10.1109/IAICT65714.2025.11100517
23. You, G., Shiue, Y., Yeh, W., Chen, X., & Chen, C. (2020). A Weighted Ensemble Learning Algorithm Based on Diversity Using a Novel Particle Swarm Optimization Approach.
24. Zhang, M., He, Z., Zhang, H., Tan, T., & Sun, Z. (2019). Toward practical remote iris recognition: A boosting based framework. Neurocomputing, 330, 238–252. https://doi.org/10.1016/j.neucom.2017.12.053