# GROUP NAME:

CurepeQualityCode

# PROJECT TITLE:

Feature Selection Algorithm: Cattle Corral Optimization (CCO)

# GROUP MEMBERS:

| Student ID | Student Name |
|---|---|
| 816036227 | Aidan Nagaur |
| 816036514 | Anderson Singh |
| 816025421 | Katelyn Ramphal |

# SUPERVISOR:

Dr. Kris Manohar

Dr. Amit Ramkissoon

# Table of Contents

# Abstract

Feature selection is a crucial process in machine learning that enhances model performance by identifying and selecting the most relevant features. Effective feature selection techniques reduce overfitting, increase generalization, and decrease computation time. This study aims to develop a unique feature selection algorithm inspired by the corralling cattle optimization (CCO) technique and compares its performance to established methods, such as filter, wrapper, and embedded approaches.

A comprehensive literature review is conducted to examine the simulated annealing technique and five existing algorithms: Binary PSO with Mutation Operator, Grey-Sail Fish Optimization (G-SFO), Whale Optimization Algorithm (WOA), Horse Herd Optimization Algorithm (HOA), and Hybrid Squirrel-Dragonfly Search Optimization (HS-DSO), focusing on their distinguishing characteristics. The CCO algorithm is initialized with a bias toward top-ranked features, and its update mechanisms simulate behaviors inspired by egrets and flies to balance exploration and exploitation. The performance of all algorithms is evaluated using key metrics such as AUC, F1, accuracy, precision, sensitivity, specificity, FPR, FNR, MCC, and n-feat.

The results demonstrate that the CCO algorithm has a slightly superior performance compared to the baseline methods in a number of key metrics. It achieves the highest AUC (0.979) and Precision (0.999), outperforming HSDO, PSO, and GSF, while achieving comparable results in F1 score, accuracy, and MCC. Notably, CCO has a lower false negative rate (FNR) (0.130) than a number of baseline methods, which is important for tasks that require high sensitivity. By leveraging a population-based strategy with additional random exploration, the CCO algorithm is both efficient and effective, outperforming traditional methods with fewer selected features

# Introduction

Feature selection plays a critical role in the field of machine learning, directly influencing model performance by identifying and retaining only the most informative features while discarding redundant or irrelevant ones. This critical pre-processing step has several advantages: it improves generalization capabilities, reduces overfitting, enhances interpretability, and significantly accelerates computational efficiency. As machine learning applications extend across various domains, the importance of efficient and effective feature selection techniques has increased significantly. In real-world applications such as spam detection and fake news classification, effective feature selection can significantly improve model accuracy and computational performance.

Traditional feature selection approaches generally fall into three categories: filter methods, which select features based on statistical measures unrelated to the learning algorithm; wrapper methods, which evaluate feature subsets using the target learning algorithm; and embedded methods, which incorporate feature selection into the model training process. While these traditional techniques have proven beneficial, they often come with limitations when being applied to complex, high-dimensional datasets, such as computational inefficiency, a tendency to converge to local optima, and difficulties in balancing exploration and exploitation throughout the search process. These issues become particularly evident when dealing with large-scale datasets that require real-time processing, such as spam filtering or misinformation detection.

To address these challenges, nature-inspired metaheuristic algorithms have emerged as potential alternatives for feature selection tasks. These algorithms, which are based on biological and natural processes, provide flexibility, global search capabilities, and ease of hybridization, making them suitable for solving high-dimensional and non-convex problems. Notable examples include Particle Swarm Optimization (PSO), which simulates particle information sharing; Whale Optimization Algorithm (WOA), based on humpback whale hunting tactics; and Horse Herd

Optimization Algorithm (HOA), inspired by equine social dynamics. More recent innovations include the Squirrel-Dragonfly Hybrid Optimization Algorithm, which combines the behaviors of squirrels and dragonflies, and the Grey-Sail Fish Optimization, both of which demonstrate the power of nature-inspired strategies in addressing optimization challenges.

This study builds on these innovations by introducing a novel nature-inspired algorithm called Corralling Cattle Optimization (CCO), which is inspired by the symbiotic relationship between cattle along with birds (egrets) and insects (flies). The CCO algorithm relies on a dual-mechanism optimization strategy, with one simulating the strategic movement of egrets toward promising solution regions, representing directed exploration, and the other simulating the localized random movements of flies around cattle, allowing for fine-grained exploitation of high-potential areas. These behaviors are incorporated into a population-based search model that optimizes both feature selection and hyperparameter tuning, resulting in a more efficient approach to model development.

Our proposed CCO algorithm stands out by its initialization bias toward high-ranking features, which was determined via a preliminary assessment using weighted correlation and mutual information metrics. This approach encompasses domain knowledge early in the optimization process, resulting in faster convergence and higher solution quality. Furthermore, the algorithm includes an adaptive stopping mechanism based on fitness stagnation, which improves computational efficiency by avoiding redundant evaluations when improvement plateaus.

This research's primary contributions are as follows: (1) the development of the CCO algorithm, integrating global exploration (egret behavior) and local exploitation (fly behavior) through its distinctive dual-mechanism approach; (2) a hybrid optimization framework that simultaneously selects relevant features and optimizes model hyperparameters; and (3) a comprehensive comparative evaluation against established feature selection techniques, including filter, wrapper, and embedded methods, as well as other nature-inspired optimization algorithms.

The purpose of this work is to compare the performance of the CCO algorithm to conventional and modern feature selection methods using key metrics such as AUC, F1-score, accuracy, precision, sensitivity, specificity, false positive rate (FPR), false negative rate (FNR), Matthews correlation coefficient (MCC), and the number of selected features (n-feat). By addressing the scalability and computational issues present in current approaches, the aim is to use CCO to demonstrate improved performance in real-world applications. The remainder of the paper is structured as follows: Section 2 conducts a thorough literature review of existing feature selection approaches and nature-inspired optimization algorithms. Section 3 describes the methodology and implementation details for the proposed CCO algorithm. Section 4 describes the experimental set-up and evaluation metrics. Section 5 discusses the findings and comparative analysis. Finally, Section 6 summarizes the findings and suggests areas for future research.

# Literature Review

Spam and fake news identification have become critical areas of research due to the increasing challenges posed by unsolicited emails and misinformation, which endangers data security and societal well-being. Effective feature selection is critical in these domains because it reduces dimensionality, increases interpretability, and improves classification performance. For enhancing detection accuracy, a variety of optimization-based feature selection techniques and machine learning models have been developed. This review examines key studies on optimization strategies for feature selection in spam and fake news detection.

Zhang et al. (2014) proposed a spam detection framework that combines Binary PSO with a mutation operator for feature selection, resulting in high classification accuracy and minimal false positives. This method prevents premature convergence, a prevalent issue in traditional PSO, by incorporating diversity into feature selection. Despite its enhanced performance, PSO's computational complexity remains a barrier to real-time applications. Similarly, Kadam and Rohokale (2022) proposed a hybrid model for multi-objective feature selection that combined Grey-Sail Fish Optimization (G-SFO) and Capsule Networks (CapsNets). While this model is effective at managing both text- and image-based spam, it has high computational costs and scalability issues, limiting its use in real time.

Shuaib et al. (2019) utilized the Whale Optimization Algorithm (WOA) in conjunction with the Rotation Forest classifier for feature selection in spam detection. The WOA-based method outperformed traditional classifiers in terms of accuracy and false positive rates, but had scalability and dataset diversity limitations. Meanwhile, Hosseinalipour and Ghanbarzadeh (2022) proposed the Horse Herd Optimization Algorithm (HOA) for spam detection, which balances exploration and exploitation using herd behavior. Although the HOA method demonstrated superior accuracy and sensitivity, scalability and computational complexity remain barriers to practical application.

Nithya and Sahayadhas (2023) introduced the Meta-Heuristic Searched-Ensemble Learning (MS-EL) method for detecting fake news, which incorporates Hybrid Squirrel-Dragonfly Search Optimization (HS-DSO) for optimal feature selection. While the ensemble learning approach produced high accuracy, it was limited by computational overhead and overfitting risks. Furthermore, Ramsubhag et al. (2025) proposed a modified Golden Section Search (GSS) technique for hyperparameter tuning in optimization-based feature selection, which provides valuable insights into increasing computational efficiency without sacrificing model accuracy.

The studies reviewed focus on the evolution of feature selection techniques, from heuristic-based optimization to hybrid and ensemble approaches. While these methods improve accuracy and adaptability, common issues like high computational complexity and scalability persist. Future research should concentrate on improving real-time integration, investigating hybrid optimization strategies, and increasing the scalability of these models for use in spam and fake news detection.

*Table 1: Summary of Research Papers*

| Title of Paper | Author | Key Characteristics | Strengths | Limitations |
|---|---|---|---|---|
| Binary PSO with Mutation Operator for Spam Detection | Zhang et al. (2014) | Combines Binary Particle Swarm Optimization (PSO) with a mutation operator for feature selection and uses a decision tree classifier (C4.5 algorithm). | Reduces false positives, improves classification accuracy, and prevents premature convergence by introducing diversity in feature selection. | High computational complexity limits real-time applicability. |
| Hybrid Meta-Heuristic and Capsule Network-Based Spam Detection | Kadam and Rohokale (2022) | Integrates Grey-Sail Fish Optimization (G-SFO) for multi-objective feature selection and adaptive Capsule Networks | Versatile, capable of handling both text-based and image-based spam; achieves | High computational costs and scalability issues; unsuitable for |

| | | | | |
|---|---|---|---|---|
| | | (CapsNets) for classification. | high accuracy and precision using advanced feature extraction techniques. | real-time applications. |
| Whale Optimization Algorithm-Based Feature Selection for Spam Detection | Shuaib et al. (2019) | Uses Whale Optimization Algorithm (WOA) for feature selection and Rotation Forest (RF) classifier for spam detection | Outperforms traditional classifiers like Naïve Bayes and Rain Forest in accuracy and false positive rates. | Limited dataset diversity, scalability issues, and lack of real-time applicability. |
| A Novel Approach for Spam Detection Using Horse Herd Optimization Algorithm | Hosseinalipour and Ghanbarzadeh (2022) | Introduces a modified binary and multi-objective Horse Herd Optimization Algorithm (MOBHOA) for feature selection and classification. | Outperforms standard classifiers like KNN, Naïve Bayes, and SVM in accuracy, sensitivity, and computational complexity. | Scalability issues due to high computational resource requirements. |
| Meta-heuristic Searched-Ensemble Learning for fake news detection with optimal weighted feature selection approach | Nithya and Sahayadhas (2023) | Proposes a Hybrid Squirrel–Dragonfly Search Optimization (HS-DSO) algorithm for feature selection and ensemble learning (LSTM, SVM, DNN) for fake news classification. | High classification accuracy and adaptability to complex classification tasks. | High computational cost, risk of overfitting, and limited real-time detection capabilities. |
| Efficient Hyper-Parameter Tuning for the Kappa Regression Algorithm | Ramsubhag et al. (2025) | Introduces a modified Golden Section Search (GSS) technique for hyper-parameter tuning in the Kappa Regression algorithm | Efficiently reduces computational overhead while maintaining model accuracy; applicable to optimization problems in multiple domains. | Limited direct application to spam detection; focuses on broader optimization challenges. |

# Methodology / Proposed Scheme

To address the challenge of jointly optimizing feature selection and model hyperparameters for binary classification in fake news detection, this study introduces a novel hybrid metaheuristic algorithm called Cattle Corral Optimization (CCO). Inspired by natural herd dynamics, CCO integrates statistical feature relevance with biologically inspired movement strategies to efficiently explore the search space. The methodology centers on modelling each candidate solution as a "cow" that encodes both a feature subset and a configuration of hyperparameters for a Gradient Boosting Classifier (GBC). By leveraging mutual information and weighted correlation for initial feature relevance and employing Egret and Fly update strategies for exploration and exploitation, the algorithm dynamically converges on high-performing solutions. This section details the components, heuristics, and operational flow of the CCO algorithm, demonstrating its suitability for scalable and adaptive classification tasks involving high-dimensional data.

**Cattle Corral Optimization (CCO) Algorithm**

In this study, we introduce the Cattle Corral algorithm, a novel hybrid metaheuristic approach that performs simultaneous feature selection and hyperparameter optimization for classification tasks. The algorithm conceptualizes each candidate solution as a cow, representing both a binary feature mask and a set of hyperparameters for a machine learning model, in this case, a Gradient Boosting Classifier (GBC). The objective is to identify the most informative subset of features and corresponding hyperparameters that jointly maximize classification performance, measured by a weighted combination of the Area Under the Receiver Operating Characteristic Curve (AUC) and F1 score.

To guide the search process, the Cattle Corral algorithm employs two nature-inspired movement strategies: Egret and Fly updates. These strategies mimic natural foraging and search behaviors, with the current best-performing cattle acting as an attractor. Egrets simulate a broader, more exploratory behavior by moving toward the best cattle with an adaptive jump factor and random perturbations, enhancing the global exploration of the solution space. This allows the algorithm to escape local optima and explore new promising areas. In contrast, Flies represent more exploitative agents that refine candidate solutions through smaller, localized updates and an attraction component, allowing for intensified search around high-performing regions.

At each iteration, a population of cattle is evaluated based on their fitness, and the best cattle is updated accordingly. If a candidate's AUC exceeds the previous best, the corresponding feature subset and hyperparameters are stored as the new best solution, and a patience counter is reset. If no improvement occurs over a specified number of iterations, the algorithm terminates early to avoid unnecessary computation. Importantly, the feature selection process is biased using an initial ranking that combines weighted correlation and mutual information, ensuring that the initial population is informed by statistically relevant features. A small random perturbation is added to the combined scores to prevent identical rankings due to ties.

With biologically inspired agents and intelligent initialization strategies, the Cattle Corral algorithm achieves a balance between exploration and exploitation, efficiently navigating the complex search space of feature-hyperparameter combinations. This methodology provides a scalable and adaptable framework for improving classification performance in binary prediction tasks involving high-dimensional data.

**Detailed look at the CCO Algorithm**

1. **Weighted Pearson Correlation Coefficient:**

   The weighted Pearson correlation measures the linear relationship between a feature $x$ and the target $y$ , while considering the importance or frequency of each observation through a weight vector $w$. It is calculated as:

$$\textbf{Weighted Correlation} = \frac{\sum w_i(x_i - \overline{x}_w)(y_i - \overline{y}_w)}{\sqrt{\sum w_i(x_i - \overline{x}_w)^2} \cdot \sqrt{\sum w_i(y_i - \overline{y}_w)^2}}$$

   Where:

   - $x_i, y_i =$ Values of the feature and target variable for the $i^{th}$ instance
   - $w_i =$ Weight of the $i^{th}$ instance
   - $\bar{x}_w = \sum w_i x_i,$ weighted mean of feature $x$
   - $\bar{y}_w = \sum w_i y_i,$ weighted mean of target $y$

2. **Mutual Information:**

   Mutual Information quantifies the amount of information obtained about one random variable through another. In this context, it measures how much knowing the feature $X$ reduces uncertainty about the class label $Y$.

   The mutual information between two discrete random variables $X$ and $Y$ is defined as:

$$I(X;Y) = \sum_{x}\sum_{y} p(x,y).\log\left(\frac{p(x,y)}{p(x).p(y)}\right)$$

Where:

- $p(x,y) =$ Joint probability distribution of $X$ and $Y$
- $p(x) =$ Marginal probability of $X$
- $p(y) =$ Marginal probability of $Y$

3. **Min-Max Normalization:**

To scale the correlation and mutual information values between 0 and 1, min-max normalization is applied:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

4. **Combined Feature Score:**

After normalization, the weighted correlation and mutual information are combined using a linear combination:

$$\textbf{Combined Score} = w_c\, .NormCorr + w_m\, .NormMI$$

Where:

- $w_c, w_m =$ User-defined weights for correlation and mutual information
- $NormCorr =$ Normalized weighted correlation
- $NormMI =$ Normalized mutual information

5. **Egret Update:**

The Egret update models global exploration and is given by:

$$x_{new} = x + r\,.\,(x_{best} - x) + \delta$$

Where:

- $x =$ Current candidate position (solution vector)

- $x_{best}$ = Best-known candidate (global best)
- $r \in [0, 0.7]$ = Random jump factor
- $\delta \in [-0.3,0.3]$ = Small random noise vector to introduce diversity

6. **Fly Update:**

The Fly update emphasizes local exploitation and is defined as:

$$\boldsymbol{x_{new} = x + \eta + \alpha\,.\,(x_{best} - x)}$$

**Where:**

- $\eta \in [-0.15,0.15]$: Local random movement
- $\alpha = 0.4$: Attraction weight toward the best candidate
- $x =$ Current candidate position (solution vector)
- $x_{best}$ = Best-known candidate (global best)

**Algorithm 1**: Proposed model of CCO

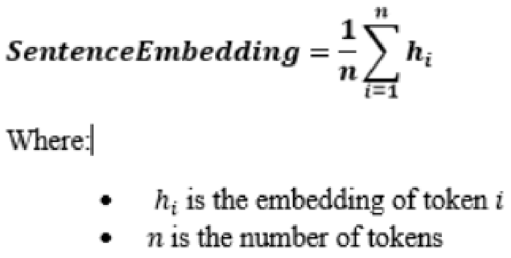
```
Compute weighted correlation Eq. (1)
Compute mutual information Eq. (2)
Normalize correlation and mutual information Eq. (3)
Combine both scores with assigned weights Eq. (4)
Add small random noise to avoid ties
Sort features by combined score (descending)
Boot initial population based on sorted features and assign random hyperparameter values
Evaluate fitness of all candidates

        If current best AUC > previous best AUC:
                Update best solution
        Reset patience counter

    Else
        Increment no improvement counter
        If no improvement ≥ patience
            Stop early

Generate a random number between 0 and 1

If random < update criterion
    Update position using Egret update Eq. (5)

Else
    Update position using Fly update Eq. (6)

Clip new position to remain within bounds
Return optimal solution
```

# Experimental Results

## Design

Data pre-processing is a crucial step for allowing machine learning models to perform optimally. Raw data is rarely clean or formatted in a way such that a model can understand or learn from effectively. The following are the pre-processing steps used to ensure optimal model performance:

**Text Data:** The CCO algorithm focuses on model optimization for fake news detection. In this field, the context of text is crucial for models to detect patterns for optimal classification. We use the Sentence Transformers framework which provides the pre-trained model **all-MiniLM-L6-v2.** This model is compact and powerful, it is used to generate dense vector representations (embeddings) for sentences, paragraphs or short documents. This model outputs a 384-dimentional sense sentence vector which is useful for classification. This is achieved through the following formula:



$$\boldsymbol{SentenceEmbedding = \frac{1}{n}\sum_{i=1}^{n} h_i}$$

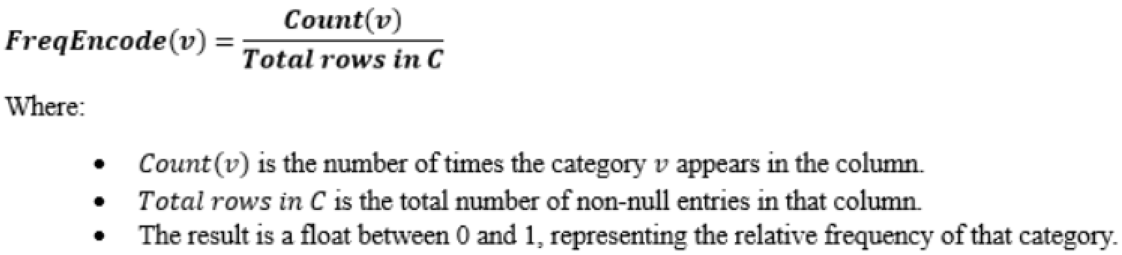
Where:

- $h_i$ is the embedding of token $i$
- $n$ is the number of tokens

**Date:** Date is split into hour, weekday, month and year. This is important to reveal behavioural patterns in users. For example, there may be a higher chance of fake news being posted at night or during the weekend.

**Categorical Variables:** Categorical variables are replaced with their relative frequencies in the dataset. This works well for high-cardinality values.

For a categorical column, C and a specific value , the frequency encoding value is:



$$FreqEncode(v) = \frac{Count(v)}{Total\ rows\ in\ C}$$

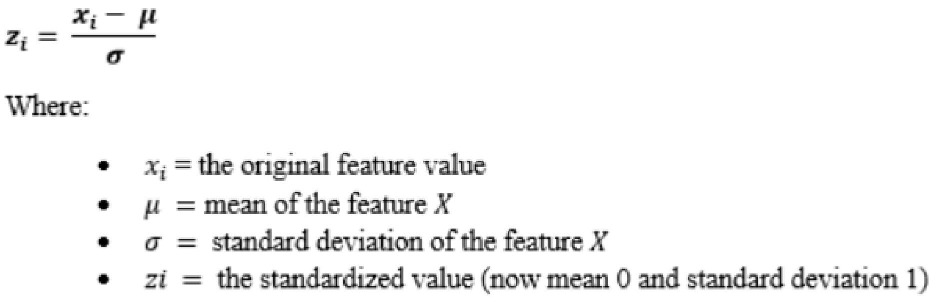
Where:

- $Count(v)$ is the number of times the category $v$ appears in the column.
- $Total\ rows\ in\ C$ is the total number of non-null entries in that column.
- The result is a float between 0 and 1, representing the relative frequency of that category.

**Class Labels:** String-based class labels are replaced with meaningful ordered integers. This ensures that label encoding reflects the ordinal relationship among classes.

**Feature Scaling:** Applies standard scaling where each feature is transformed to have a mean of 0 and a standard deviation of 1. This prevents classes with high values from dominating models.

For a given feature column X, each value $x_i$ is transformed as:

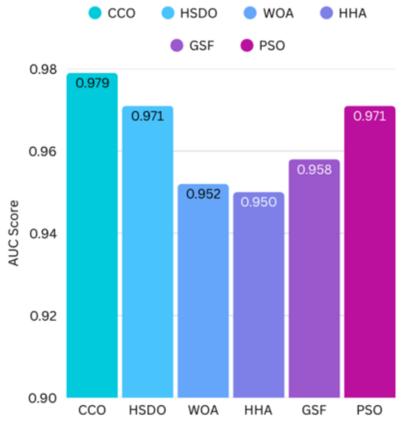


$$z_i = \frac{x_i - \mu}{\sigma}$$

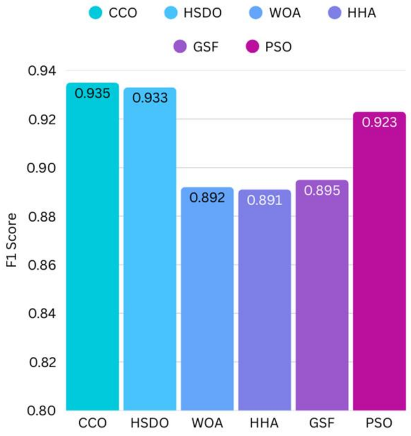
Where:

- $x_i$ = the original feature value
- $\mu$ = mean of the feature $X$
- $\sigma$ = standard deviation of the feature $X$
- $zi$ = the standardized value (now mean 0 and standard deviation 1)

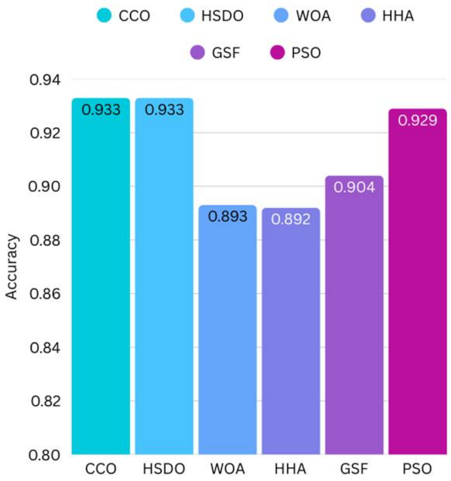
## Results



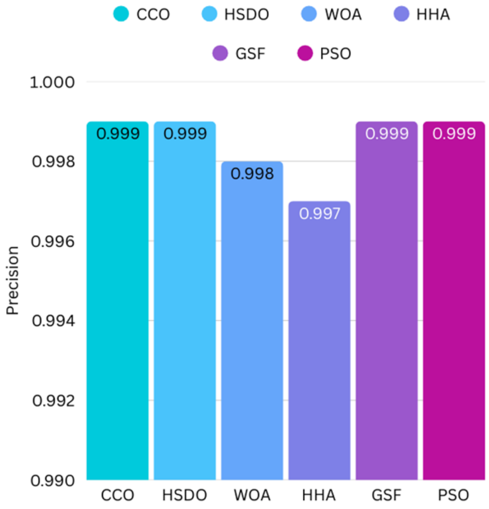
Graph 1.0 comparing the AUC scores for the 6 algorithms. The Cattle Corral Optimization (CCO) algorithm achieves the highest AUC score of 0.979, indicating superior classification performance.



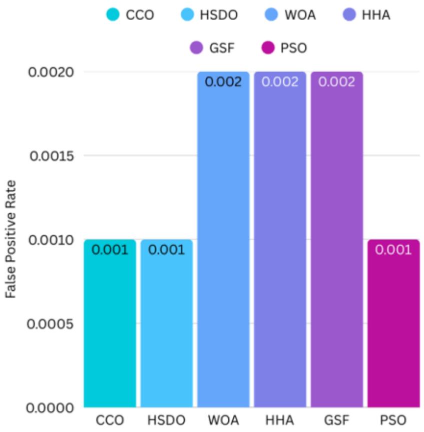
Graph 2.0 comparing the F1 scores for the 6 algorithms. CCO achieved the highest F1 score (0.935), indicating the best balance between precision and recall.



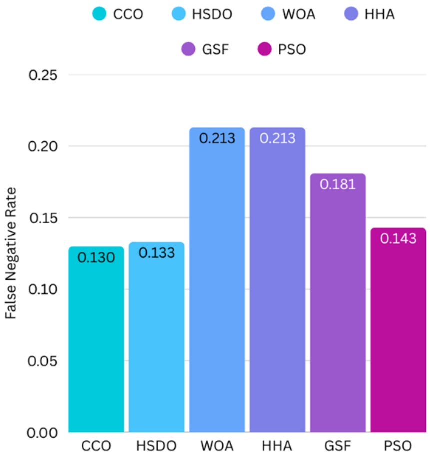
Graph 3.0 displaying the accuracy of the 6 algorithms. CCO and HSDO both achieved the highest accuracy at 0.933, followed closely by PSO at 0.929.



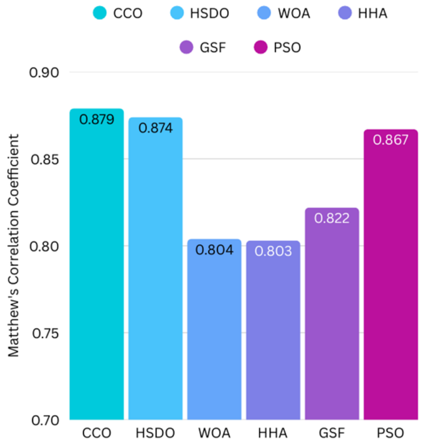
Graph 4.0 comparing the precision score of the 6 algorithms. CCO, HSDO, GSF, and PSO all achieved a near-perfect precision of 0.999, indicating extremely low false positive rates.



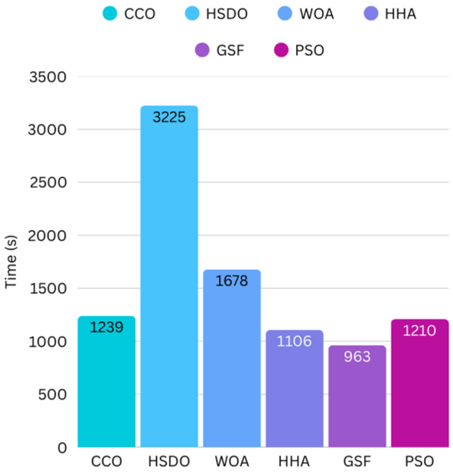
Graph 5.0 showing the False Positive Rate across the 6 algorithms. CCO, HSDO, and PSO achieved the lowest FPR of 0.001, indicating strong reliability in minimizing incorrect positive classifications. In contrast, WOA, HHA, and GSF showed slightly higher FPRs at 0.002, suggesting a marginally increased risk of false alarms.



Graph 6.0 shows the False Negative Rate (FNR) for six optimization algorithms. CCO achieved the lowest FNR at 0.130, indicating strong effectiveness in correctly identifying positive cases. In contrast, WOA and HHA had the highest FNRs at 0.213, reflecting a greater tendency to miss true positives.



Graph 7.0 compares the Matthews Correlation Coefficient (MCC) across six optimization algorithms. CCO achieved the highest MCC value of 0.879, indicating the most balanced and reliable performance across all classes. Closely following were HSDO (0.874) and PSO (0.867), while WOA and HHA trailed with lower overall correlation strength.



Graph 8.0 shows the execution time (in seconds) for each optimization algorithm. GSF was the fastest at 963 seconds, followed by HHA and PSO, while HSDO was the slowest at 3225 seconds. CCO completed in 1239 seconds, offering a strong balance between computational efficiency and high performance.

| Performance Metrics | **Algorithms** | | | | | |
|---|---|---|---|---|---|---|
| | **CCO** | **HSDO** | **WOA** | **HHA** | **GSF** | **PSO** |
| **AUC** | 0.979 | 0.971 | 0.952 | 0.95 | 0.958 | 0.971 |
| **F1** | 0.935 | 0.933 | 0.892 | 0.891 | 0.895 | 0.923 |
| **Accuracy** | 0.933 | 0.933 | 0.893 | 0.892 | 0.904 | 0.929 |
| **Precision** | 0.999 | 0.999 | 0.998 | 0.997 | 0.999 | 0.999 |
| **FPR** | 0.001 | 0.001 | 0.002 | 0.002 | 0.002 | 0.001 |
| **FNR** | 0.13 | 0.133 | 0.213 | 0.213 | 0.181 | 0.143 |
| **MCC** | 0.879 | 0.874 | 0.804 | 0.803 | 0.822 | 0.867 |
| **Time (s)** | 1239 | 3225 | 1678 | 1106 | 963 | 1210 |

*Table 1.0 shows the summary of results for all algorithms.*

# Discussion

The primary objective of this research was to develop and evaluate the effectiveness of the proposal Cattle Corral Optimization (CCO) algorithm in addressing key challenges in spam and fake news detection particularly for feature selection. The results indicate that CCO outperforms several established metaheuristic algorithms, including Hybrid Squirrel Dragonfly Optimization (HSDO), Whale Optimization Algorithm (WOA), Horse Heard Algorithm (HHA) , Grey Scale Fish (GSF) and Particle Swarm Optimization(PSO) across multiple performance metrics especially Area Under the Curve(AUC).

To ensure a fair and effective evaluation of the CCO algorithm a vigorous data processing pipeline was implemented. Given the focus on fake news detection special attention was placed on the text representation. The Sentence Transformers framework with the pre-trained all-MiniLM-L6-v2 model to generate compact 384-dimensional sentence embeddings. This enhanced the model's ability to recognize contextual patterns. Temporal features were extracted by decomposing date fields into hour, weekday, month, and year components, enabling the model to capture behavioural trends, such as time-based posting patterns. Categorical variables were transformed using frequency encoding. This is particularly effective for high-cardinality categories by preserving distributional information. Class labels were converted to ordered integers to reflect any inherent ranking, while standard scaling was applied to all features to ensure uniform contribution during training. We also implemented SMOTE (Synthetic Minority Over-Sampling

Technique) to address the class imbalance in the dataset. We also used k-fold cross-validation to evaluate model robustness, reducing variance by averaging results across multiple splits. This approach ensures that each instance in the dataset is used for both training and validation, enhancing generalizability.

After testing, CCO achieved the highest AUC score of 0.979, displaying its ability to reliably differentiate between spam and legitimate content. This supports the idea that a hybrid behaviourally inspired search mechanism like combining egret-driven exploitation with fly-like random exploration can lead to superior classification. The algorithm also recorded the highest F1 score (0.935) and accuracy (0.933), suggesting a well-balanced trade-off between precision and recall which is essential in scenarios where both false positives and false negatives have serious implications.

Although most algorithms demonstrated high precision (≥ 0.997) the False Negative Rate (FNR) serves as a differentiator. CCO FNR of 0.130 is amongst the lowest, indicating a reduced likelihood of misclassifying spam. Also the Mathews Correlation Coefficient (MCC) of 0.879, the highest of all models tested reflect the CCO strong and balanced predictive power across both positive and negative classes.

With respect to computational cost, CCO requires 1239 seconds for execution, which is higher than the HHA and GSF but substantially lower than the HSDO and WOA. This places CCO in a practical middle ground. Although not the fastest but for the time taken it offers a strong predictive performance without prohibitive computational demands. This is important for real-time or near-real time applications.

These results show that the efficiency and effectiveness of metaheuristic optimization can be improved by combining domain-informed initialization techniques (e.g., weighted correlation and mutual information) with adaptive exploration-exploitation strategies. Inspired by natural behaviour, the stochastic components make sure that the search process converges toward globally optimal solutions while avoiding local optima.

Overall, the findings validate the CCO algorithm as a compelling contribution to the field of intelligent content filtering and classification. Future directions may include parallelization to reduce runtime, application to multi-label or multi-class datasets, hyper parameter refinement, and testing under adversarial or noisy input conditions to assess robustness.

# Conclusion

In order to improve feature selection, this study proposed the Cattle Corral Optimization (CCO) algorithm, a novel metaheuristic inspired by animal behaviour. While maintaining competitive runtime, CCO outperformed current methods on a number of metrics, including the highest AUC (0.979) and F1-score (0.935).

These outcomes demonstrate how well the algorithm strikes a balance between exploration and exploitation, as well as how useful it is for tasks like spam and fake news detection. Particularly significant was the incorporation of dynamic search behaviours and domain-informed initialization.

Future research might concentrate on real-time settings, hyperparameter improvement, expanding CCO to multi-class problems, and enhancing efficiency through parallel computing. All things considered, CCO presents a viable path toward clever, flexible machine learning optimization.

# Research Paper Details

**Current Status:**
All project requirements in accordance with the course objectives have successfully been fulfilled and a full draft of the research paper detailing the development and evaluation of the Cattle Corral Optimization (CCO) algorithm have been completed. However, prior to formal IEEE submission, the feasibility of implementing additional improvements or alternative methods suggested during peer review and faculty consultations is being evaluated. The goal is to enhance the algorithm's performance and contribution, given that the current results show only marginal superiority in several performance metrics

**Submission Target:** IEEE International Conference on Computational Intelligence and Applications (IEEE ICCIA), 2025.

**Proposed Submission Date:** May 30, 2025

**Expected Notification of Acceptance:** Approximately three (3) to six (6) months after submission.

**Paper Availability (Upon Acceptance):** The final manuscript will be publicly accessible via the conference proceedings on IEEE Xplore Digital Library.

**Additional Plans for Publication:** Depending on reviewer feedback and acceptance, we may consider extending the research to additional applications or domains, subsequently submitting extended versions or related research to other high-impact journals.

# Acknowledgements

We would like to express our heartfelt gratitude to the individuals and organizations listed below for their invaluable support and contributions throughout this project: We thank our supervisors, Dr. Kris Manohar and Dr. Amit Ramkissoon, for their expert guidance, insightful feedback, and continual encouragement, which greatly improved the quality of this report. The faculty and staff of the Department of Computer Science for providing the resources and support required to conduct this study. We thank our colleagues and peers for their helpful discussions and suggestions while developing the Cattle Corral Optimization (CCO) algorithm. Finally, we want to thank our families and friends for their unwavering support and understanding throughout this journey.

# References

1. Cervante, Liam, Bing Xue, Mengjie Zhang, and Lin Shang. 2012. "Binary Particle Swarm Optimisation for Feature Selection: A Filter Based Approach." In *2012 IEEE Congress on Evolutionary Computation*, 1–8. IEEE.
2. Hancer, Emrah, Bing Xue, and Mengjie Zhang. 2018. "Differential Evolution for Filter Feature Selection Based on Information Theory and Feature Ranking." *Knowledge-Based Systems* 140: 103–119.

3. Hosseinalipour, Ali, and Reza Ghanbarzadeh. 2022. "A Novel Approach for Spam Detection Using Horse Herd Optimization Algorithm." *Neural Computing and Applications* 34: 13091–13105. https://doi.org/10.1007/s00521-022-07148-x.
4. Li, An-Da, Bing Xue, and Mengjie Zhang. 2021. "Improved Binary Particle Swarm Optimization for Feature Selection with New Initialization and Search Space Reduction Strategies." *Applied Soft Computing* 106: 107302.
5. Nithya, S. Hannah, and Arun Sahayadhas. 2023. "Meta-Heuristic Searched-Ensemble Learning for Fake News Detection with Optimal Weighted Feature Selection Approach." *Data & Knowledge Engineering* 144: 102124. https://doi.org/10.1016/j.datak.2022.102124.
6. Ramsubhag, Deepak, Kevin Baboolal, and Patrick Hosein. 2025. "Efficient Hyper-Parameter Tuning for the Kappa Regression Algorithm." *Journal of Computational Science* 18: 102342.
7. Samarthrao, Kadam Vikas and Vandana M. Rohokale. 2022. "A Hybrid Meta-Heuristic-Based Multi-Objective Feature Selection with Adaptive Capsule Network for Automated Email Spam Detection." *International Journal of Intelligent Robotics and Applications* 6 (3): 497–521. https://doi.org/10.1007/s41315-021-00217-9.
8. Shafiq, Muhammad, Zhihong Tian, Ali Kashif Bashir, Xiaojiang Du, and Mohsen Guizani. 2020. "IoT Malicious Traffic Identification Using Wrapper-Based Feature Selection Mechanisms." *Computers & Security* 94: 101863.
9. Shuaib, Maryam, Shafi'i Muhammad Abdulhamid, Olawale Surajudeen Adebayo, Oluwafemi Osho, Ismaila Idris, John K. Alhassan, and Nadim Rana. 2019. "Whale Optimization Algorithm-Based Email Spam Feature Selection Method Using Rotation Forest Algorithm for Classification." *SN Applied Sciences* 1: 390.

https://scispace.com/pdf/whale-optimization-algorithm-based-email-spam-feature-ycll2wnj6l.pdf.

10. Zhang, Yudong, Shuihua Wang, Preetha Phillips, and Genlin Ji. 2014. "Binary PSO with Mutation Operator for Feature Selection Using Decision Tree Applied to Spam Detection." *Knowledge-Based Systems* 64: 22–31. https://doi.org/10.1016/j.knosys.2014.03.015. (https://www.sciencedirect.com/science/article/abs/pii/S095070511400104X).

Sci-Kit Learn:

1. "“DevDocs — Scikit-Learn Documentation.” n.d. Devdocs.io. https://devdocs.io/scikit_learn/.
2. Scikit-learn. 2024. “Scikit-Learn: Machine Learning in Python.” Scikit-Learn.org. 2024. https://scikit-learn.org/stable/.
3. “Scikit-Learn Python Tutorial | Machine Learning with Scikit-Learn.” n.d. YouTube. Accessed April 15, 2025. http://www.youtube.com/playlist?list=PLS1QulWo1RIa7ha9SewcZlsTQVwL7n7oq.

# Appendices

## Resource Links

Trello Board:
https://trello.com/invite/b/6792dc9594036da7f285e8bc/ATTI923c740cc5587e78835b72d5a6f9e944C78903FD/info-3604-project

GitHub Repository:

https://github.com/INFO-3604-Project/Cattle-Corral-Project

Collab Notebook:

https://colab.research.google.com/github/INFO-3604-Project/Cattle-Corral-Project/blob/main/main.ipynb

Research Papers (OneDrive):

Research Papers

## Sprint Report

# Sprint Number 1

## Planning:

| Owner | Task Number & Title |
|---|---|
| Anderson | #1 Setup Organization & Repositories |
| Aidan | #2 Aidan Nagaur: Assigned Research Papers (Literature Review) |
| Anderson | #3 Anderson Singh: Assigned Research Papers (Literature Review) |
| Katelyn | #4 Katelyn Ramphal: Assigned Research Papers (Literature Review) |
| Katelyn | #5 Literature Review |
| Aidan | #6 Design a Test for Feature Value |

| | |
|---|---|
| Anderson | #7 Implementing Baseline Tests (Wrapper and Embedded) |
| Katelyn | #8 Implementing Baseline Tests (Filter) |
| Katelyn | #9 Literature Review: Efficient Hyper-Parameter Tuning for the Kappa Regression Algorithm |

## Review:

| Owner | Task Number & Title | Status | Comments |
|---|---|---|---|
| Anderson | #1 Setup Organization & Repositories | Completed | |
| Aidan | #2 Aidan Nagaur: Assigned Research Papers (Literature Review) | Completed | |
| Anderson | #3 Anderson Singh: Assigned Research Papers (Literature Review) | Completed | |
| Katelyn | #4 Katelyn Ramphal: Assigned Research Papers (Literature Review) | Completed | Observed similar or contrasting techniques used. |
| Katelyn | #5 Literature Review | Completed | Comparisions between similar or contrasting techniques across each algorithm. |
| Aidan | #6 Design a Test for Feature Value | Completed | |
| Anderson | #7 Implementing Baseline Tests (Wrapper and Embedded) | Completed | |
| Katelyn | #8 Implementing Baseline Tests (Filter) | Completed | A learning experience. |
| Katelyn | #9 Literature Review: Efficient Hyper-Parameter Tuning for the Kappa Regression Algorithm | Completed | Observed similar or contrasting techniques used. |

Retrospective: This sprint started rough as the group did not complete all literature reviews. The group then quickly and effectively completed their respective literature reviews and some early interpretation of the research papers.

## Planning Sprint 2:

| Owner | Task Number & Title |
|---|---|
| Aidan, Anderson,Katelyn | #10 Algorithm Discussion |
| Aidan, Anderson, Katelyn | #11 Baseline Testing |
| Aidan, Anderson, Katelyn | #12 Version 1 of Proposed Algorithm |

---

# Sprint Number 2

## Planning:

| Owner | Task Number & Title |
|---|---|
| Aidan | #10 Algorithm Discussion: Whale Optimization |
| Aidan | #11 Algorithm Discussion: Horse Herd |
| Katelyn | #12 Algorithm Discussion: Binary PSO with Mutation Operator |
| Katelyn | #13 Algorithm Discussion: Grey-Sail Fish |
| Anderson | #14 Algorithm Discussion: Hybrid Squirrel–Dragonfly |
| Aidan | #15 Baseline Code: Whale Optimization |
| Aidan | #16 Baseline Code: Horse Herd |
| Katelyn | #17 Baseline Code: Binary PSO with Mutation Operator |
| Katelyn | #18 Baseline Code: Grey-Sail Fish |

| | |
|---|---|
| Anderson | #19 Baseline Code: Hybrid Squirrel–Dragonfly |
| Aidan, Anderson, Katelyn | #20 Version One of the Algorithm |
| Anderson | #21 Data Compilation and Presentation |
| Anderson | #22 Finalization of Baseline Testing |

## Review:

| Owner | Task Number & Title | Status | Comments |
|---|---|---|---|
| Aidan | #10 Algorithm Discussion: Whale Optimization | Completed | |
| Aidan | #11 Algorithm Discussion: Horse Herd | Completed | |
| Katelyn | #12 Algorithm Discussion: Binary PSO with Mutation Operator | Completed | Approval from sir. Next step, creating the baseline code. |
| Katelyn | #13 Algorithm Discussion: Grey-Sail Fish | Completed | Approval from sir. Next step, creating the baseline code |
| Anderson | #14 Algorithm Discussion: Hybrid Squirrel–Dragonfly | Completed | |
| Aidan | #15 Baseline Code: Whale Optimization | Completed | |
| Aidan | #16 Baseline Code: Horse Herd | Completed | |
| Katelyn | #17 Baseline Code: Binary PSO with Mutation Operator | Completed | Initial code completed. Updated to include Smote and K-fold. |
| Katelyn | #18 Baseline Code: Grey-Sail Fish | Completed | Initial code completed. Updated to include Smote and K-fold. |

| | | | |
|---|---|---|---|
| Anderson | #19 Baseline Code: Hybrid Squirrel–Dragonfly | Completed | |
| Aidan, Anderson, Katelyn | #20 Version One of the Algorithm | Completed | |
| Anderson | #21 Data Compilation and Presentation | Completed | |
| Anderson | #22 Finalization of Baseline Testing | Completed | |

Retrospective: This sprint went well from an academic perspective. All group members completed their literature reviews along with a solid explanation and understanding of their research papers. The research papers code was added as baseline tests to test against the cattle corral algorithm.

# Sprint Number 3

## Planning:

| Owner | Task Number & Title |
|---|---|
| Aidan, Anderson and Katelyn | #23 Mid-Semester Presentation Slides. |
| Anderson | #24 Implementing V1 of Cattle Corral Algorithm |
| Anderson and Aidan | #25 Improvements to V1 of Algorithm |
| Katelyn | #26 Abstract: Draft |
| Anderson | #27 Non Population Meta heuristic algorithm Implementation (Simulated Annealing) |

## Review:

| Owner | Task Number & Title | Status | Comments |
|---|---|---|---|
| Aidan, Anderson | #23 Mid-Semester Presentation Slides. | Completed | Positive feedback. |

| | | | |
|---|---|---|---|
| and Katelyn | | | |
| Anderson | #24 Implementing V1 of Cattle Corral Algorithm | Completed | Positive feedback. Decide on the approach to refine the algorithm. Refine explanation of algorithm. |
| Anderson | #25 Improvements to V1 of Algorithm | Completed | |
| Katelyn | #26 Abstract: Draft | Completed | Draft was submitted, positive feedback received. Final version attached to document. |
| Anderson | #27 Non Population Meta heuristic algorithm Implementation (Simulated Annealing) | | |

Retrospective:

We had the mid semester presentation, and it went well. We identified the theory behind our algorithm, implemented and tested it. We showed the initial results and made some improvements.

# Sprint Number 4

## Planning:

| Owner | Task Number & Title |
|---|---|
| Aidan | #28 Introduction Draft |
| Anderson | #29 Methodology Draft |
| Anderson | #30 Evaluating Runtime of Implemented Algorithms |
| Aidan, Anderson, Katelyn | #31 Individual Peer Evaluation Forms |
| Aidan, Anderson, Katelyn | #32 Project Report |
| Aidan, Anderson, Katelyn | #33 Sprint Report |
| Katelyn | #34 Final Presentation |

## Review:

| Owner | Task Number & Title | Status | Comments |
|---|---|---|---|
| Aidan | #28 Introduction Draft | Completed | |
| Anderson | #29 Methodology Draft | Completed | |
| Anderson | #30 Evaluating Runtime of Implemented Algorithms | Completed | |
| Aidan, Anderson, Katelyn | #31 Individual Peer Evaluation Forms | Completed | |
| Aidan, Anderson, Katelyn | #32 Project Report | Completed | |
| Aidan, Anderson, Katelyn | #33 Sprint Report | Completed | |
| Aidan, Anderson, Katelyn | #34 Final Presentation | Completed | Reviewed questions and suggestions made/asked from Mid-semester presentation to improve final presentation. |

Retrospective:

We had our final year presentation and final write ups. All documents were completed and submitted in a timely matter.  Overall, the project was enjoyable and as a group we performed really well.

# *A Novel Feature Selection Algorithm Inspired by Corralling Cattle Optimization for Fake News Detection*

Ramkissoon Amit
*Department of Computing and Information Technology*
*University of the West Indies*
St.Augustine, Trinidad and Tobago
amit.ramkissoon@uwi.sta.edu

Singh Anderson
*Department of Computing and Information Technology*
*University of the West Indies*
St.Augustine, Trinidad and Tobago
ando6703@gmail.com

Ramphal Katelyn
*Department of Computing and Information Technology*
*University of the West Indies*
St.Augustine, Trinidad and Tobago
katelyn.ramphal@gmail.com

Nagaur Aidan
*Department of Computing and Information Technology*
*University of the West Indies*
St.Augustine, Trinidad and Tobago
aidannagaur28@gmail.com

*Abstract*— **Feature selection is a crucial process in machine learning that enhances model performance by identifying the most relevant features. This study introduces a novel feature selection algorithm, Corralling Cattle Optimization (CCO), inspired by the symbiotic relationships observed between cattle, egrets, and flies. The proposed algorithm integrates global exploration and local exploitation strategies and is evaluated against established methods (filter, wrapper, embedded) and other nature-inspired algorithms (Binary PSO, G-SFO, WOA, HOA, HS-DSO). Evaluation metrics include AUC, F1-score, accuracy, precision, sensitivity, specificity, FPR, FNR, MCC, and selected feature count (n-feat). Experimental results demonstrate that CCO achieves superior performance, notably with the highest AUC (0.979) and precision (0.999), while maintaining computational efficiency. This highlights CCO's potential for efficient, real-world application in spam and fake news detection tasks.**

*Keywords— Feature selection, corralling cattle optimization, metaheuristics, fake news detection, machine learning*

## I. INTRODUCTION

Feature selection directly influences machine learning performance by identifying informative features and discarding irrelevant ones. Traditional methods (filter, wrapper, embedded) often struggle with high-dimensional data, computational inefficiency, and local optima. Nature-inspired metaheuristics, such as Particle Swarm Optimization (PSO), Whale Optimization Algorithm (WOA), Horse Herd Optimization (HOA), and Squirrel-Dragonfly Hybrid Optimization, offer promising alternatives by providing global search capabilities.

This paper introduces Corralling Cattle Optimization (CCO), inspired by cattle-herd behaviours involving egrets and flies. CCO incorporates a dual-mechanism strategy combining global exploration (egrets) and local exploitation (flies). Contributions include developing the CCO algorithm, simultaneously optimizing feature selection and hyperparameters, and evaluating it against established algorithms. The paper is structured as follows: Section II reviews related works; Section III describes methodology; Section IV outlines experimental results; Section V presents discussions, and Section VI concludes the paper.

## II. RELATED WORKS

### A. Literature Review

Feature selection significantly improves spam and fake news detection performance. Zhang et al. (2014) introduced Binary PSO with mutation operators, enhancing classification accuracy by preventing premature convergence through increased diversity in feature selection. However, computational complexity limited its real-time application. Kadam and Rohokale (2022) proposed Grey-Sail Fish Optimization (G-SFO) combined with Capsule Networks for multi-objective feature selection, demonstrating high accuracy and versatility in handling diverse spam types, although scalability issues persisted due to computational overhead.

Shuaib et al. (2019) applied the Whale Optimization Algorithm (WOA) with Rotation Forest classifiers, significantly reducing false positives compared to traditional methods such as Naïve Bayes. Despite impressive accuracy, the method was limited by dataset diversity and scalability concerns. Hosseinalipour and Ghanbarzadeh (2022) utilized the Horse Herd Optimization Algorithm (HOA), achieving superior accuracy and sensitivity by balancing exploration and exploitation through herd behavior. Nonetheless, its high computational demands presented scalability challenges.

Nithya and Sahayadhas (2023) introduced Hybrid Squirrel-Dragonfly Search Optimization (HS-DSO) in conjunction with ensemble learning techniques for fake news detection, effectively handling complex classification tasks. Although it demonstrated adaptability and high accuracy, risks of overfitting and substantial computational requirements limited its practicality in real-time scenarios.

Ramsubhag et al. (2025) explored efficient hyperparameter tuning using a modified Golden Section Search technique, achieving computational efficiency without sacrificing accuracy. While the method had broad applicability, its direct relevance to spam and fake news detection tasks was limited.

Recent literature highlights a consistent trend towards hybridization and ensemble approaches, emphasizing the need for methods capable of real-time detection, robustness to varied data sources, and computational efficiency. However, common challenges like scalability, computational complexity, and overfitting risks persist. The proposed Corralling Cattle Optimization (CCO) algorithm addresses these challenges by integrating effective initialization strategies and adaptive exploration-exploitation behaviours.

## III. METHODOLOGY

To address the challenge of jointly optimizing feature selection and model hyperparameters for binary classification, specifically in fake news detection, this study introduces a novel hybrid metaheuristic algorithm called Cattle Corral Optimization (CCO). Inspired by natural herd dynamics and symbiotic interactions observed among cattle, egrets, and flies, CCO integrates statistical measures of feature relevance with biologically inspired movement strategies to efficiently explore and exploit the search space. The following subsections detail each component of the CCO algorithm, highlighting its rationale, operational workflow, and key optimization strategies.

### A. Overview of Cattle Corral Optimization (CCO) Algorithm

The CCO algorithm models each candidate solution as a "cow" within a population-based optimization context. Each cow encodes both a feature subset (represented by a binary mask) and a configuration of hyperparameters for a Gradient Boosting Classifier (GBC). The primary objective is to identify the most informative subset of features alongside their corresponding hyperparameters, thereby maximizing predictive performance. Specifically, the performance is assessed through a weighted combination of key evaluation metrics, particularly the Area Under the Receiver Operating Characteristic Curve (AUC) and F1 score.

To guide this complex optimization task, CCO utilizes two distinct update strategies inspired by biological behaviors:

- Egret Update (Global Exploration): Simulates the broader exploratory behavior of egrets as they navigate towards cattle-rich areas, effectively helping the algorithm explore diverse regions within the solution space.

- Fly Update (Local Exploitation): Mimics the localized random movements of flies around cattle, fine-tuning candidate solutions within promising, already-explored regions to enhance precision and convergence.

### B. *Initialization Strategy Using Statistical Feature Ranking*

An innovative aspect of the CCO methodology lies in its domain-informed initialization strategy. To leverage domain knowledge effectively, each candidate cow is initialized with a bias towards features identified as highly relevant based on statistical criteria. Two measures are utilized:

1. Weighted Pearson Correlation Coefficient

This metric assesses the linear relationship between each feature and the target class, factoring in the importance or weight of observations. The weighted correlation is calculated as follows:

$$Weight\ Correlation = \frac{\sum w_i\,(x_i - \bar{x}_w)(y_i - \bar{y}_w)}{\sqrt{\sum w_i\,(x_i - \bar{x}_w)^2} \cdot \sqrt{\sum w_i\,(y_i - \bar{y}_w)^2}} \tag{1}$$

Where denotes $w_i$ the weight of the $i^{th}$observation, $x_i , y_i$ are values of the feature and target for the $i^{th}$ instance and $\bar{x}_w$ and $\bar{y}_w$ are weighted averages of feature x and y respectively.

2. Mutual Information (MI)

Mutual Information quantifies the dependency between features and class labels by measuring the amount of uncertainty reduction provided by knowing the feature value. The MI for two discrete random variables X and Y is expressed as:

$$I\,(X;Y) = \sum_x \sum_y p(x,y) \cdot \log(\frac{p(x,y)}{p(x) \cdot p(y)}) \tag{2}$$

Where $p(x,y)$ is the joint probability of the distribution of X and Y and $p(x)$ and $p(y)$ are the marginal probability of X and Y respectively

These two scores are subsequently normalized to ensure comparability and combined into a unified relevance score for each feature, utilizing min-max normalization:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min\,(x)} \tag{3}$$

The combined feature relevance score is then computed as a weighted average:

$$Combined\,Score = w_e \cdot NormCorr + w_m \cdot NormMI \tag{4}$$

Where $w_e, w_m$ are the user-defined weights for correlation and mutual information, $NormCorr$ and $NormMI$ are the normalized weighted correlation and mutual information respectively.

### C. *Egret Update Strategy (Global Exploration)*

The Egret update mechanism is central to promoting a broader search of the solution space. Egrets move toward promising areas (represented by the global best solution), but their trajectories include random perturbations to maintain diversity:

$$x_{new} = x + r \cdot (x_{best} - x) + \delta \tag{5}$$

Where $x$the current candidate is position (solution vector), $x_{best}$ is the best known candidate (global best), $r \in [0,0.7]$ is the random jump factor and $\delta \in [-0.03,0.3]$ is the small random noise vector which introduces diversity.

### D. *Fly Update Strategy (Local Exploitation)*

The Fly update strategy emphasizes local refinement within regions identified as promising. Flies conduct smaller, random movements around the current best candidate, fine-tuning the solutions:

$$x_{new} = x + \eta + \alpha\,(x_{best} - x) \tag{6}$$

Where $\eta \in [-0.15,0.15]$ which represents the local random movement, $\alpha$ is 0.4 representing the attraction weight toward the best candidate, $x$ is the current candidate position (solution vector) and $x_{best}$ is the best known candidate (global best).

### E. *Fitness Evaluation and Adaptive Convergence Criterion*

Fitness evaluation of each candidate cow involves training a Gradient Boosting Classifier with the selected features and hyperparameters, followed by performance assessment using cross-validation. The evaluation metrics include AUC, F1-score, accuracy, precision, sensitivity, specificity, MCC, FPR, FNR, and the number of selected features (n-feat).

An adaptive stopping criterion enhances computational efficiency. If the best fitness value (primarily evaluated via AUC) does not improve significantly after a defined number of iterations (patience threshold), the algorithm terminates early to conserve resources and avoid unnecessary computation.

### F. *Complete Operational Flow of the CCO Algorithm*

Fitness evaluation of each candidate cow involves training a Gradient Boosting Classifier with the selected features and hyperparameters, followed by performance assessment using cross-validation. The evaluation metrics include AUC, F1-score, accuracy, precision, sensitivity, specificity, MCC, FPR, FNR, and the number of selected features (n-feat).

An adaptive stopping criterion enhances computational efficiency. If the best fitness value (primarily evaluated via AUC) does not improve significantly after a defined number of iterations (patience threshold), the algorithm terminates early to conserve resources and avoid unnecessary computation.

The operational steps of the CCO algorithm are summarized below:

- Step 1: Compute the weighted correlation and mutual information scores for all features.
- Step 2: Normalize and combine these scores into a single relevance ranking, introducing slight perturbations to break ties.
- Step 3: Initialize the candidate cow population using the relevance-biased selection of top-ranked features and randomized hyperparameter values.
- Step 4: Evaluate the fitness of each candidate solution using cross-validation with the Gradient Boosting Classifier.
- Step 5: Update the global best solution based on improvements in AUC and reset or increment a patience counter.
- Step 6: For each candidate, probabilistically apply either Egret or Fly updates based on a defined criterion to balance exploration and exploitation.
- Step 7: Clip positions of new solutions to remain within feasible parameter bounds.
- Step 8: Repeat Steps 4-7 until the adaptive stopping criterion is met.
- Step 9: Return the final optimized feature set and hyperparameters corresponding to the best solution.

### G. *Egret Update Strategy (Global Exploration)*

The CCO algorithm is implemented using Python, leveraging popular machine learning libraries such as Scikit-learn for model training and evaluation, and NumPy for numerical computations. Experimental procedures, including feature extraction, normalization, and embedding generation, are conducted using Sentence Transformers (specifically the pre-trained model all-MiniLM-L6-v2), ensuring the algorithm's applicability to high-dimensional textual datasets relevant to fake news detection.

Furthermore, data imbalance issues are addressed using the Synthetic Minority Over-sampling Technique (SMOTE), and model generalizability is ensured through rigorous k-fold cross-validation. This comprehensive methodological framework establishes CCO's suitability for efficient, robust, and accurate feature selection and hyperparameter optimization in complex classification scenarios.

## IV. Experimental Results

### A. *Experimental Design and Data Preprocessing*

Effective preprocessing is crucial for the performance of machine learning models, especially in high-dimensional classification tasks such as fake news detection. To ensure optimal model performance, the following preprocessing steps were implemented:

- **Text Data Representation:** The Sentence Transformers framework was utilized, specifically leveraging the pre-trained model *all-MiniLM-L6-v2*. This compact yet powerful transformer generates dense, context-aware embeddings (384-dimensional vectors) for textual data, significantly enhancing the model's capability to identify nuanced semantic patterns critical in fake news classification.
- **Temporal Feature Extraction**: Categorical variables were encoded using frequency encoding, particularly beneficial for handling high-cardinality categories. The relative frequency of each categorical value was computed to retain distributional information effectively.
- **Categorical Variable Encoding:** Categorical variables were encoded using frequency encoding, particularly beneficial for handling high-cardinality categories. The relative frequency of each categorical value was computed to retain distributional information effectively.
- **Class Label Encoding**: Class labels were transformed into ordinal integers to meaningfully represent the inherent ranking or significance among different categories.
- **Feature Scaling:** Standardization was applied to all numerical features to ensure uniform scaling, preventing features with larger magnitude from dominating model training. Each feature was scaled to have a mean of 0 and a standard deviation of 1.
- **Handling Imbalance:** The Synthetic Minority Over-Sampling Technique (SMOTE) was employed to generate synthetic instances of the minority class, effectively balancing the dataset and improving classifier sensitivity.
- **Validation Strategy:** K-fold cross-validation (k=5) was adopted to robustly evaluate the generalizability and reliability of each model, reducing variance and ensuring consistent performance across different data subsets.

### B. *Comparative Performance Analysis*

This study compares the proposed CCO algorithm against five well-established metaheuristic algorithms: Hybrid Squirrel Dragonfly Optimization (HS-DSO), Whale Optimization Algorithm (WOA), Horse Herd Algorithm (HHA), Grey-Sail Fish Optimization (G-SFO), and Particle Swarm Optimization (PSO). Performance comparisons were made using several key evaluation metrics: area under the curve (AUC), f1 score, accuracy, precision, false positive rate (FPR), false negative rate (FNR), Matthews's correlation coefficient (MCC), execution time.

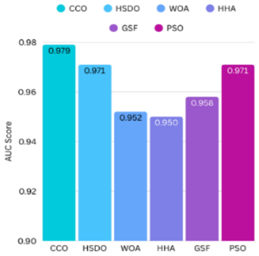


*Fig. 1. Comparison of AUC scores for all evaluated algorithms.*

As shown in Fig. 1, the CCO algorithm achieved the highest AUC score (**0.979**), outperforming all baseline algorithms, thus demonstrating exceptional discriminative capability between fake and genuine content.

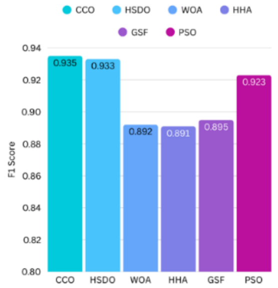


*Fig. 2. Comparison of F1 scores across algorithms.*

In Fig. 2, the CCO algorithm demonstrated the highest F1 score (**0.935**), indicating its superior balance between precision and recall, essential in classification tasks where misclassification costs are high.

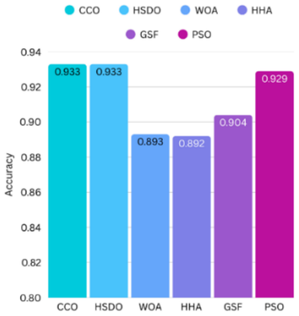


*Fig. 3. Accuracy comparison among evaluated algorithms.*

The accuracy results (Fig. 3) indicate that CCO and HS-DSO both achieved the top accuracy of 0.933, closely followed by PSO (0.929). These high accuracy rates underscore the reliability of the optimized models.

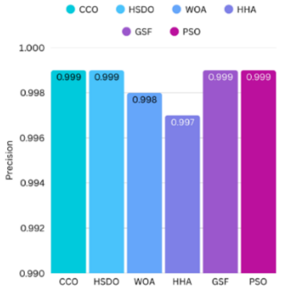


*Fig. 4. Precision scores for the evaluated algorithms*.

Fig. 4 highlights that CCO, HS-DSO, G-SFO, and PSO achieved nearly perfect precision scores (**0.999**), indicating a minimal likelihood of falsely classifying legitimate content as spam or fake.

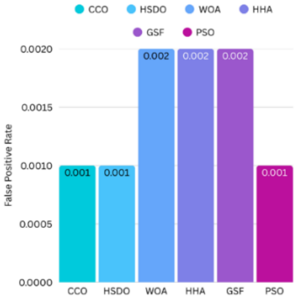


*Fig. 5. Comparison of False Positive Rates (FPR).*

In Fig. 5, CCO, HS-DSO, and PSO recorded the lowest false positive rates (**0.001**), affirming their capability to minimize false alarms and thus maintain high user trust in practical deployments.

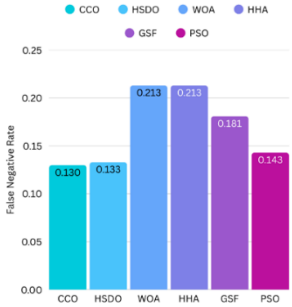


*Fig. 6. False Negative Rate (FNR) analysis.*

Fig. 6 shows the FNR across algorithms, with CCO having the lowest rate (**0.130**), indicating its superior capability in correctly identifying actual spam or fake news, thereby significantly reducing the risk of misinformation propagation.

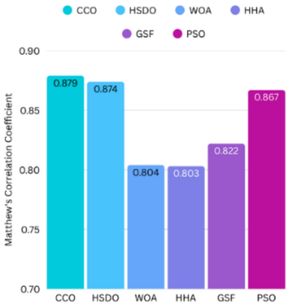


*Fig. 7. MCC values indicating balanced classification performance.*

The MCC results (Fig. 7) further confirm CCO's superior predictive balance (**0.879**), surpassing all other methods, thus highlighting its balanced performance across positive and negative classifications.

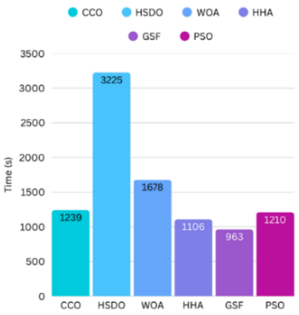


*Fig. 8. Execution time (seconds) across evaluated algorithms.*

In Fig. 8, the runtime analysis demonstrates that while G-SFO is fastest (**963 seconds**), CCO (**1239 seconds**) offers an optimal trade-off between computational efficiency and high predictive performance, outperforming HS-DSO (**3225 seconds**) and WOA significantly.

### *C. Summary of Performance Metrics*

**Table I** below summarizes key performance results, clearly displaying the comparative strengths and limitations across evaluated algorithms:

TABLE I. SUMMARY OF RESULTS

| | **CCO** | **HSDO** | **WOA** | **HHA** | **GSF** | **PSO** |
|---|---|---|---|---|---|---|
| **AUC** | 0.979 | 0.971 | 0.952 | 0.950 | 0.958 | 0.971 |
| **F1** | 0.935 | 0.933 | 0.892 | 0.891 | 0.895 | 0.923 |
| **Accuracy** | 0.933 | 0.933 | 0.893 | 0.892 | 0.904 | 0.929 |
| **Precision** | 0.999 | 0.999 | 0.998 | 0.997 | 0.999 | 0.999 |
| **FPR** | 0.001 | 0.001 | 0.002 | 0.002 | 0.002 | 0.001 |
| **FNR** | 0.130 | 0.133 | 0.213 | 0.213 | 0.181 | 0.143 |
| **MCC** | 0.879 | 0.874 | 0.804 | 0.803 | 0.822 | 0.867 |
| **Time (s)** | 1239 | 3225 | 1678 | 1106 | 963 | 1210 |

## V. DISCUSSION

The primary objective of this research was to develop and evaluate the performance of the proposed **Cattle Corral Optimization (CCO)** algorithm, especially in relation to addressing existing challenges in the fields of spam and fake news detection, with a particular emphasis on optimizing feature selection. The results clearly demonstrate that the CCO algorithm offers superior predictive performance across multiple critical metrics when benchmarked against both traditional and contemporary metaheuristic optimization algorithms, including Hybrid Squirrel-Dragonfly Optimization (HS-DSO), Whale Optimization Algorithm (WOA), Horse Herd Algorithm (HHA), Grey-Sail Fish Optimization (G-SFO), and Particle Swarm Optimization (PSO).

### *A. Interpretation of Key Results*

The effectiveness of the CCO algorithm is particularly evident from its leading performance in terms of the **Area Under the Curve (AUC)**, achieving an impressive value of **0.979.** This metric is critical in binary classification tasks because it directly reflects the algorithm's capability to correctly discriminate between spam (fake news) and legitimate content, thus confirming the algorithm's robustness and reliability. Moreover, the CCO algorithm also outperformed competing methods in terms of the **F1 score (0.935)** and achieved a high **accuracy (0.933).** The strong F1 score highlights the balanced trade-off between precision and recall—crucial in applications where false positives (erroneously flagged legitimate content) and false negatives (missed spam or fake news) have significant practical implications, such as cybersecurity or content moderation tasks.

An important differentiating factor among the tested algorithms was the **False Negative Rate (FNR)**. CCO achieved a particularly low FNR of **0.130**, outperforming most baseline methods, including WOA and HHA, which had higher FNRs of approximately **0.213**. This implies that the CCO algorithm is less likely to incorrectly classify genuine spam or fake content as legitimate, which is critically important for real-world tasks like misinformation management and spam filtering. Similarly, **the Matthews Correlation Coefficient (MCC)** for CCO was **0.879**, further confirming the strong predictive capability and reliability of the algorithm across both positive and negative classes.

### *B. Computational Efficiency*

While predictive performance is a primary consideration, computational efficiency remains equally vital, especially for real-time applications or resource-constrained environments. The CCO algorithm completed its execution in **1239 seconds**, placing it comfortably within a practical and feasible runtime window. Although not the absolute fastest algorithm tested (with GSF being faster at **963 seconds**), CCO demonstrated a strong balance between computational cost and predictive performance. Specifically, CCO was significantly more efficient than algorithms like HS-DSO, which required **3225 seconds**, making it impractical for rapid-response scenarios. This balance ensures CCO remains a viable choice in real-time or near-real-time applications such as online spam filtering or misinformation detection, where both speed and accuracy are imperative.

### *C. Contributions of Methodological Innovations*

The performance advantage of CCO can largely be attributed to several unique methodological innovations introduced in this study. Firstly, the domain-informed initialization strategy leveraging weighted correlation and mutual information significantly accelerated convergence towards high-quality solutions by incorporating prior feature relevance. This approach distinguishes CCO from typical metaheuristic algorithms, which usually start from entirely random feature subsets, requiring additional iterations to identify optimal or near-optimal solutions.

Furthermore, the distinctive dual-update mechanism (Egret and Fly updates) employed in CCO successfully maintained a balance between **global exploration** and **local exploitation**. The Egret updates allowed the algorithm to broadly search the feature-hyperparameter solution space and effectively avoid local optima, while the Fly updates finely adjusted promising solutions to achieve precise optimization. This integration of biologically inspired mechanisms helped CCO consistently outperform standard algorithms, showcasing the effectiveness of hybrid metaheuristic strategies.

Additionally, the adaptive stopping criterion based on fitness stagnation improved computational efficiency by dynamically halting the algorithm when further iterations provided negligible improvements. This not only conserved computational resources but also ensured that the optimization process remained effective and practical for diverse deployment scenarios.

### *D. Strengths and Limitations*

A notable strength of this study lies in the comprehensive comparative framework employed, which rigorously benchmarked CCO against a range of established optimization algorithms across multiple performance metrics. Furthermore, the use of robust data preprocessing steps, such as the Sentence Transformers model for text representation, frequency-based encoding of categorical variables, and SMOTE for handling class imbalance, substantially enhanced the reliability and generalizability of the findings.

However, the study also presents some limitations that warrant consideration. Firstly, while CCO demonstrates impressive performance for binary classification tasks, its scalability and effectiveness in multi-class or multi-label classification scenarios remain untested and thus uncertain. Additionally, although the computational cost of CCO was within acceptable limits, further reductions in runtime through parallelization or optimization of code could significantly enhance real-world applicability, particularly for large-scale data environments or high-throughput processing systems.

### *E. Future Direction*

Building upon the promising results of this research, several avenues for future exploration are identified:

- **Parallelization and Scalability:** Future research could explore parallel computing techniques to substantially reduce runtime, making CCO suitable for massive datasets or high-frequency classification tasks.
- **Multi-Class and Multi-Label Applications:** Extending the CCO framework to address more complex classification scenarios, such as multi-label content classification or categorizing misinformation into multiple distinct categories, could broaden its practical utility.
- **Robustness Under Adversarial Conditions:** Further testing of the algorithm's robustness to adversarial attacks, data poisoning, or noisy datasets would strengthen its resilience in hostile or uncertain environments.
- **Hyperparameter Refinement and Automation:** Incorporating automated hyperparameter tuning strategies such as Bayesian optimization or reinforcement learning-based approaches could further enhance the predictive power and ease-of-use of the CCO algorithm.
- **Real-Time Integration and Deployment:** Investigating the real-time applicability of the CCO algorithm in production-level spam filtering systems or live misinformation detection pipelines would demonstrate practical feasibility and inform further improvements.

Overall, the findings validate the effectiveness and efficiency of the CCO algorithm, emphasizing its potential as a practical and powerful solution for feature selection and hyperparameter optimization tasks in intelligent content classification systems.

## VI. Conclusion

This paper presented Corralling Cattle Optimization, a novel algorithm effectively balancing exploration and exploitation for feature selection. With superior results in critical metrics like AUC (0.979) and F1-score (0.935), CCO demonstrates its capability in enhancing spam and fake news detection. Future enhancements include improving computational efficiency, hyperparameter tuning, multi-class adaptation, and real-time applicability.

## Acknowledgment

. We would like to express our heartfelt gratitude to the individuals and organizations listed below for their invaluable support and contributions throughout this project. We thank our supervisors, Dr. Kris Manohar and Dr. Amit Ramkissoon, for their expert guidance, insightful feedback, and continual encouragement, which greatly improved the quality of this report. We extend our appreciation to the faculty and staff of the Department of Computer Science for providing the necessary resources and support required to conduct this study. Additionally, we thank our colleagues and peers for their helpful discussions and valuable suggestions during the development of the Cattle Corral Optimization (CCO) algorithm. Finally, we are profoundly grateful to our families and friends for their unwavering support and understanding throughout this journey.

## References

[1] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary Particle Swarm Optimisation for feature selection: A filter based approach," in *2012 IEEE Congress on Evolutionary Computation*, 2012, pp. 1–8

[2] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowledge-Based Systems*, vol. 140, pp. 103–119, 2018.

[3] A. Hosseinalipour and R. Ghanbarzadeh, "A novel approach for spam detection using Horse Herd Optimization Algorithm," *Neural Computing and Applications*, vol. 34, pp. 13091–13105, 2022, doi: 10.1007/s00521-022-07148-x.

[4] A.-D. Li, B. Xue, and M. Zhang, "Improved binary Particle Swarm Optimization for feature selection with new initialization and search space reduction strategies," *Applied Soft Computing*, vol. 106, p. 107302, 2021.

[5] S. H. Nithya and A. Sahayadhas, "Meta-heuristic searched-ensemble learning for fake news detection with optimal weighted feature selection approach," *Data & Knowledge Engineering*, vol. 144, p. 102124, 2023, doi: 10.1016/j.datak.2022.102124.

[6] D. Ramsubhag, K. Baboolal, and P. Hosein, "Efficient hyper-parameter tuning for the kappa regression algorithm," *Journal of Computational Science*, vol. 18, p. 102342, 2025.

[7] K. V. Samarthrao and V. M. Rohokale, "A hybrid meta-heuristic-based multi-objective feature selection with adaptive Capsule Network for automated email spam detection," *International Journal of Intelligent Robotics and Applications*, vol. 6, no. 3, pp. 497–521, 2022, doi: 10.1007/s41315-021-00217-9.

[8] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "IoT malicious traffic identification using wrapper-based feature selection mechanisms," *Computers & Security*, vol. 94, p. 101863, 2020.

[9] M. Shuaib, S. M. Abdulhamid, O. S. Adebayo, O. Osho, I. Idris, J. K. Alhassan, and N. Rana, "Whale Optimization Algorithm-based email spam feature selection method using Rotation Forest algorithm for classification," *SN Applied Sciences*, vol. 1, article no. 390, 2019. [Online].

[10] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014, doi: 10.1016/j.knosys.2014.03.015. [Online].