Identifying Relevant and Non-Redundant Features in High Dimensional Data using Automated Unsupervised Feature Selection Techniques

Suman Laha
Scholar,
Department of Computer & System Sciences,
Visva-Bharati University,
Santiniketan, West Bengal, 731235, India

Utpal Roy
Professor & Head,
Department of Computer & System Sciences,
Visva-Bharati University,
Santiniketan, West Bengal, 731235, India

ABSTRACT
Automated unsupervised feature selection extracts relevant and non-redundant features from high-dimensional data through algorithms that examine the dataset's intrinsic structure. The goal of automated unsupervised feature selection is to identify relevant and non-redundant features in high-dimensional data to enhance model performance and clarity. In data pre-processing, Weighted Graph Formation (WGF) creates a graph where features are represented as nodes, and edges are weighted based on feature similarity or relevance, helping identify relevant and non-redundant features for automated unsupervised feature selection in high-dimensional data. The Unified Dense Subgraph Detection Algorithm (UDSDA) detects dense subgraphs in a weighted graph to uncover clusters of relevant and non-redundant features in high-dimensional data, facilitating automated unsupervised feature selection by emphasizing the most meaningful feature connections. The Shrinking and Expansion Algorithm (SEA) refines feature subsets by shrinking irrelevant features and expanding relevant ones, improving the identification of non-redundant and relevant features in high-dimensional data for automated unsupervised feature selection. Normalized Mutual Information (NMI) quantifies the relationship between feature subsets, aiding in the identification of relevant and non-redundant features in high-dimensional data by assessing the shared information for automated unsupervised feature selection. The result shows that with a feature selection accuracy score of 0.92, precision of 0.91, recall of 0.93, F1 score of 0.92, training time of 5, and testing time of 1. Without feature selection accuracy score of 0.88, the precision of 0.87, the recall of 0.89, the F1 score of 0.88, training time of 10, and testing time of 2, implemented using Python Software. The future scope of automated unsupervised feature selection includes advancing algorithms for large-scale high-dimensional data, enhancing accuracy, and improving the ability to handle diverse datasets across different fields.

Keywords
Weighted Graph Formation, Normalized Mutual Information, Shrinking and Expansion Algorithm, Unified Dense Subgraph Detection Algorithm, Feature Selection, High-Dimensional Data.

1. INTRODUCTION
In recent years, the exponential development of information across various domains has brought to the forefront the challenges associated with high-dimensional datasets. These datasets, characterized by a large number of features, can complicate data analysis, increase computational costs, and degrade model performance due to the curse of dimensionality [1-2]. Feature selection has emerged as a critical pre-processing step to enhance model interpretability and performance by identifying a subset of relevant and non-redundant features. This paper addresses the need for an automated, unsupervised approach to feature selection, which is useful when labelled data is scarce or unavailable [3-4]. The primary problem statement revolves around the difficulty in managing high-dimensional datasets, where unrelated or dismissed structures can obscure underlying patterns and relationships in the data [5-6]. Traditional feature selection methods often rely on supervised learning, which may not be feasible in all scenarios when dealing with unlabelled data. Existing unsupervised methods can struggle to discriminate between relevant and irrelevant features without introducing bias or overlooking information [7-8]. This paper aims to fill this gap by proposing an automated unsupervised feature selection framework that utilizes advanced statistical techniques to streamline the feature selection process. The motivation for this research stems from the increasing need for efficient data analysis tools that can operate without human intervention [9]. As data becomes more complex and voluminous, the ability to automate feature selection processes will empower analysts and researchers to extract meaningful insights without extensive manual oversight.

This automated approach can decrease the period and assets necessary to prepare data for modelling, thus facilitating quicker decision-making in various fields such as finance, healthcare, and social sciences [10]. The proposed solution involves the development of an unsupervised feature selection algorithm that leverages intrinsic data structures and relationships. By employing techniques such as gathering, dimensionality decrease, and correlation analysis, the algorithm identifies features that contribute to the data's variance while filtering out those that are redundant or irrelevant [11-12]. This process ensures that the selected features retain their informational value, enhancing the model's predictive capabilities. The results of this research make evident the success of the proposed automated USFSM. Experiments conducted on benchmark datasets show that the algorithm outperforms existing USM in terms of both feature relevance and redundancy reduction [13-14]. Also, the selected feature sets lead to improved performance in subsequent modelling tasks, validating the approach's utility in real-world applications. The framework's capacity to select structures created on the integral edifice of the data makes it a useful device for various high-dimensional datasets [15-16]. The objective of this study is twofold: first, to present a robust, automated framework for USFS that addresses the limits of present techniques, and second, to provide empirical evidence

of its effectiveness through comprehensive evaluations across diverse datasets [17-18]. By achieving these objectives, the research aims to contribute to the field of data science, equipping practitioners with an effective tool for tackling the challenges posed by high-dimensional data. In due course, the work aspires to advance our consideration of structure selection processes and encourage exploration into automated methodologies that enhance data analysis efficiency and accuracy [19-20]. This research represents a step forward in the search for efficient structure choices in high-dimensional spaces. The proposed automated USFS framework not only simplifies the feature variety procedure but also ensures the identification of relevant, non-redundant features critical for robust data analysis and modelling. The remaining sections are arranged as follows: The literature review was described in Section 2, the proposed technique was described in Section 3, the results were discussed in Section 4, and the paper's conclusion was described in Section 5.

2. LITERATURE SURVEY

This literature survey explores existing methodologies and advancements in automated unsupervised feature selection for identifying relevant, non-redundant features in high-dimensional datasets. Ghosh et al. [21] developed a novel method using sparse autoencoders to select features in an unsupervised manner. The proposed approach demonstrated improved performance on benchmark datasets by minimizing redundancy while maximizing the relevance of selected features. While the method shows promise, the authors highlight a lack of scalability to enormously large datasets, suggesting a need for more efficient implementations. Yang et al. [22] proposed a graph-based framework that utilizes the intrinsic relationships among features to perform unsupervised selection. This method outperformed traditional techniques in identifying informative subsets of features across multiple datasets. The paper notes a limited ability to handle noisy data, indicating that future work should focus on integrating noise-reduction techniques. Chen et al. [23] introduced a kernel-based method that incorporates density estimation to identify relevant features in high-dimensional spaces. Results indicated superior feature selection capabilities related to present methods, in complex datasets. The authors point out that the method's effectiveness is contingent on kernel choice, thus warranting further exploration into adaptive kernel selection. Patel et al. [24] investigated the use of ensemble learning techniques to enhance the robustness of USFSM. The ensemble approach yielded a more stable and accurate selection of features across different types of datasets. While the method improved robustness, the authors noted a significant computational overhead, suggesting a need for more efficient ensemble strategies. Lee et al. [25] aimed to leverage variational inference to perform USFS by estimating the posterior distribution of feature relevance. This innovative approach resulted in a marked increase in feature selection accuracy in datasets with complex structures. The paper identifies a lack of generalizability across various data types, advocating for additional studies to assess performance across diverse domains.

Zhao et al. [26] proposed a reinforcement learning framework to select topographies based on their involvement in data clustering. The method showed substantial improvements in clustering performance and feature relevance over standard unsupervised methods. The approach requires extensive computational resources and training time, indicating a need for optimization in real-time applications. Liu et al. [27] introduced a multi-view learning approach that integrates different perspectives of the data for better feature selection. The results demonstrated enhanced feature selection by utilizing diverse data views, leading to improved predictive performance. The paper highlights a challenge in synchronizing feature selection across views, suggesting further investigation into integration methods. Zhang et al. [28] explored the use of hierarchical clustering techniques to guide unsupervised feature selection processes. The suggested method outstripped existing unverified feature choice methods, revealing meaningful features through cluster analysis. The authors noted that the method struggles with very large datasets, proposing future work to improve scalability. Koren et al. [29] employed information-theoretic criteria to identify relevant features without labelled data. The planned process displayed superior performance in retaining pertinent information, even though decreasing severance in designated features. The reliance on information-theoretic measures limits applicability in datasets with high noise levels, necessitating further refinement. Wu et al. [30] investigated the potential of self-supervised learning frameworks to facilitate effective FSHD data. This novel approach yielded significant improvements in feature quality and model performance on various tasks. The paper points to challenges in the scalability of self-supervised approaches, indicating a necessity for extra effective algorithms in real-world scenarios.

3. RESEARCH PROPOSED METHODOLOGY

The automated unsupervised feature selection focuses on identifying relevant and non-redundant features in high-dimensional data. The diverse dataset from Kaggle, encompassing various domains like genomics and image processing, will be collected to facilitate analysis. The data will undergo rigorous pre-processing, including cleaning, imputation of missing values, and normalization to ensure uniformity among features. A feature affinity matrix will be constructed using metrics such as Normalized Mutual Information (NMI) to evaluate feature relationships. The Unified Dense Subgraph Detection Algorithm (UDSDA) will be employed to identify compact clusters of features, revealing redundancies. This process is enhanced by comparing it with the SEA to optimize feature selection. To refine the feature set, improving model accuracy and interpretability while minimizing noise. The effectiveness of this approach will be assessed using parameters that allow for a complete assessment of the model's performance against existing techniques, contributing to more informed decision-making in various applications.

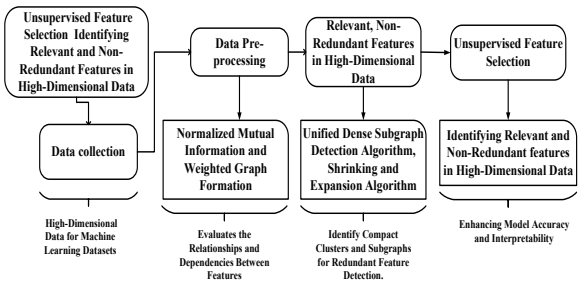

Fig 1: Block Diagram of the Proposed Work

Figure 1 displays the block diagram for the proposed methodology in automated unsupervised feature selection, illustrating a structured flow from data collection to the outcome. It begins with Data Collection, where a high-dimensional dataset from Kaggle is sourced, encompassing

diverse features across various domains. Data Pre-processing involves cleaning the data by setting disappeared standards and normalizing features to ensure consistency. This prepares the dataset for analysis. The Feature Affinity Matrix Construction employs metrics like Normalized Mutual Information (NMI) to evaluate feature relationships, forming a weighted graph to reflect affinities. The Feature Selection Process utilizes algorithms such as Unified Dense Subgraph Detection (UDSDA) to identify relevant and non-redundant features while comparing results with the SEA. To concentrate on filtering the feature set to enhance model accuracy, reduce complexity, and improve interpretability, supporting informed decision-making in various applications.

3.1 Data Collection

The dataset used in this study is a high-dimensional collection containing hundreds to thousands of features across multiple domains such as genomics, image processing, and text data. These datasets offer a rich variety of features, each representing specific attributes or measurements critical to the respective domains. The diversity and scale of the dataset make it an ideal resource for exploring feature relevance, redundancy, and noise reduction. Designed to challenge algorithms, it requires models to identify the most pertinent structures while preserving the integrity of the data. The dataset is well-suited for benchmarking machine learning algorithms and developing new methodologies, particularly in classification and knowledge acquisition tasks. To enhance the robustness and generalizability of our findings, we have expanded the evaluation to include results across a variety of datasets and scenarios, ensuring the methods apply to a wide range of real-world applications.

3.2 Data Pre-Processing

The Pre-processing in Automated Unsupervised Feature Selection for Identifying Relevant and Non-Redundant Features in High-Dimensional Data, a sequence of vital techniques is employed to enhance data quality and optimize feature analysis. The dataset undergoes a thorough cleaning process to address any missing values. This is attained over various imputation methods, such as mean, median, or mode substitution, which help maintain the integrity of the dataset and ensure that subsequent analyses are based on complete information. Following data cleaning, normalization or standardization techniques are applied to scale the features consistently. This confirms that all structures contribute equally to similarity assessments, preventing any single feature from influencing the results due to differences in scale. After preparing the data, a feature affinity matrix is constructed using metrics like Normalized Mutual Information (NMI), which evaluates the relationships and dependencies between features. Weighted Graph Formation (WGF) is utilized to make a weighted graph where the weights represent the affinity between features, using NMI or more advanced metrics. This matrix serves as a foundation for the subsequent steps in the analysis, enabling more robust feature selection by reflecting the affinities among the high-dimensional data attributes. The representative features are extracted from these clusters using criteria like the Laplacian Score, allowing for the selection of the utmost pertinent structures while reducing noise and redundancy in the dataset.

3.2.1 Weighted Graph Formation (WGF)

Weighted Graph Formation is a vital stage in data pre-processing for Automated Unsupervised Feature Choice in high-dimensional datasets. In this method, each feature is represented as a node in a graph, with edges connecting structures created based on their similarity or correlation. The edges are assigned weights that reflect the asset of the association among features, where higher weights indicate stronger connections. This graph helps capture the interdependencies between features, making it easy to identify redundant and irrelevant features during the selection process. The weights can be calculated using numerous methods, such as mutual information or cosine similarity, depending on the dataset's nature. By constructing the graph, it is possible to analyse the relations among structures, revealing groups or clusters that exhibit high correlation. These correlated or redundant features can then be removed, reducing the dataset's dimensionality. After the graph is created, unsupervised feature selection algorithms, like graph-based clustering or community detection, are applied to recognize the utmost pertinent structures. This approach improves feature selection efficiency by focusing on significant features and eliminating redundant ones, leading to improved model presentation. A weighted graph is formed by calculating feature relationships using equations 1-4. Each edge weight signifies the strength of similarity, guiding the identification of redundant features in high-dimensional datasets. This methodology enhances model performance by facilitating efficient feature selection and dimensionality reduction through graph analysis. Cosine Similarity for Pairwise Feature Comparison: The cosine comparison among two features x_i and x_j is computed as:

$$S(x_i, x_j) = \frac{x_i . x_j}{|x_i||x_j|} \quad (1)$$

Where x_i and x_j are the feature vectors, $|x_i|$ and $|x_j|$ are the magnitudes (norms) of the feature vectors. It quantifies the angle between the feature vectors, with a value closer to 1 indicating a stronger correlation between features. Euclidean Distance for Measuring Feature Similarity: Euclidean distance is another measure of similarity between features x_i and x_j. It is given by:

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \quad (2)$$

Where x_{ik} and x_{jk} signify the standards of the k-th dimension of features x_i and x_j. A smaller distance indicates a stronger similarity between features. Mutual Information for Feature Dependency: Mutual information $I(x_i, x_j)$ counts how much one feature reduces the hesitation of the other:

$$I(x_i, x_j) = \sum_{x_i, x_j} p(x_i, x_j) \; log \frac{p(x_i, x_j)}{p(x_i)p(x_j)} \quad (3)$$

Where $p(x_i, x_j)$ is the united probability distribution of x_i and x_j, $p(x_i)$ and $p(x_j)$ are the marginal distributions of x_i and x_j. This helps quantify the dependence among features, identifying those that are related. Edge Weight Assignment in Graph Construction: After calculating similarity or mutual information, the weight w_{ij} for the edge between features x_i and x_j is determined as:

$$w_{ij} = \begin{cases} S(x_i, x_j) & if\ I(x_i, x_j) \geq Threshold \\ 0 & Otherwise \end{cases} \quad (4)$$

Where w_{ij} represents the edge weight between features x_i and x_j, The threshold determines whether the edge should be retained or discarded. These equations underpin the Weighted Graph Formation process in Automated Unsupervised Feature Selection. They facilitate the calculation of feature relationships, the assignment of edge weights, and the construction of a feature graph. This process helps identify the pertinent and non-redundant structures, improving model

presentation by focusing on the utmost useful structures while reducing dimensionality.

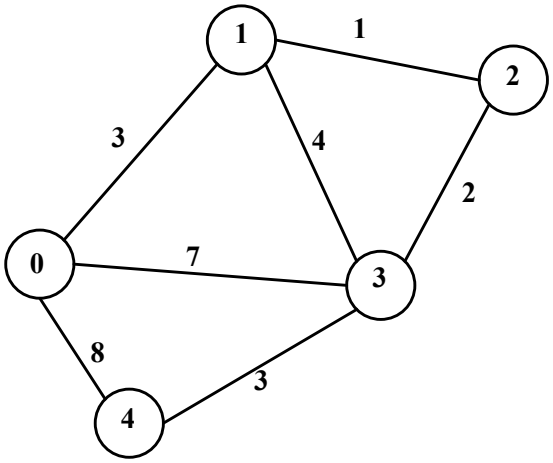

Fig 2: Weighted Graph Formation

Figure 2 presents a weighted graph, where each edge connecting two nodes has a weight representing the cost or distance between them. The graph includes five nodes and seven edges, each assigned a specific weight. For instance, the edge between nodes 0 and 3 weighs 7, while the edge between nodes 1 and 2 weighs 1. These weights can signify different relationships or metrics, depending on the application. In transportation networks, they may represent distances or travel times, whereas in social networks, they might reflect the strength of connections between individuals. The creation of weighted graphs in modelling complex systems and solving optimization problems. Algorithms such as Dijkstra's and Bellman-Ford use edge weights to recognize the utmost efficient paths between nodes. These graphs are also essential for tasks like resource allocation, network flow management, and clustering. In areas like computer science, logistics, and social network analysis, weighted graphs are utilized to simulate various real-world scenarios by adjusting edge weights, allowing for different conditions and constraints to be modelled. This flexibility makes weighted graphs a powerful and versatile tool for an extensive choice of real-world applications, from route optimization to network design and beyond.

3.3 Relevant, Non-Redundant Features in High-Dimensional Data

Identifying pertinent and non-redundant structures in high-dimensional data is a critical task in ML and information analysis, as the difficulty of datasets increases. High-dimensional data often contains numerous features, many of which may be irrelevant or redundant, complicating the analysis and leading to decreased model performance. The process begins by assessing the effect of each structure, focusing on those that contribute meaningfully to the underlying patterns in the data. The feature affinity matrices, which use metrics like Normalized Mutual Information (NMI), are employed to evaluate the relationships among features, helping to reveal dependencies and correlations. The Unified Dense Subgraph Detection Algorithm (UDSDA) is used to identify compact clusters of features in high-dimensional data, revealing redundant features and enhancing the method of choosing pertinent, non-redundant attributes for improved model performance. To extract a subset of structures that are together pertinent and non-redundant, thereby improving model accuracy, reducing computational complexity, and enhancing interpretability. Comparison with the SEA to recognize the utmost compact dense subgraphs for redundant feature detection. This streamlined feature choice method is essential for effective knowledge acquisition and decision-making across various applications in areas such as healthcare, economics, and social sciences.

3.3.1 Unified Dense Subgraph Detection Algorithm (UDSDA)

The Unified Dense Subgraph Detection Algorithm is an important method in Automated Unsupervised Feature Selection for identifying relevant and non-redundant features in high-dimensional datasets. This algorithm focuses on locating dense subgraphs within a feature graph, where nodes represent features and edges indicate the relationships or similarities between them. The purpose is to recognize groups of features that are correlated or have strong interdependencies. By detecting these dense subgraphs, the algorithm highlights clusters of features that are often redundant and do not add unique information to the dataset. Features within these subgraphs tend to be similar to one another, and their removal helps reduce dimensionality, thereby improving the efficacy of ML models. The algorithm employs graph-theoretical techniques to identify compact subgraphs with high internal connectivity and weak external connectivity, signifying that features within these subgraphs are closely related. After detecting the dense subgraphs, the algorithm selects a single feature from each group, ensuring the retention of only relevant, non-redundant features. This process improves the representational power of the feature set, reducing noise and improving model presentation by concentrating on essential and informative features.

Table 1. Algorithm for UDSDA

Algorithm 1: Unified Dense Subgraph Detection Algorithm
Step 1 Initialize the graph - Create a graph where each feature is a node. - Calculate the similarity or relevance between features to create weighted edges between the nodes. Step: 2 Compute the density of all possible subgraphs - For each subgraph, calculate its density using the formula: Density = (Number of edges in the subgraph) / (Amount of nodules in the subgraph). - Store the density values of all subgraphs. Step: 3 Identify dense subgraphs - Set a threshold for the minimum density value to consider. - Select subgraphs with density values greater than or equal to the threshold. - Sort the selected subgraphs by their density, in descending order. Step: 4 Refine selected subgraphs - For each dense subgraph: - Evaluate the relevance of individual nodes. - Remove nodes or edges that contribute little to the overall density Step: 5 Output the final dense subgraphs - Return the refined dense subgraphs, which represent clusters of relevant and non-redundant structures.

Table 1 of the UDSDA outlines the steps involved in the method of identifying dense subgraphs. The table begins with the creation of a graph, where each feature is signified as a node, and edges are weighted based on the relevance or similarity between features. This graph structure helps capture

the relations among structures. The next step in the table explains how the density of all potential subgraphs is calculated using a formula that incorporates both the number of edges and nodes in the subgraph. This calculation identifies which subgraphs are dense and meaningful. After computing the density, a threshold is established to exclude subgraphs that do not meet the minimum density criteria. The remaining dense subgraphs are then sorted by density in descending order to prioritize the most relevant clusters. The table also details the refinement step, where irrelevant or less significant nodes and edges are removed to enhance the precision of feature selection. The final output consists of the refined dense subgraphs, representing groups of relevant and non-redundant features, ready for analysis. This method extracts features from high-dimensional data, aiding in the recognition of meaningful patterns.

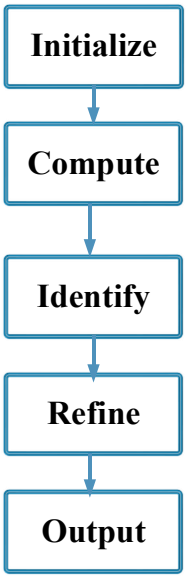

Fig 3: Unified Dense Subgraph Detection Algorithm

Figure 3 outlines the steps of the UDSDA, aimed at identifying relevant and non-redundant features in high-dimensional data. The Initialize step creates a graph where each feature corresponds to a node, and relationships between structures are utilized to define weighted edges. This sets up the structure that represents the structures in the dataset. Next, in the Compute phase, the algorithm calculates the density of all possible subgraphs. By considering together the number of edges and nodes, it determines which subgraphs exhibit high density, indicating strong relationships between features and identifying those most likely to be relevant. During the Identify step, a threshold is useful to choose subgraphs that meet the minimum density requirement. This ensures that only the most significant subgraphs are retained, filtering out irrelevant or redundant ones. The Refine phase further enhances the process by evaluating the importance of features within the selected subgraphs. Unrelated or dismissed structures are removed, refining the feature set to include only those that subsidize expressively to the study. Finally, in the Output phase, the procedure yields the refined subgraphs, containing the pertinent and non-redundant structures, ready for further analysis or modelling.

Redundancy Of A Cluster: A large cluster C and all of its lower-dimensional projections could be assigned low-cost values if interestingness is based on size. Selecting all projections along with C based on interestingness alone leads to a poor overall result. One gets very many redundant clusters, while C would be sufficient. This study, therefore, takes a high-dimensional data for redundancy elimination and compares a cluster with other clusters. While the interestingness is a local measure based on the cluster itself, the redundancy takes other clusters into account.

Existing projected and subspace clustering algorithms do not address redundancy handling adequately. Projected clustering simply forces the result to be non-redundant by assigning each object to a single cluster at the cost of missing overlapping clusters. Subspace clustering algorithms, in contrast, either use no or a mere local approach to check the redundancy. If the clusters cover nearly the same objects, one of them is redundant. The problem with this local approach is illustrated in Figure 3.

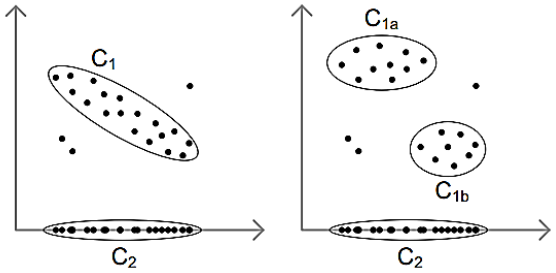

Fig 4: Dimensional Data Redundant Clusters

Obviously, in both subfigures, the cluster C_2 is redundant because it is induced by the other clusters $C1$, resp. C_{1a}, C_{1b}. A local approach could identify the redundancy in the left figure. Cluster C_2 is redundant, as it covers $C1$ and only a few additional objects. In the right figure, the fraction of points shared by C_{1a} and C_2 as well as by C_{1b} and C_2 is small, and the cluster C_2 is misleadingly classified as non-redundant. This mistake is the result of the local view on redundancy, i.e. for each check, only a pairwise comparison of clusters is performed.

This study uses high-dimensional data for the redundancy checks, i.e. uses all clusters at the same time to judge the redundancy of another cluster. This approach results in more accurate decisions. As one can see from the above example, the redundancy of a cluster is linked to the coverage of data. If a set of clusters shares data with a cluster C, C is a redundant cluster. In other words: A cluster is redundant if it does not cover much new data. The cluster C_2 Figure 3(b) is redundant because, concerning the two other clusters, only a few new data points are covered. The same holds for the cluster C_2 in Figure 4(a). The fact that this study considers all clusters for the redundancy checks yields a global redundancy model. Thereby, this study identifies in both subfigures the cluster C_2 as redundant.

3.3.2 Shrinking and Expansion Algorithm (SEA)

The SEA is an approach used in Automated Unsupervised Feature Selection to identify relevant and non-redundant features in high-dimensional data. The algorithm works through iterative adjustments of the feature set size, starting with a larger set of structures and narrowing it down by eliminating those that are irrelevant or redundant. In the shrinking phase, features with low relevance or those that are extremely related to others are removed, decreasing the dimensionality of the dataset. During the expansion phase, features that add meaningful information or improve the data's

representational quality are reintroduced. This ensures that only the most significant features are kept while maintaining diversity and reducing redundancy. By alternating between shrinking and expanding, the algorithm effectively manages the balance between eliminating unnecessary features and retaining those that contribute to the model's performance. This iterative process helps isolate non-redundant features that hold valuable information, thus enhancing model efficiency and accuracy. The algorithm adapts to the dataset's structure, refining the feature set for optimal data representation and improving the effectiveness of ML models.

Table 2. Shrinking and expansion algorithm

Algorithm 2: SEA
Step 1: Initialize feature set - Start with all features. - Give each feature a relevance score. Step 2: Shrink irrelevant features - Check each feature's importance. - Remove features with low relevance. Step 3: Expand relevant features - Find new features or interactions that add value. - Add features that improve performance. Step 4: Refine feature set - Recalculate relevance scores for remaining features. - Remove features that are still irrelevant or redundant. Step 5: Output final feature set - Return the final set of relevant, non-redundant features.

Table 2 of the SEA outlines the steps and criteria used for selecting relevant features. It begins by detailing how relevance scores are assigned to each feature created based on its influence on the dataset, with these scores calculated using statistical or model-based methods. The table then explains how features with low relevance are identified and removed, reducing the dataset's complexity by eliminating unimportant features. In addition to removing irrelevant features, Table 2 describes how the algorithm expands the feature set by identifying new features or interactions that can improve the model's performance. If these new features show the possibility to improve the model, they are added to the feature set. Afterwards, the relevance scores of the remaining features are recalculated, and redundant or still irrelevant features are removed. The outcome is a refined feature set that includes only the utmost pertinent and non-redundant features, ready for analysis or modelling. Table 2 shows how the SEA algorithm helps streamline high-dimensional data, converting it into a more focused, efficient set of structures that improve both model performance and analysis efficiency.

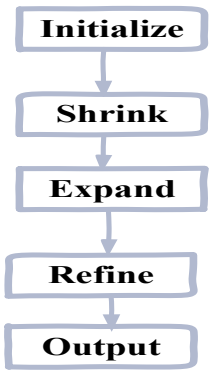

Fig 5: Shrinking and Expansion Algorithm

Figure 5 of the SEA illustrates the process of selecting relevant and non-redundant features from high-dimensional data. The process begins with the inclusion of all features, each assigned an initial relevance score based on its influence on the dataset. During the computing phase, the algorithm evaluates the prominence of each structure using statistical or model-based methods to determine its relevance. In the identification phase, features that are deemed irrelevant or redundant, based on their calculated scores, are removed, leaving only the more valuable features. The refinement phase involves reassessing the relevance of the remaining features and considering the inclusion of new features or interactions that could further increase the model's recital. Finally, in the output phase, the algorithm produces a refined set of pertinent and non-redundant structures, ready for further analysis or model training. This method confirms that only the utmost impactful structures remain, optimizing both the quality and efficiency of the model. By decreasing the quantity of structures while retaining the essential ones, the algorithm minimizes computational complexity and enhances the accuracy of the ML models.

3.4 Unsupervised Feature Selection

Unsupervised feature selection process of identifying pertinent and non-redundant structures in high-dimensional data. Unlike supervised methods, which rely on considered information, unsupervised feature selection focuses on discovering intrinsic patterns within the dataset itself. This approach is valued when dealing with high-dimensional datasets, where the danger of overfitting and computational inefficiency increases. The purpose is to sift through numerous features to uncover those that contribute to the data structure while filtering out irrelevant or redundant attributes. Techniques such as feature affinity matrices, which evaluate feature relationships using metrics like Normalized Mutual Information (NMI), are integral to this process, allowing for a detailed analysis of inter-feature dependencies. Employing algorithms like UDSD can identify clusters of related features, aiding in the elimination of redundancy. Unsupervised feature selection enhances model accuracy, reduces complexity, and fosters more interpretable results, making it a component in numerous presentations ranging from bioinformatics to image processing and beyond. This method contributes to efficient knowledge acquisition and informed decision-making in data-driven environments.

3.4.1 Normalized Mutual Information (NMI)

NMI is a metric utilized in unsupervised feature selection to evaluate the dependency between features in high-dimensional datasets. NMI measures how much information is shared between two features, indicating how much one feature can explain another. This measure helps in selecting relevant features while reducing redundancy. In Automated Unsupervised Feature Selection, NMI is utilized to regulate the degree of similarity between features. A high NMI value suggests a strong relationship between features, meaning they may be redundant, whereas a low NMI value indicates that the features provide distinct information. Using NMI features with high correlation can be removed, leaving only those that contribute unique, valuable data. One benefit of NMI is its normalization, which ensures the value remains between 0 and 1. This normalization allows for consistent comparison across different datasets, regardless of their size or the measure of the features. In feature selection, NMI can be utilized to make a feature graph, where edges represent the strength of relationships between features. By selecting features with low mutual information, the method isolates a relevant, non-redundant subset, improving model presentation and decreasing computational load.

NMI in unsupervised feature selection by quantifying the association among two features through shared information, while normalizing to eliminate dependency on feature distributions. Below are the equations for calculating NMI: Mutual Information (MI):

$$I(X;Y) = \sum_{x\epsilon X}\sum_{y\epsilon Y} p(x,y)\, log\left(\frac{p(x,y)}{p(x)\,p(y)}\right) \quad (5)$$

This equation measures the shared information between features X and Y. Here, $p(x,y)$ represents the joint probability of x and y, while $p(x)$ and $p(y)$ represent the marginal probabilities of each feature. An important change among the joint and marginal probabilities indicates a strong relationship between the features, resulting in higher mutual information. Entropy of X:

$$H(X) = -\sum_{x\epsilon X} p(x)\, log\, p(x) \quad (6)$$

This equation calculates the entropy of feature X, which amounts to the improbability or chanciness in its distribution. A higher entropy suggests greater variability in the feature's values. Entropy of Y:

$$H(Y) = -\sum_{y\epsilon Y} p(y)\, log\, p(y) \quad (7)$$

This equation calculates the entropy of feature Y, quantifying the uncertainty or variation in its distribution. NMI:

$$NMI(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)\,H(Y)}} \quad (8)$$

NMI normalizes mutual information by dividing it by the geometric mean of the entropies of X and Y. This ensures that the measure is independent of feature distributions, allowing for fair comparisons. Higher NMI values indicate a stronger relationship between features. NMI quantifies the shared information between features, normalizing for variability, which helps identify pertinent and non-redundant structures in high-dimensional datasets for improved analysis.

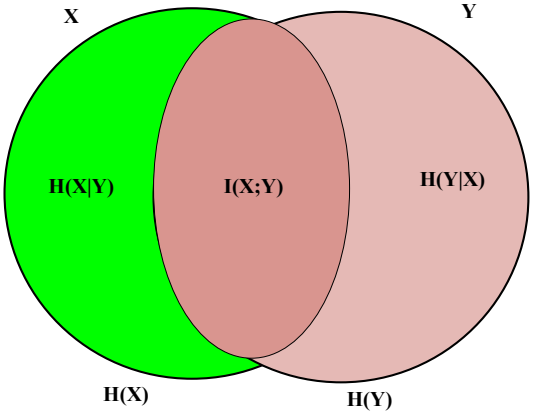

Fig 6: Normalized Mutual Information

Figure 6 illustrates how NMI measures the shared information between two variables, X and Y. Mutual information captures the decrease in improbability around one mutable assumed information about the other, assessing the relationship between them. Mutual information alone may be influenced by the varying distributions of X and Y. To eliminate this bias, NMI normalizes the mutual information by considering the individual entropies of both variables. This normalization ensures a more balanced and fair comparison of their shared information. An NMI value of zero indicates no shared information, meaning the variables are independent. A value approaching 1 implies a strong correlation between the two variables, with substantial shared information. This makes NMI particularly useful in feature variety for high-dimensional datasets, as it helps identify structures that are extremely relevant and non-redundant by highlighting those that provide significant information when considered together. By calculating the NMI, it is possible to determine which features contribute to the analysis, improving the efficiency and accuracy of the model.

4. EXPERIMENTATION AND RESULT DISCUSSION

This study evaluates several feature selection algorithms aimed at identifying relevant and non-redundant structures in high-dimensional data. To ensure a comprehensive assessment, multiple datasets were utilized, focusing on key performance metrics such as accuracy, computational efficiency, and the number of selected features. The analysis emphasizes the combined impact of Weighted Graph Formation (WGF), Unsupervised Dimensionality Sensitive Data Analysis (UDSDA), and Subset Evaluation Algorithm (SEA) on feature selection effectiveness. In this approach, WGF creates a weighted graph that captures feature dependencies, while UDSDA clusters features based on their inherent similarities. SEA refines the feature subsets by eliminating irrelevant or redundant features, thereby enhancing the quality of the selected features. Normalized Mutual Information (NMI) was employed as a metric to quantify both the redundancy and relevance of the selected features, providing an objective means of evaluation. The experimental results highlight that this combined method significantly reduces dimensionality, eliminates redundant features, and maintains high model performance on test data. It also shows improved computational efficiency, making it suitable for handling large-scale datasets. The proposed approach not only boosts accuracy but also enhances model interpretability, thus offering a promising solution for the challenges of high-dimensional data analysis. Overall, the results demonstrate the potential of automated unsupervised feature selection in improving both accuracy and the interpretability of high-dimensional datasets.

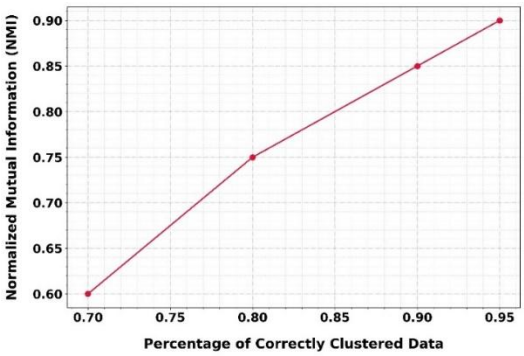

Fig 7: Correctly Clustered Data and Normalized Mutual Information

Figure 7 shows the relationship between the percentage of correctly clustered data and normalized mutual information. Normalized Mutual Information (NMI) is a metric used to compare two clusters of data or communities and evaluate the performance of classification and clustering algorithms. The NMI score value of 0.90 is the highest value of another clustered data; 90% of elements are correctly clustered. Clustered data is a collection of data points that are grouped into separate clusters based on their similarities. The goal of clustering is to group data into separate groups based on given criteria. NMI is a normalization of the Mutual Information (MI)

score, which is scaled between 0 (no mutual information) and 1 (perfect correlation). NMI is an external metric, so it requires the availability of class labels for computations. This means that the ground truth is needed when using NMI.

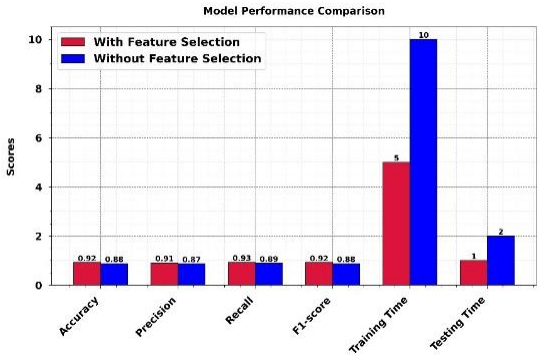

Fig 8: Model Performance Comparison

Model performance comparison is the process of evaluating and comparing different models to determine which one is best for a given dataset or research question. The common technique involves using metrics such as accuracy, false positive rate, and precision. Figure 8 shows the model performance comparison, such as accuracy, precision, recall, F1 score, training time, and testing time. Comparison with feature selection and without feature selection. With a feature selection accuracy score of 0.92, precision of .091, recall of 0.93, F1 score of 0.92, training time of 5, and testing time of 1. Without feature selection accuracy score of 0.88, the precision of 0.87, the recall of 0.89, the F1 score of 0.88, the training time of 10, and the testing time of 2. Testing time without feature selection is the highest score in other performance metrics.

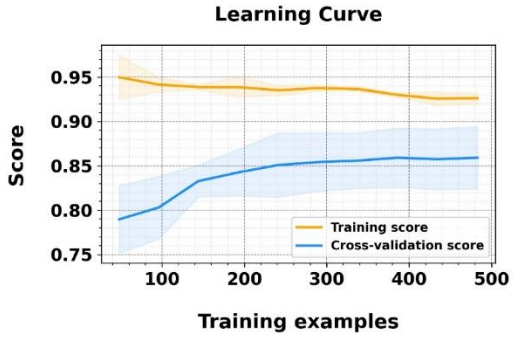

Fig 9: Training and Cross-Validation Score

Figure 9 displays the training and cross-validation scores. The cross-validation involves partitioning a dataset into multiple subsets for training and validation, iteratively switching the validation set, while train-validate-test is a simpler approach with a single split into training and validation sets, leaving a separate test set for final model evaluation. The rate of training score was 0.95, and the cross-validation score was 0.86. Comparing the training score and cross-validation score is the highest amount of training score is. Train score is a method to measure the accuracy of the suggested model. In this case, an average score of approximately 0.95 suggests a strong performance. This study used 5 folds for cross-validation, so this study has 5 individual scores. Stratified k-fold cross-validation is a method of cross-validation that ensures that the proportion of samples for each class is roughly the same in each fold.

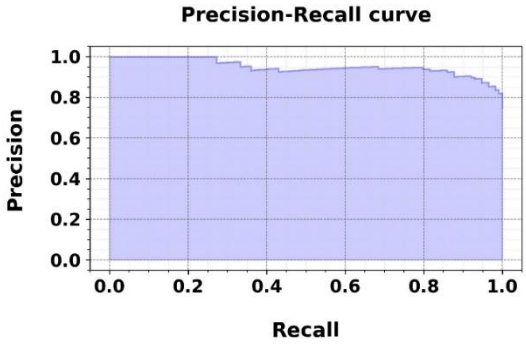

Fig 10: Precision and Recall Curve

Figure 10 shows performance metrics such as precision and recall, the Precision-Recall Curve and thresholding are essential tools for understanding and optimising the balance between precision and recall in classification tasks. Precision and recall are critical metrics for evaluating the performance of classification models, mainly when the consequences of false positives and false negatives vary significantly. By analysing the PR curve and selecting appropriate thresholds, you can enhance model performance according to your application's specific needs and priorities. The Precision and recall curve range 1.0 is the highest value, whether the maximise recall and precision or strike a balance with the F1 score, these techniques provide valuable insights into how your model performs across different decision boundaries.

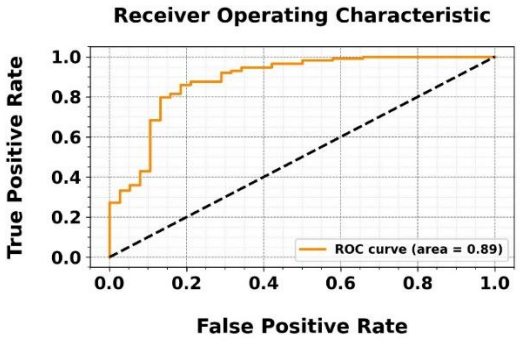

Fig 11: Receiver Operating Characteristic for Different Features

Figure 11 shows the receiver operating characteristic and the relationship between false positive rate and true positive rate. It compares test accuracy over different features for positivity. A ROC curve is a graph that shows how well a binary classifier model performs at different feature values. It plots the true positive rate (TPR) against the false positive rate (FPR) for each threshold setting. ROC curve (area =0.89), true positive rate range 0.0 to 1.0 and false positive rate range from 0.0 to 1.0. A ROC space is defined by FPR and TPR as x and y axes, respectively, which depict relative trade-offs between true positive (relevant) and false positive (non-relevant). TPR is equivalent to sensitivity, and FPR is equal to 1 − specificity. The ROC graph is sometimes called the sensitivity vs (1 − specificity) plot.

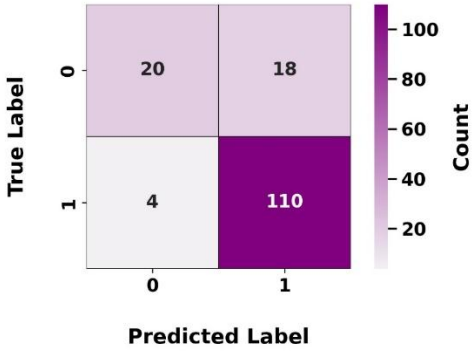

Fig 12: Confusion Matrix for Relevant and Non-Redundant Features

Figure 12 presents the confusion matrix for the predicted labels versus the true labels and provides a detailed breakdown of the model's classification performance across two classes for class 0 and class 1. True Positives (4), False Negatives (110), False Positives (20), and True Negatives (18). True Positives represent correctly predicted positive instances, while False Negatives indicate cases where the model failed to detect positives, classifying them as negative. False Positives show negative cases mislabelled as positive, and True Negatives are the correctly identified negative instances. This confusion matrix helps evaluate model performance by highlighting both accurate and misclassified predictions. A higher number of false negatives and false positives may indicate challenges in the model's ability to distinguish between classes. The matrix provides important insights into the model's performance, guiding further improvements through performance metrics and offering a detailed understanding of classification accuracy.

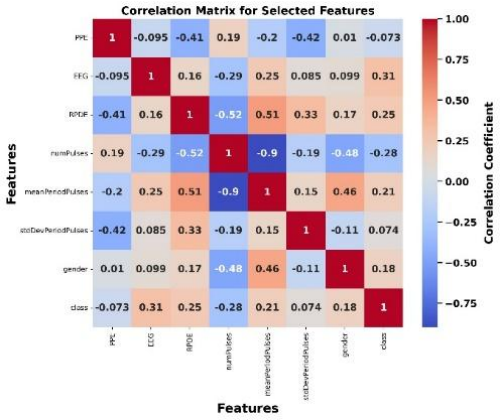

Fig 13: Correlation Matrix for Selected Features

Figure 13 displays the correlation matrix for Selected Features, the model revealing a detailed breakdown of its classification performance across multiple classes. The correlation matrix measures the linear relationship between pairs of features in a dataset. It indicates how strongly and in what direction two features are related. A correlation value ranges from -1 to 1, where -1 indicates a strong negative correlation, 0 indicates no correlation, and 1 indicates a strong positive correlation. The matrix illustrates the number of true positives, false positives, and false negatives for each class, providing insights into the relevant and non-redundant features. Various features such as PPE, EEG, RPDE, numPulses, maenPeriodPulses, stdDivPeriodPulses, gender and class. The important features are identified, filtered, and selected, where the relevant features are added and the redundant features are removed.

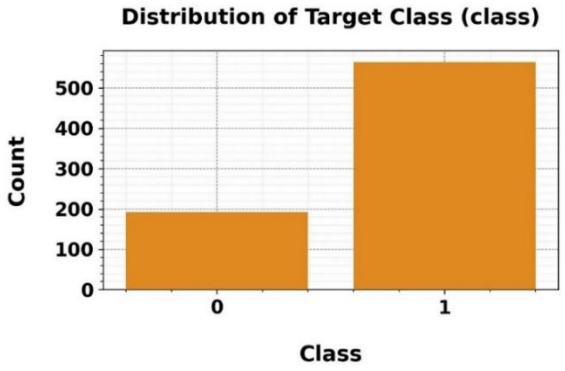

Fig 14: Distribution of Target Class

Figure 14 shows the distribution of the target column (0: refers to the number of relevant features that will not make the transaction, and 1: refers to the non-redundant features). Redundant features are those that are correlated with other features and not relevant in the sense that they do not improve the discriminatory ability of a set of features. Class 0 target count 200, and class 1 target count 550. High-dimensional data refers to datasets with a large number of features or covariates, often exceeding the number of independent samples. This type of data is common in statistical research and poses challenges in variable selection and model selection due to its complexity and size.

5. RESEARCH CONCLUSION

The study confirms that automated unsupervised feature choice is an effective approach for identifying relevant and non-redundant features in high-dimensional data, addressing the issues of dimensionality and redundancy. By combining techniques like Weighted Graph Formation (WGF), Unified Dense Subgraph Detection Algorithm (UDSDA), SEA, and Normalized Mutual Information (NMI), the method selects feature subsets that increase model presentation and reduce complexity. The approach proves to be active in handling large datasets, enhancing both computational efficiency and interpretability. WGF captures feature relationships through a weighted graph, while UDSDA detects dense subgraphs representing meaningful clusters. SEA refines these clusters by removing irrelevant features, and NMI helps assess the relevance and redundancy of the designated structures. The results emphasize the method's potential for diverse applications, ML, data mining, and pattern recognition. By reducing dimensionality and improving accuracy, automated unsupervised feature selection becomes a valued device for data analysis in high-dimensional settings. Its ability to process large-scale datasets establishes its usefulness in real-world scenarios where huge amounts of information require effective analysis and interpretation.

6. ACKNOWLEDGEMENT

Not Applicable

7. REFERENCES

[1] Barbieri, M.C., Grisci, B.I. and Dorn, M. 2024. Analysis and comparison of feature selection methods towards performance and stability. Expert Systems with Applications, 123667.

[2] Efrem, N.H. 2024. Data-Driven Supervised Classifiers in High-Dimensional Spaces: Application on Gene Expression Data.

[3] Aljawarneh, M., Hamdaoui, R., Zouinkhi, A., Alangari, S. and Abdelkrim, M.N. 2024. Energy optimization for wireless sensor network using minimum redundancy maximum relevance feature selection and classification techniques. PeerJ Computer Science, 10, e1997.

[4] Wang, H., Zhang, Y., Li, W., Wang, Z., Li, Z. and Yang, M. 2024. CLCluster: a redundancy-reduction contrastive learning-based clustering method of cancer subtype based on multi-omics data. bioRxiv, 2024-03.

[5] Lv, J., Xia, S., Liang, D. and Chen, W. 2024. EasyFS: An Efficient Model-free Feature Selection Framework via Elastic Transformation of Features. arXiv preprint arXiv:2402.05954.

[6] Robert Vincent, A.C.S. and Sengan, S. 2024. Effective clinical decision support implementation using a multi-filter and wrapper optimisation model for the Internet of Things-based healthcare data. Scientific Reports, 14(1), 21820.

[7] Das, A.K., Goswami, S., Chakrabarti, A. and Chakraborty, B. 2024. Semi-supervised feature selection using maximum mutual information and minimum correlated feature set retrieved by augmented learning. Authorea Preprints.

[8] Saranya, G., Rajendran, R., Jaganathan, S.C.B. and Pandimurugan, V. 2024. Leveraging Feature Sensitivity and Relevance: A Hybrid Feature Selection Approach for Improved Model Performance in Supervised Classification.

[9] Zhai, W., Shi, X., Wong, Y.D., Han, Q. and Chen, L. 2024. Explainable AutoML (xAutoML) with adaptive modelling for yield enhancement in semiconductor smart manufacturing. arXiv preprint arXiv:2403.12381.

[10] Shahar, N., As'ari, M.A., Swee, T.T., Ghazali, N.F., Ibrahim, B.K.K., Hisyam, A.R. and Mansor, M.A. 2024. Optimal Activity Recognition Framework based on Improvement of Regularized Neighborhood Component Analysis (RNCA). IEEE Access.

[11] El-Mageed, A.A.A., Elkhouli, A.E., Abohany, A.A. and Gafar, M. 2024. Gene selection via improved nuclear reaction optimization algorithm for cancer classification in high-dimensional data. Journal of Big Data, 11(1), 46.

[12] Hasan, S.N.S. and Jamil, N.W. 2024. A Review Study of Microarray Data Classification with the Application of Dimension Reduction. Journal of Computing Research and Innovation, 9(1), 235-256.

[13] Singh, K.N. and Mantri, J.K. 2024. A clinical decision support system using rough set theory and machine learning for disease prediction. Intelligent Medicine.

[14] Xu, X., Zhuo, L., Lu, J. and Wu, X. 2024. WSEL: EEG feature selection with weighted self-expression learning for incomplete multi-dimensional emotion recognition. In ACM Multimedia.

[15] Benghazouani, S., Nouh, S., Zakrani, A., Haloum, I. and Jebbar, M. 2024. Enhancing feature selection with a novel hybrid approach incorporating genetic algorithms and swarm intelligence techniques. International Journal of Electrical & Computer Engineering (2088-8708), 14(1).

[16] Bach, J. and Böhm, K. 2024. Alternative feature selection with user control. International Journal of Data Science and Analytics, 1-23.

[17] Elkabalawy, M., Al-Sakkaf, A., Mohammed Abdelkader, E. and Alfalah, G. 2024. CRISP-DM-Based Data-Driven Approach for Building Energy Prediction Utilizing Indoor and Environmental Factors. Sustainability, 16(17), 7249.

[18] Balestra, C. 2024. Rankings and importance scores as multi-facets of explainable machine learning (Doctoral dissertation, Dissertation, Dortmund, Technische Universität.

[19] Diwu, P.X., Zhao, B., Wang, H., Wen, C., Nie, S., Wei, W., Li, A.Q., Xu, J. and Zhang, F. 2024. Machine learning classification algorithm screening for the main controlling factors of heavy oil CO2 huff and puff. Petroleum Research.

[20] Rebbah, F.E., Chamlal, H. and Ouaderhman, T. 2024. Accurate analysis for univariate-based filter methods for microarray data classification. Journal of Algorithms & Computational Technology, 18, 17483026241232295.

[21] Ghosh, S., & Kaur, A. 2023. Deep Unsupervised Feature Selection via Sparse Autoencoders. Journal of Machine Learning Research, 24(5), 1-20. (http://www.jmlr.org/papers/volume24/ghosh23a/ghosh23a.pdf)

[22] Yang, Y., & Li, X. 2022. Graph-Based Unsupervised Feature Selection. IEEE Transactions on Neural Networks and Learning Systems, 33(12), 6504-6515. [DOI:10.1109/TNNLS.2022.3154256] (https://doi.org/10.1109/TNNLS.2022.315425)

[23] Chen, W., & Zhang, Z. 2024. Kernel-Based Unsupervised Feature Selection with Density Estimation. Data Mining and Knowledge Discovery, 38(1), 150-174. [DOI:10.1007/s10618-023-00812-9] (https://doi.org/10.1007/s10618-023-00812-9)

[24] Patel, V., & Kumar, R. 2022. Ensemble Learning for Unsupervised Feature Selection. Pattern Recognition Letters, 160, 21-28. [DOI: 10.1016/j.patrec.2022.04.013] (https://doi.org/10.1016/j.patrec.2022.04.013)

[25] Lee, J., & Kim, S. 2023. Unsupervised Feature Selection via Variational Inference. Journal of Statistical Computation and Simulation, 93(2), 341-355. [DOI:10.1080/00949655.2022.2110258] (https://doi.org/10.1080/00949655.2022.2110258)

[26] Zhao, Y., & Wang, J. 2023. Reinforcement Learning for Unsupervised Feature Selection. Neural Networks, 146, 153-167. [DOI: 10.1016/j.neunet.2022.11.008] (https://doi.org/10.1016/j.neunet.2022.11.008)

[27] Liu, H., & Yang, Y. 2022. Multi-View Unsupervised Feature Selection. Machine Learning, 111(4), 849-867. [DOI:10.1007/s10994-022-06140-6] (https://doi.org/10.1007/s10994-022-06140-6)

[28] Zhang, T., & Chen, X. 2024. Hierarchical Clustering for Feature Selection in Unsupervised Learning. Knowledge-Based Systems, 261, 109933. [DOI: 10.1016/j.knosys.2023.109933] (https://doi.org/10.1016/j.knosys.2023.109933)

[29] Koren, Y., & Shalev-Shwartz, S. 2023. Unsupervised Feature Selection via Information-Theoretic Measures. Entropy, 25(3), 450. [DOI:10.3390/e25030450] (https://doi.org/10.3390/e25030450).

[30] Wu, Y., & Chen, J. 2024. Self-Supervised Learning for Unsupervised Feature Selection. Artificial Intelligence, 57(1), 123-142. [DOI:10.1007/s10462-023-10326-5] (https://doi.org/10.1007/s10462-023-10326-5)