International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

A Modified Binary Grey Wolf Optimization for Feature Selection Using Elite Wolf in Unstructured Data Stream

Suman R. Tiwari 1*[1], Kaushik K. Rana[2],Viral H. Borisagar[3]

Submitted: 25/05/2024 **Revised**: 27/05/2024 **Accepted**: 02/06/2024

Abstract: Stream clustering poses challenges in feature selection due to data dynamics, variety, and a lack of labels in incoming data streams. While existing methods rely on labelled data, assuming structure in heterogeneous, unlabeled streams is unrealistic. To address this, we introduce a novel feature selection method, modified binary grey wolf optimization for stream feature selection (MBGWOSFS) using elite wolf, utilizing Evolutionary algorithm for unsupervised learning in streaming environments. Our novel feature selection method, aims to enhance clustering performance by selecting relevant features from unstructured data streams. Evaluation using internal metrics like Dunn Index, Davies-Bouldin Index, Calinski-Harabasz Index, and Silhouette Score, separation and compactness demonstrates that MBGWOSFS outperforms traditional methods by providing effective feature selection without relying on labelled data or predefined structures. With varying feature counts and high Dunn indices ranging from 57.846 to 72.7538, the method excels in cluster separation, reinforcing strong data similarity within clusters with Silhouette scores between 0.0324 and 0.047. Further, the well-balanced cluster quality, reflected in DB index and CH index values of 2.631 to 3.264 and 0.3688 to 0.43 respectively, showcases the adaptability and superior effectiveness of MBGWOSFS in text stream.

Keywords: *cosine proximity, landmark window, session window, tumbling window*

1. Introduction

In the realm of data mining, text clustering plays a crucial role in uncovering hidden patterns and insights within text data. Clustering involves grouping text based on their similarities and distinguishing them into distinct groups based on their unique properties[1], [2]. An effective clustering algorithm should aim to create compact and well-separated clusters.

A stream is an infinite, ordered sequence of instances that flows continuously[2]. Because of the enormous amount, speed, and diversity of textual data being processed, text stream clustering in particular presents a substantial difficulty. Incoming data streams present additional challenges when it comes to applying clustering because of their sequential nature, inability to be stored at single place, one-time readability, and one-time availability[1], [2], [3], [4].

Moreover, the sources of incoming features can change over time, leading to the need to address concept drift [5] and feature drift [4] in a streaming environment. Concept drift is a situation in which data from different sources arrive, resulting in the emergence of new clusters[2]. On the other hand, feature drift[6] occurs when the source of the data remains unchanged, but features evolve over time, making it challenging to directly apply traditional clustering approaches in streaming environments .

It is important to use unstructured data for clustering because a significant amount of data today is in unstructured formats like text, audio, and images. By employing clustering algorithms on unstructured data, valuable insights can be extracted, patterns can be identified, and relationships can be uncovered, leading to more accurate and meaningful analysis of the data.

In stream data processing, a direct approach to process incoming stream is not feasible. Instead, different methods are employed, such as the online-offline phase[2]and window-based approach[7]. The window-based approach, a popular technique, modifies conventional batch methods for stream processing by segmenting the incoming stream into smaller batches known as window. These windows provide snapshots of the dataset over which queries are evaluated periodically. Windows can be categorized as either count-based, where parameters like window length and slide interval are defined in terms of elements, or time-based, where these parameters are defined in terms of time. There are various types of windows such as sliding window[2],

[1] Research Scholar , Gujarat Technological University & Lecturer ,Computer Engg. ,R.C.Technical Institute , Ahmedabad, India -382424
Email: srt.rcticomputer@gmail.com
ORCID ID : 0000-0001-8769-644X
[2] Assistant professor , Computer Eng. , VGEC Chandkheda , Ahmedabad , India - 382424
Email :kkr@vgecg.ac.in
ORCID ID : 0000-0002-6251-343X
[3] Assistant professor , Computer Eng., VGEC Chandkheda , Ahmedabad ,India - 382424
Email : viralborisagar@gmail.com
ORCID ID : 0000-0002-4816-7374
** Corresponding Author Email: srt.rcticomputer@gmail.com*

tumbling window, landmark window[2], and session window. In our proposed work, we utilize a time-based sliding window for efficient execution and processing of stream data.

Feature selection is crucial in pre-processing text data[8] particularly in text stream clustering, to reduce dimensionality and improve clustering performance by selecting most informative features from original feature set [9], [10]. Feature selection is recognized as computationally challenging problem for optimization , because it falls in the category of NP-Hard problem [3], [8], [10], [11]. It is not ideal to search entire feature space sequentially for large dimensions [12]. There are various methods available in the feature selection area, namely filter, wrapper, and embedding or hybrid approaches[8], [13], [14], [15], [16]. Filter methods assess feature quality based on relevancy scores and often require the pre-specification of the number of features [4]. Being independent of machine learning algorithms, filter methods are quicker compared to other methods. On the other hand, wrapper methods utilize clustering or classification accuracy as a feedback to select relevant features, making them slower than filter methods but more precise. Typically, forward selection or backward elimination techniques are implemented in wrapper-based feature selection methods. Embedded methods, also known as hybrid methods, combine aspects of both filter and wrapper approaches to enhance the selection process.

However, traditional feature selection methods struggle in streaming environments due to the dynamic nature because it assume that the entire feature space is available in advance and that the characteristics of the feature remains unchanged over time[13], [17], [18] [19]. Hence drawback of existing traditional feature selection method within data stream prompt us towards the exploration of evolutionary based techniques like Particle Swarm Optimization[12], [16], Genetic Algorithms, Ant Colony Optimization [7], [14], [20], Artificial Bee colony and Grey Wolf Optimization [8], [10], [16], [21], [22], [23], [24], for more efficient and accurate feature selection.

Evolutionary algorithm excels in efficiently exploring vast feature spaces, making them well-suited for processing large amounts of data in real-time. Grey Wolf Optimization, in particular, has shown promise in identifying relevant features in datasets. By harnessing the adaptive and self-organizing capabilities of evolutionary algorithm, researchers can efficiently tackle feature selection challenges in streaming data scenarios and enhance clustering accuracy.

In this research, the Modified Binary Grey Wolf for Stream Feature Selection (MBGWOSFS) technique is introduced for unstructured unsupervised streaming data. The paper is structured as follows:

- Section 2: Provides an in-depth analysis of existing research in the field of feature selection within a streaming context.
- Section 3: Explores the details of Standard Binary Grey Wolf Optimization (GWO) for better understanding.
- Section 4: Describes the comprehensive methodology employed, introducing the proposed MBGWOSFS algorithm and detailing the evaluation metrics used for unsupervised scenarios , Additionally, it discusses the incorporation of k-means clustering, a dynamic approach that autonomously determines the optimal value of the parameter 'k' for streaming data.
- Section 5: Discusses the Experimental Setup and presents the outcomes and results.
- Section 6: Concludes the study by summarizing the findings and suggesting avenues for future research and development.

2. Releated Work

This section looks at what other studies have discovered in selecting features for stream clustering. It helps establish a strong base for the current study by examining past research on how features are selected in stream clustering. By analyzing earlier studies, including what they found, how they did their research, the problems they encountered, and where more research is needed, this paper aims to provide context for the new study in area of feature selection for stream environment without labelled data and unstructured format .

The majority of current feature selection techniques for stream data are supervised[17], with unsupervised methods following a supervised framework for assessing performance and leveraging structured format. After reviewing the literature, it is evident that many researchers have employed evolutionary algorithms to tackle a range of feature optimization problem.

Yeoh et al.[5] has proposed the OpStream algorithm for stream clustering by harnessing metaheuristic optimization. They have collected incoming stream into landmark window and then OpStream is applied. Whale Optimization Algorithm is used to generate the initial centre points within their proposed work , ensuring efficient clustering performance for dynamic streams.

Fahy & Yang[4] has introduced a dynamic feature mask(DFM) approach for clustering high dimensional streams. They conducted a comparison between dynamic feature mask, static feature mask techniques and no feature mask technique. The dynamic feature mask method demonstrated superior performance compared to the static and no mask methods. In dynamic approach, the feature mask is updated each time a drift is detected in the data stream. To select the masked features, they utilized variance, Laplacian score, and a multi-cluster feature selection method.

In the realm of feature selection, Dazhi Wang et al. [10] utilized the Binary Grey Wolf Optimizer to enhance classification tasks by incorporating a population adaption strategy. Their methodology includes adaptive individual update procedures, a head wolf fine-tuning mechanism, and the ReliefF filter-based method for calculating feature weights, ensuring improved exploitation ability and convergence speed.

Zhang Li et al. [24]proposed the feature selection mechanism using velocity guided Grey Wolf optimization algorithm, integrating adaptive weights and Laplace operators. They bring a fresh perspective with their innovative method, incorporating dynamic adaptive weighting mechanisms, a position update formula based on velocity with individual memory function to improve local exploration, along with utilizing a Laplace crossover strategy to increase population diversity and address challenges with local optima.

In a unique blend of optimization techniques, El-Hasnony et al. [25] introduced a binary variant of the wrapper feature selection combining Grey Wolf Optimization and Particle Swarm Optimization. Leveraging a k-nearest neighbor classifier with Euclidean distance metrics, their method integrates a tent chaotic map, sigmoid function, and innovative strategies to tackle local optima problems and craft binary search spaces suitable for feature selection challenges.

Yan Xuyang et al.[17] proposed a novel approach to estimate feature stream density distributions, introducing dynamic clustering techniques to explore feature redundancy efficiently. Their methodology is an unsupervised online technique that maximizes feature relevance and reduces redundancy, ensuring minimal repetition in the extraction of critical features from continuous stream.

Almusallam Naif et al.[18] aimed to find representative streaming features without requiring data labeling by developing a streaming feature-specific unsupervised feature selection method. Their solution expands the k-means clustering algorithm to efficiently determine the significance of newly arriving features.

Fahy Conor et al. [20] Introduced the Fast Density Ant Colony Stream Clustering Algorithm, a probabilistic approach for clustering based on tumbling windows. This method focuses on pinpointing dense areas in the feature space that are delineated by low-density zones, allowing for effective and precise clustering.

Shuliang Xu et al. [26] Presented the Self-Adaption Neighbourhood Density Clustering (SNDC) technique designed for handling mixed data streams with evolving concepts. By utilizing mapping techniques for categorical attributes and non-linear dimensionality reduction, SNDC automatically selects optimal initial clusters and efficiently processes and clusters data points based on similarity and density measures.

Lastly, Wong Raymond et al.[12] Suggested the Accelerated PSO swarm-based feature selection technique tailored for mining big data streams. This method achieves enhanced analytical accuracy while maintaining reasonable processing times, making it a valuable tool for real-time big data analysis.

3. Grey Wolf Optimization

Grey wolf optimization is recent metaheuristic swarm intelligence method Proposed by Seyedali Mirjalili in 2014[27] , which is inspired by social structure and behaviour of grey wolf optimization.

GWO mimics the hierarchical structure of a wolf pack [16], [27]and their hunting strategy is used to solve optimization problems , where each wolf represents a potential solution. In hierarchy of grey wolf optimization top most wolf is considered as alpha(α), followed by Beta (β) and delta (δ) wolf. The alpha(α) wolf is considered the most dominant and powerful within the pack's structure, hence it is considered as most powerful solution followed by beta(β) wolf.

The Beta wolf is considered as second-best solution and assist alpha wolf in decision making. They are also considered powerful wolves and take charge in the absence of the alpha wolf. The Delta(δ) and Omega(ω) wolves are regarded as weaker in the hierarchy, serving roles such as caretakers, scouts, and the elderly wolf.

In grey wolf optimization ,each wolf updates its position concerning the alpha, beta, and delta wolves . It optimizes the process with three steps : encircling the pray , hunting and attacking the pray[27]. Mathematical equations for encircling the pray is shown in (1) and (2)

$$\vec{D} = \left|\overrightarrow{C} * \vec{X}_p(t) - \vec{X}\right| \tag{1}$$

$$X(t+1) = \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D} \tag{2}$$

$$\vec{X}(t+1) = (X_1 + X_2 + X_3)/3 \tag{3}$$

The parameters $\vec{A}$ and $\vec{D}$ are calculated $\vec{A} = 2 \cdot \vec{r}_1 \cdot \overline{\vec{a}} - \vec{a}$ and $\vec{C} = 2 \cdot \vec{r}_2$. Each wolf updates the next position with use of (3) , where $\overrightarrow{X_1}, \overrightarrow{X_2} and \overrightarrow{X_3}$ are current position of alpha , beta and delta wolf respectively, parameter t indicates the current iteration and $\overrightarrow{X_p}$ indicates the position of target wolf.

4. Proposed Methodology

The framework we have developed is the Modified Binary Grey Wolf Optimization for stream feature selection (MBGWOFS) using elite wolf for unsupervised unstructured stream. This extends Grey Wolf Optimization to effectively select features in a streaming environment, determining the optimal features for each incoming streaming window. The entire process of proposed method is visually represented in Fig. 1.

Data extraction, which involves gathering and transforming raw data into a structured format necessary for analysis. Data extraction is essential as it allows us to obtain the specific dataset required for tasks such as clustering. We utilized the Apache Spark library to extract data and transform it into a structured format. By reading the stream into a streaming data frame, we were able to efficiently process and analyze the data in a structured manner.

We have implemented a non-overlapping time-based sliding window approach for handling incoming data streams. To ensure that each window's elements are unique and not repeated in subsequent windows, we have set the slide interval and window size to be the same. Every sliding window is processed sequentially to uphold data continuity. The initial window serves as the base window to train the model.

Given the heterogeneous nature of the data stream formats, direct clustering or conversion of stream is not feasible. Therefore, data must be converted into appropriate format to enable the necessary pre-processing steps for clustering.

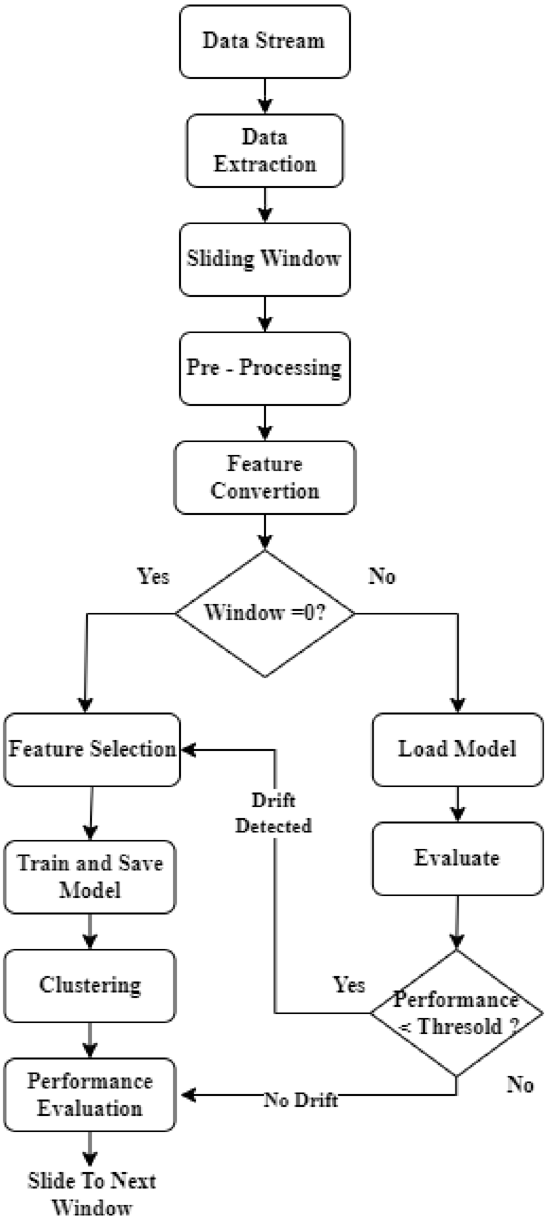

Fig. 1. Flow of Proposed Methodology

The dataset underwent preprocessing steps, including converting text to lowercase, removing stop words, special symbols, digits, and words shorter than three characters. Finally, lemmatization was applied to standardize the text into its root form.

Text features need to be converted into numerical format in order to be processed by machine learning algorithms. By representing text data numerically, the algorithms can effectively analyze and derive patterns from the textual information. We have converted textual information using cosine proximity measure.

After the conversion of the stream window into a numerical format, the proposed feature selection method MBGWOSFS as visualized in Fig. 2. is applied. For the initial window, a model is trained and saved for future use. In subsequent windows, the existing model is utilized for evaluation. If the performance evaluation deteriorates than a predetermined threshold, indicating a drift in the incoming stream, the model is retrained using the newly available features.

In a streaming environment, simple K-means encounters challenges due to the need to predefine number of clusters. As the stream source fluctuates constantly, it becomes a dynamic problem to figure out how many clusters is ideal for K-means. To overcome this, the silhouette score from the current window stream is utilized to dynamically adjust the number of clusters. This adaptive process allows for the potential growth or dissolution of clusters based on the evolving nature of the stream.

4.1 Feature Selection

Feature selection is a binary problem, so we focused on the binary version of Grey Wolf Optimization, which has several advantages over other swarm intelligence techniques due to having very few parameters and not requiring derivative information[8]. However, the main drawback of Grey Wolf Optimization is that it may get trapped in local optima, potentially hindering the discovery of the optimal solution. To address this limitation and ensure exploration of the entire search space, we incorporated the concept of random scaling and proposed modified grey wolf optimization for stream feature selection as in Fig. 2.

In our proposed MBGWOSF, we merged the least powerful hierarchy into one, utilizing three categories of wolves for feature optimization. Our aim was to reduce the number of parameters needed for Grey Wolf Optimization and improve accuracy. The goal of the Grey Wolf Optimization parameter is to specify the search space for the Grey Wolf.

To identify the proper search space, we derived the random scaling factor and selected the parameter 'a' for each wolf. Instead of using (1), (2), and (3) for the calculation of the next steps for each wolf, we randomly generated the value of the parameter 'a' between the minimum and maximum scaling numbers. This value was used to generate the position for the current wolf. Each wolf contains a set of features with values of either '0' or '1', where '1' denotes the selection of the feature and '0' suggests the absence of the feature in the wolf's set. To prevent the issue of local optima, we implemented a different strategy to select the set of features for the wolf based on whether they were in the exploration or exploitation phase.

During the exploration phase, a wolf's scaling factor is generated based on the parameter 'a', which determines the position of the new wolf. The fitness of the wolf is calculated using (4) and compared against the fitness of each type of wolf. If the fitness surpasses that of the alpha wolf, then the alpha wolf adopts the position of the current wolf, and the elite feature vector is updated.

4.2 Target Fitness

In our research, we have developed a novel fitness scoring mechanism, denoted as the Fitness Score (FS), designed to maximize the Dunn validity index while minimizing the number of selected features. To achieve this objective, we have incorporated a penalty system for the selection of features. Specifically, a penalty is imposed on the fitness of a wolf in the feature selection process when an excessive number of features are selected, thereby striking a balance between maximizing Dunn validity index and optimizing feature selection efficiency. Equation (4) is used to calculate the fitness of wolf during optimization process.

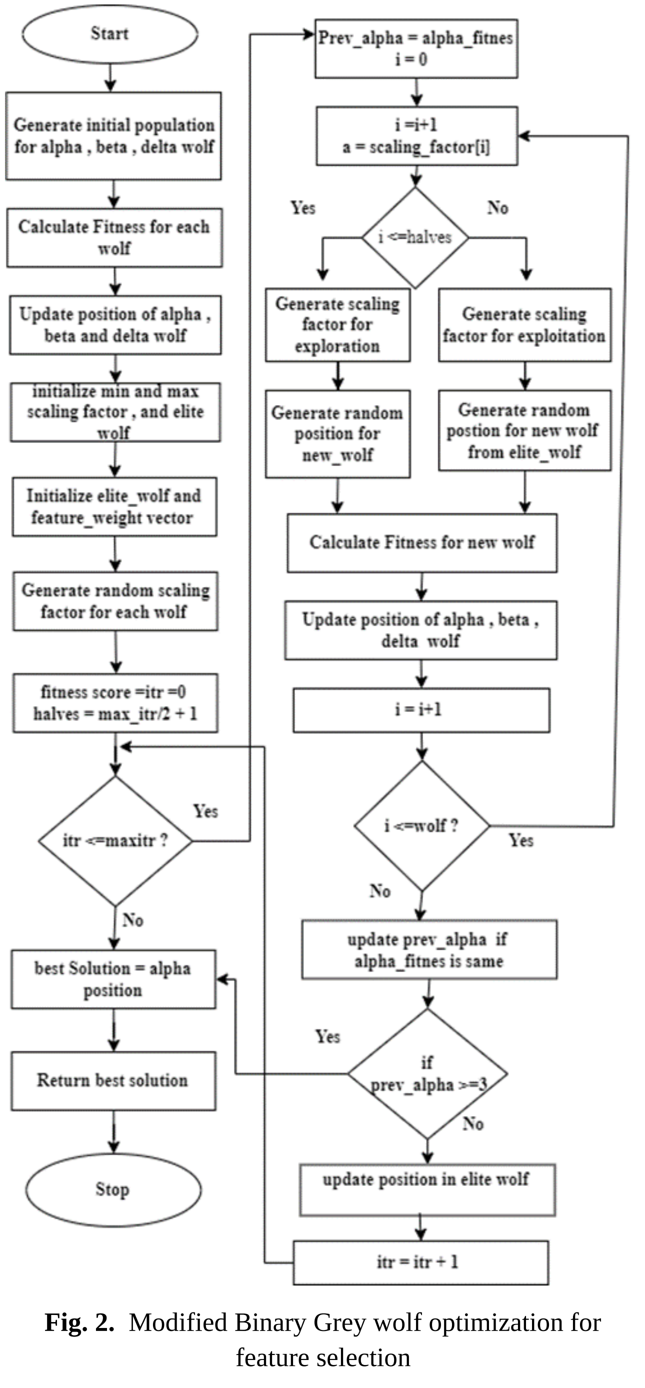
Start
Generate initial population for alpha , beta , delta wolf
Calculate Fitness for each wolf
Update position of alpha , beta and delta wolf
initialize min and max scaling factor , and elite wolf
Initialize elite_wolf and feature_weight vector
Generate random scaling factor for each wolf
fitness score =itr =0
halves = max_itr/2 + 1
itr <=maxitr ?
Yes
No
best Solution = alpha position
Return best solution
Stop
Prev_alpha = alpha_fitnes
i = 0
i =i+1
a = scaling_factor[i]
Yes
No
i <=halves
Generate scaling factor for exploration
Generate scaling factor for exploitation
Generate random position for new_wolf
Generate random postion for new wolf from elite_wolf
Calculate Fitness for new wolf
Update position of alpha , beta , delta wolf
i = i+1
i <=wolf ?
Yes
No
update prev_alpha if alpha_fitnes is same
Yes
if
prev_alpha >=3
No
update position in elite wolf
itr = itr + 1

Fig. 2. Modified Binary Grey wolf optimization for feature selection

$$FS = 1 - \frac{Dunn\ Index - Min\ Dunn}{Max\ Dunn - \ Min\ Dunn} - Penalty * \frac{Feature}{Max\ Feature}\ , if\ max\ Dunn\ \neq Min\ Dunn\ else\ 0$$

(4)

4.3 Performance Evaluation

Measures developed for static scenarios and for labelled data are limited in their ability to accurately capture errors resulting from the dynamic nature of evolving clusters[28], traditional static measures may fail to account for the evolving nature stream.

The performance of stream can be evaluated using two different approaches: internal and external measures[1], [28], depending on the availability of ground truth label for the dataset. Our study measured the performance of the proposed method using an internal evaluation. This method does not rely on ground truth labels for assessing clustering performance; instead, it assesses the clustering quality by analyzing the cluster structure.

In evaluating clustering performance, the ideal quality of clustering is defined by minimal intra-cluster distance and maximal inter-cluster distance. We utilized internal measures to assess clustering quality, eliminating the need for ground truth labels. Some of the internal performance metrics employed in our study included the Dunn Index[29], Calinski-Harabasz Index, Davies Bouldin Index[29], silhouette score, compactness(CP), and separation(SP).

Compactness measures the proximity of data points within a cluster, determined by the distances between data points within the same cluster. Separation measures the dissimilarity between clusters [2].

The Davies Bouldin Index[2][30] represents the ratio of compactness to separation, where a value of 0 indicates compact and well-separated clusters. A lower Davies Bouldin Index signifies non-overlapping clusters. It is calculated from Ri values, which are based on intra-cluster dispersion (Si and Sj) and separation (Mij). Davies Bouldin index (DB – index) is calculated using (5), (6) and (7).

$$\text{Davies Bouldin Index } = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Ri} \qquad (5)$$

$$\mathrm{Rij} = \frac{\mathrm{Si + Sj}}{\mathrm{Mij}} \qquad (6)$$

$$\mathrm{Ri} = \mathrm{maximum(Rij)} \qquad (7)$$

The Dunn validity index assesses the balance between separation and compactness[2], It is computed as the ratio of the minimum separation between points in distinct clusters to the maximum diameter within clusters[30]. A higher DI index signifies well-separated and compact clusters.

$$\text{Dunn index } = \frac{\text{Minimum Seperation}}{\text{Maximum Diameter}} \qquad (8)$$

Silhouette score[28] and silhouette plot are used to measure the separation between the cluster. It shows how each point is close within cluster with other points in neighboring cluster.

The Silhouette score(SC) is utilized to evaluate the separation between clusters, illustrating the closeness of points within a cluster compared to neighboring clusters. The Silhouette coefficient is calculated as the difference between mean intra-cluster distance (a) and mean nearest cluster distance (b) divided by the maximum of a and b.

$$\text{Shilhouette coefficient } = \frac{(b - a)}{\max(\mathrm{a,b})} \qquad (9)$$

The Calinski-Harabasz (C-H) index determines the ratio of inter-cluster dispersion to intra-cluster dispersion across all clusters [29], [30]. It involves calculations of BGSS (between-group sum of squares) and WGSS (within-group sum of squares) as in (13), with K represents the total number of clusters and N being the number of observations. BGSS and WGSS are computed based on the center points of the dataset and centroids of clusters, respectively.

$$\text{Calinski} - \text{Harabasz index } = \frac{\mathrm{BGSS}}{\mathrm{WGSS}} \times \frac{\mathrm{N-K}}{\mathrm{K-1}}$$

(10)

$$\mathrm{BGSS} = \sum_{\mathrm{k=1}}^{\mathrm{k}} \mathrm{N_k} \times \|C_k - C\|^2 \qquad (11)$$

where N_k indicate number of observations in cluster k , C is center point of dataset and C_k is centroid of cluster k. O_{i_k} is ith observation of cluster k , $WGSS_k$ indicate within group of sum square for cluster k and calculated using (12).

$$\mathrm{WGSSk} \ = \sum_{i=1}^{Nk} \left\| O_{i_k} - C_k \right\|^2 \qquad (12)$$

$$\mathrm{WGSS} = \sum_{\mathrm{K=1}}^{\mathrm{K}} \mathrm{WGSS_k} \qquad (13)$$

5. Experimental Methodology

5.1 Environment Setup

The experiments were performed in a streaming environment using Apache Spark 3.2.2 and Python 3.10.14 on a computer system with 8GB of RAM and a 500GB HDD. Apache Spark was selected for its efficient handling of streaming data, allowing for real-time processing and analysis. The hardware configuration, including 8GB of RAM and a 500GB HDD, offered ample resources to manage the computational demands of the experiments, guaranteeing smooth execution and dependable results.

5.2 Dataset

For the evaluation of our clustering methodologies, we utilized a range of well-known datasets including Reuters-21578, 20Newsgroups, AGnews, and BBCNews. The Reuters-21578 dataset, containing articles from the Reuters financial newswire service, provides a diverse set of documents for clustering tasks. The 20Newsgroups dataset, with newsgroup discussions categorized into 20 topics, offers a varied collection of text for clustering algorithms. AGnews, featuring news articles across different categories, presents a practical challenge for clustering tasks. Additionally, the BBCNews dataset, comprising news articles from various domains, enables the exploration of diverse topics in text clustering. By employing these datasets, our objective was to assess the performance and effectiveness of our proposed approaches across different text data scenarios.

5.3 Result Discussion

In this section, we embark on a comprehensive discussion of the results obtained from our study focusing on stream clustering with feature selection using Modified Grey Wolf Optimization (GWO). Our investigation involved the implementation of the proposed GWO approach and a comparative analysis against standard binary GWO, standard binary Particle Swarm Optimization (PSO), and a scenario where features are not selected in the stream environment. By evaluating these methodologies, we sought to assess the efficacy of our proposed GWO technique in enhancing stream clustering performance through optimized feature selection. The ensuing discussion will delve into the comparative outcomes, the implications of the results on stream clustering productivity, and the potential advancements introduced by the proposed GWO method.

For each dataset utilized in the study, individual tables were meticulously constructed to present the evaluation metrics corresponding to the different feature selection methods employed. These tables were specifically tailored to showcase the performance of each feature selection approach across distinct datasets, allowing for a detailed comparison of their effectiveness. By segregating the results by dataset and feature selection method, a comprehensive analysis was facilitated, enabling a comprehensive understanding of how each method performed across various evaluation metrics and datasets.

The performance evaluation results for the 20newsgroup dataset are displayed in Table 1, as indicated by the outcomes. Higher values of the Dunn index indicate better cluster separation in result. On this dataset, MBGWOSFS and BGWO outperform the baseline NoFS method and BPSO, with MBGWOSFS achieving the highest Dunn index of 72.7538. MBGWOSFS shows the lowest DB index and CH index values, signifying more compact and well-separated clusters. It also demonstrates a superior balance between compactness and separation. The MBGWOFS feature selection method, with 106 selected features, stands out as the most promising for clustering on 20Newsgroup data stream, showing superior performance across various evaluation metrics compared to other methods.

According to Table 2., The MBGWOSFS method, with 83 features, yielded a Dunn index of 57.846 and a Silhouette score of 0.047, surpassing both the NoFS , BGWO and BPSO methods in cluster separation and data similarity within clusters. The DB index and CH index improved to 3.264 and 0.43, respectively, showcasing more compact and well-separated clusters on Reuters-21578 dataset.

Performance analysis on AGNews dataset shows that BGWO , MBGWOFS and BPSO methods seem to perform comparably in terms of cluster separation and data similarity within clusters. While there may be minor variations in the metrics, the overall clustering quality and performance across the methods appear to be consistent.

With 114 features in Table 4., the MBGWOSFS method on BBCNews dataset showcases a robust Dunn index of 68.345, reflecting superior cluster separation. The Silhouette score of 0.033 implies well-matched data within clusters, while the DB index of 2.631 and CH index of 0.386 suggest balanced clustering characteristics. The compactness and separation values align at 0.033 each, further indicating well-structured clusters.

Summary of proposed MBGWOSFS method on different dataset is provided in Table 5, The proposed MBGWOSFS method demonstrates strong performance across all datasets, showcasing effective cluster separation, high data similarity, and overall quality clustering results. This method proves to be successful in achieving positive outcomes when applied to various datasets, including 20Newsgroup, Reuters-21578, AGNews, and BBCNews.

The average number of optimal clusters shown as in Fig.5 , evolved during stream clustering using different methods showcases interesting patterns across various datasets. The MBGWOSFS method consistently showed an average number of optimal clusters across all datasets. This stability and adaptability to different datasets suggest that the MBGWOSFS method effectively identifies and adapts to the underlying structures within the data, resulting in a reliable clustering outcome. This ability to evolve the optimal number of clusters, based on the dataset characteristics, contributes to the method's robustness and performance consistency across diverse datasets.

We will now discuss the results method-wise, evaluating their effectiveness in the realm of feature selection within a streaming environment. We will commence the analysis with NoFS, signifying the scenario where no feature selection method is applied. Based on the clustering results obtained for the NoFS(No Feature selection) method across different datasets, the following conclusions can be drawn:

- The Dunn index values ranged between 36.1085 and 43.1796 across the datasets, indicating a moderate level of cluster separation achieved by the NoFS method. While the method consistently provided acceptable cluster distinctions, it did not showcase significant improvements in separation across the datasets.
- The Silhouette scores varied between 0.023 and 0.027, indicating that the clusters formed by the NoFS method had fair consistency in data similarity within clusters but were not highly cohesive in all cases.
- The DB index and CH index values, measuring compactness and separation of clusters, showed values ranging from 4.7622 to 5.5994 and 0.4413 to 0.4525, respectively. While the NoFS method achieved decent compactness, there is a minor room for enhancing the separation of clusters for more distinct groupings.

In conclusion, the NoFS method demonstrates a consistent performance in achieving moderate cluster separation and data similarity within clusters across the datasets. While it provides reasonable clustering outcomes, there is potential for improvement in enhancing cluster distinctiveness and separation for more effective clustering results.

After evaluating the clustering results for the Binary Grey Wolf Optimization (BGWO) method across various datasets, several significant conclusions can be made.:

- The Dunn index values for the BGWO method showcase a stable performance, ranging from 56.955 to 69.7452. This consistency in cluster separation indicates that the BGWO method is reliable in distinguishing clusters effectively.
- The Silhouette scores range between 0.005 and 0.027, suggesting some variability in the data similarity within clusters across the datasets. While the method may show slightly different levels of cohesion, it generally maintains a satisfactory data similarity.
- The DB and CH index values indicate balanced clustering quality, with values ranging from 1.64 to 2.8076 and 0.267 to 0.3753, respectively. This balance between cluster compactness and separation highlights the method's ability to generate meaningful cluster structures.
- The Compactness values fluctuate between 0.005 and 0.08, suggesting potential areas for enhancing the compactness of clusters in some datasets.

In conclusion, the BGWO method exhibits consistent cluster separation and satisfactory data similarity within clusters across the datasets. While there are slight variations in the clustering quality, the method demonstrates effectiveness in generating well-defined cluster structures with balanced compactness and separation. Improving compactness in certain instances may further enhance the method's clustering performance.

Following the analysis of the Binary Particle Swarm Optimization (BPSO) method across different datasets, the conclusions are as follows:

- The BPSO method consistently demonstrates a stable Dunn index, with values ranging from 55.534 to 68.345. This suggests reliable cluster separation capabilities across the datasets analysed.
- The Silhouette scores exhibit variability, with values spanning from 0.0045 to 0.033. This indicates that while the method can achieve good data similarity within clusters, there are instances where it may perform better in maintaining cohesion.
- The DB and CH indices showcase the method's ability to balance cluster quality, with values fluctuating across the datasets. The Compactness and Separation metrics also reflect this balance, indicating a mix of compactness and separation in the cluster structures.
- Instances where the Compactness values are higher suggest areas where the method can improve in creating denser and more compact clusters for certain datasets.

In conclusion, the BPSO method demonstrates consistent performance in cluster separation across different datasets. While showing variability in data similarity, the method maintains a good balance in cluster quality. Further enhancements in compactness for specific datasets may improve the overall clustering effectiveness of the BPSO method.

Proposed MBGWOSFS stream clustering approach, reveals consistent and successful clustering outcomes across different datasets:

- The MBGWOSFS method consistently demonstrates a strong Dunn index, ranging from 57.846 to 72.7538, indicating effective cluster separation in all datasets.
- The Silhouette scores, with values between 0.0324 and 0.047, highlight strong data similarity within clusters, showcasing cohesive clustering structures.
- The DB and CH indices, with values ranging from 2.631 to 3.264 and 0.3688 to 0.43 respectively, indicate a balanced cluster quality with optimal compactness and separation.
- The MBGWOSFS method outperforms other methods by consistently achieving higher Dunn index values and

comparable Silhouette scores. Its ability to strike a balance between cluster quality measures, compactness, and separation places it as a robust and effective clustering method across various datasets.

Overall, the MBGWOSFS method exhibits superior cluster separation, strong data similarity within clusters, and balanced cluster quality characteristics, making it a reliable and high-performing clustering approach compared to other methods analysed. Its consistent performance and ability to deliver well-defined clusters showcase its effectiveness in producing quality clustering results.

Table 1. Performance measurement on 20Newsgroup DataStream

Method	#	Dunn index	SH	DB index	CH index	CP	SP
NoFS	--	43.1796	0.0241	5.5994	0.4525	0.0241	0.0241
BGWO	113	69.7452	0.0217	2.8076	0.3753	0.0324	0.0324
MBG-WOFS	106	72.7538	0.0348	2.8122	0.3823	0.0348	0.0348
BPSO	112	68.2152	0.0225	1.9085	0.3631	0.0386	0.0347

Table 2. Performance measurement on Reuters-21578 DataStream

Method	#	Dunn index	SH	DB index	CH index	CP	SP
NoFS		36.1085	0.027	4.7622	0.4413	0.027	0.027
BGWO	82	56.955	0.027	2.146	0.346	0.08	0.027
MBG-WOSFS	83	57.846	0.047	3.264	0.43	0.047	0.047
BPSO	85	55.534	0.023	2.326	0.327	0.0798	0.025

Table 3. Performance measurement on AGNews DataStream

Method	#	Dunn index	SH	DB index	CH index	CP	SP
NoFS	--	42.8621	0.0232	5.5768	0.4515	0.023	0.023
BGWO	113	67.398	0.02	2.606	0.367	0.02	0.02
MBG-WOSFS	115	67.8874	0.0324	2.8251	0.3688	0.027	0.026
BPSO	115	66.286	0.0197	2.762	0.323	0.024	0.021

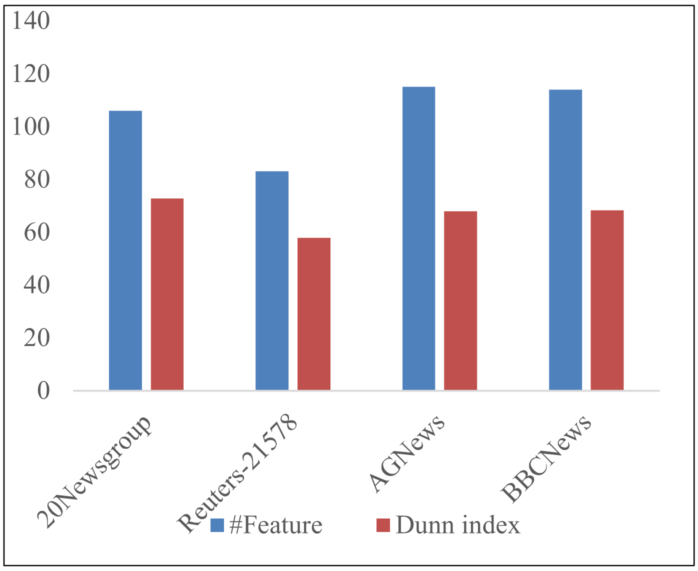

Fig 3. Comparison of #Feature selected and Accuracy on DataStream

Table 4. Performance measurement on BBCNews DataStream

Method	#	Dunn index	SH	DB index	CH index	CP	SP
NoFS	--	41.925	0.025	5.394	0.448	0.023	0.023
BGWO	76	57.128	0.005	1.64	0.267	0.005	0.005
MBGW-OSFS	114	68.345	0.033	2.631	0.386	0.033	0.033
BPSO	79	55.823	0.0045	1.451	0.247	0.004	0.004

Table 5. Performance measurement of Proposed MBGWOSFS

Data Stream	#	Dunn index	SH	DB index	CH index	CP	SP
20News	106	72.7538	0.0348	2.812	0.3823	0.0348	0.0348
Reuters-21578	83	57.846	0.047	3.264	0.43	0.047	0.047
AGNews	115	67.8874	0.0324	2.825	0.3688	0.0256	0.0256
BBCNews	114	68.345	0.033	2.631	0.386	0.033	0.033

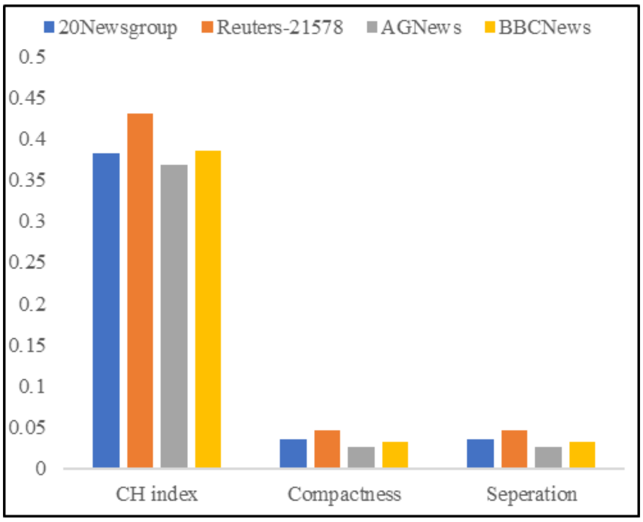

Fig 4. Accuracy Measurement for Data

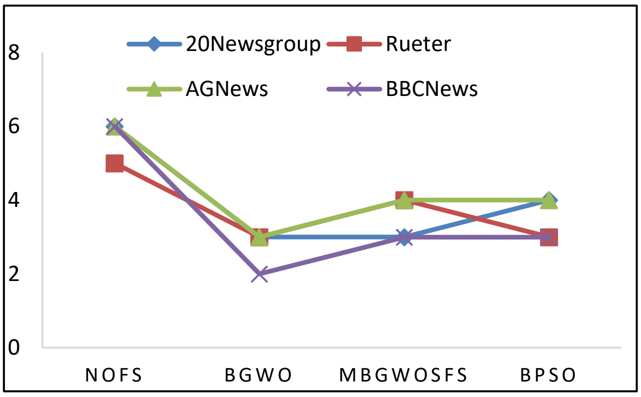

Fig 5. Average number of Optimal Cluster Discovered in different data stream

6. Conclusion and Future Work

We have proposed modified grey wolf optimization for feature selection for stream data , which uses different approach for generation of wolf position during exploration and exploitation phase . We have successfully addressed the problem of grey wolf optimization and avoid to trap in local optima , apart from that we have used dynamic k-means clustering algorithm which will dynamically adapt the value of optimal cluster using silhouette score . Proposed work consistently shows the higher dunn index , indicating better cluster separation , lower value of DB index and C-H index shows that proposed work has created more compact and well separated cluster . Hence it Emerged as a robust method, consistently delivering high-quality clustering with well-defined clusters and strong cluster separation. Its ability to adapt to dataset nuances and consistently evolve the optimal number of clusters makes it a reliable and effective choice for stream clustering tasks as compared to other method.

In the future, we plan to explore alternative methods for data conversion. While our current approach involves numerical conversion using hashing for clustering compatibility, we aim to adopt a more advanced technique to enhanced text conversion and improved clustering outcomes. Additionally, our algorithm has been implemented within the dynamic k-means clustering framework. Moving forward, we intend to assess the performance of our proposed method across various different clustering algorithms.

References

[1] U. Kokate, A. Deshpande, P. Mahalle, and P. Patil, “Data Stream Clustering Techniques, Applications, and Models: Comparative Analysis and Discussion,” *BDCC*, vol. 2, no. 4, p. 32, Oct. 2018, doi: 10.3390/bdcc2040032.

[2] A. Zubaroğlu and V. Atalay, “Data stream clustering: a review,” *Artif Intell Rev*, vol. 54, no. 2, pp. 1201–1236, Feb. 2021, doi: 10.1007/s10462-020-09874-x.

[3] S. R. Tiwari and K. K. Rana, “Challenges and Future Research Directions for Stream Clustering,” in *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, IEEE, 2023, pp. 525–531. Accessed: May 18, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10125674/

[4] C. Fahy and S. Yang, “Dynamic Feature Selection for Clustering High Dimensional Data Streams,” *IEEE Access*, vol. 7, pp. 127128–127140, 2019, doi: 10.1109/ACCESS.2019.2932308.

[5] J. M. Yeoh, F. Caraffini, E. Homapour, V. Santucci, and A. Milani, “A Clustering System for Dynamic Data Streams Based on Metaheuristic Optimisation,” *Mathematics*, vol. 7, no. 12, p. 1229, Dec. 2019, doi: 10.3390/math7121229.

[6] D. Zhao and Y. S. Koh, “Feature Drift Detection in Evolving Data Streams,” in *Database and Expert Systems Applications*, vol. 12392, S. Hartmann, J. Küng, G. Kotsis, A. M. Tjoa, and I. Khalil, Eds., in Lecture Notes in Computer Science, vol. 12392. , Cham: Springer International Publishing, 2020, pp. 335–349. doi: 10.1007/978-3-030-59051-2_23.

[7] S. Harde and V. Sahare, “Design and implementation of ACO feature selection algorithm for data stream mining,” in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, IEEE, 2016, pp. 1047–1051. Accessed: Apr. 26, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7877746/

[8] A. K. Abasi, A. T. Khader, M. A. Al-Betar, S. Naim, S. N. Makhadmeh, and Z. A. A. Alyasseri, “An Improved Text Feature Selection for Clustering Using Binary Grey Wolf Optimizer,” in *Proceedings of the*

11th National Technical Seminar on Unmanned System Technology 2019, vol. 666, Z. Md Zain, H. Ahmad, D. Pebrianti, M. Mustafa, N. R. H. Abdullah, R. Samad, and M. Mat Noh, Eds., in Lecture Notes in Electrical Engineering, vol. 666. , Singapore: Springer Nature Singapore, 2021, pp. 503–516. doi: 10.1007/978-981-15-5281-6_34.

[9] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary grey wolf optimization approaches for feature selection,” *Neurocomputing*, vol. 172, pp. 371–381, 2016, Accessed: Apr. 26, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231215010504

[10] D. Wang, Y. Ji, H. Wang, and M. Huang, “Binary grey wolf optimizer with a novel population adaptation strategy for feature selection,” *IET Control Theory &amp; Appl*, vol. 17, no. 17, pp. 2313–2331, Nov. 2023, doi: 10.1049/cth2.12498.

[11] R. Ahmadi, G. Ekbatanifard, and P. Bayat, “A Modified Grey Wolf Optimizer Based Data Clustering Algorithm,” *Applied Artificial Intelligence*, vol. 35, no. 1, pp. 63–79, Jan. 2021, doi: 10.1080/08839514.2020.1842109.

[12] S. Fong, R. Wong, and A. V. Vasilakos, “Accelerated PSO Swarm Search Feature Selection for Data Stream Mining Big Data,” *IEEE Trans. Serv. Comput.*, vol. 9, no. 1, pp. 33–45, Jan. 2016, doi: 10.1109/TSC.2015.2439695.

[13] X. Hu, P. Zhou, P. Li, J. Wang, and X. Wu, “A survey on online feature selection with streaming features,” *Front. Comput. Sci.*, vol. 12, no. 3, pp. 479–493, Jun. 2018, doi: 10.1007/s11704-016-5489-3.

[14] S. Tabakhi, P. Moradi, and F. Akhlaghian, “An unsupervised feature selection algorithm based on ant colony optimization,” *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112–123, Jun. 2014, doi: 10.1016/j.engappai.2014.03.007.

[15] H. Amazal and M. Kissi, “A New Big Data Feature Selection Approach for Text Classification,” *Scientific Programming*, vol. 2021, pp. 1–10, Apr. 2021, doi: 10.1155/2021/6645345.

[16] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, “Binary optimization using hybrid grey wolf optimization for feature selection,” *Ieee Access*, vol. 7, pp. 39496–39508, 2019, Accessed: Apr. 26, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8672550/

[17] X. Yan, A. Homaifar, M. Sarkar, B. Lartey, and K. D. Gupta, “An online unsupervised streaming features selection through dynamic feature clustering,” *IEEE Transactions on Artificial Intelligence*, 2022, Accessed: Apr. 29, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9851506/

[18] N. Almusallam, Z. Tari, J. Chan, and A. AlHarthi, “UFSSF - An Efficient Unsupervised Feature Selection for Streaming Features,” in *Advances in Knowledge Discovery and Data Mining*, vol. 10938, D. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, and L. Rashidi, Eds., in Lecture Notes in Computer Science, vol. 10938. , Cham: Springer International Publishing, 2018, pp. 495–507. doi: 10.1007/978-3-319-93037-4_39.

[19] S. Mansalis, E. Ntoutsi, N. Pelekis, and Y. Theodoridis, “An evaluation of data stream clustering algorithms,” *Statistical Analysis*, vol. 11, no. 4, pp. 167–187, Aug. 2018, doi: 10.1002/sam.11380.

[20] C. Fahy, S. Yang, and M. Gongora, “Ant Colony Stream Clustering: A Fast Density Clustering Algorithm for Dynamic Data Streams,” *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2215–2228, Jun. 2019, doi: 10.1109/TCYB.2018.2822552.

[21] “A Novel Feature selection using Binary Gray wolf optimization with featuer weights for unstructured stream clustering”.

[22] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, “Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification,” *Neural Comput & Applic*, vol. 32, no. 16, pp. 12201–12220, Aug. 2020, doi: 10.1007/s00521-019-04368-6.

[23] R. Purushothaman, S. P. Rajagopalan, and G. Dhandapani, “Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering,” *Applied Soft Computing*, vol. 96, p. 106651, 2020, Accessed: Apr. 26, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620305895

[24] L. Zhang and X. Chen, “A Velocity-Guided Grey Wolf Optimization Algorithm With Adaptive Weights and Laplace Operators for Feature Selection in Data Classification,” *IEEE Access*, vol. 12, pp. 39887–39901, 2024, doi: 10.1109/ACCESS.2024.3376235.

[25] I. M. El-Hasnony, S. I. Barakat, M. Elhoseny, and R. R. Mostafa, “Improved feature selection model for big data analytics,” *IEEE Access*, vol. 8, pp. 66989–67004, 2020, Accessed: Apr. 29, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9058715/

[26] S. Xu, L. Feng, S. Liu, and H. Qiao, “Self-adaption neighborhood density clustering method for mixed data stream with concept drift,” *Engineering Applications of Artificial Intelligence*, vol. 89, p. 103451, Mar. 2020, doi: 10.1016/j.engappai.2019.103451.

[27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[28] H. Kremer *et al.*, “An effective evaluation measure for clustering on evolving data streams,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego California USA: ACM, Aug. 2011, pp. 868–876. doi: 10.1145/2020408.2020555.

[29] L. E. Ekemeyong Awong and T. Zielinska, “Comparative Analysis of the Clustering Quality in Self-Organizing Maps for Human Posture Classification,” *Sensors*, vol. 23, no. 18, p. 7925, Sep. 2023, doi: 10.3390/s23187925.

[30] M. Hassani and T. Seidl, “Using internal evaluation measures to validate the quality of diverse stream clustering algorithms,” *Vietnam J Comput Sci*, vol. 4, no. 3, pp. 171–183, Aug. 2017, doi: 10.1007/s40595-016-0086-9.