SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I11P111

Volume 11 Issue 11, 100-113, November 2024
© 2024 Seventh Sense Research Group®

Original Article

CNI-VIF: Enhanced Feature Selection for Graph Databases by Integrating Composite Node Information in VIF

Anagha Patil[1], Arti Deshpande[2]

[1]*TSEC, Maharashtra, India.*
[2]*Computer Engineering Department, TSEC, Maharashtra, India.*

[1]*Corresponding Author : anagha.patil@vcet.edu.in*

Received: 07 September 2024 Revised: 08 October 2024 Accepted: 06 November 2024 Published: 30 November 2024

Abstract *- Feature selection and dimensionality reduction are critical techniques in today's data-centric world, where vast and complex datasets necessitate efficient and effective methods for analysis and decision-making. In this research, an enhanced feature selection technique, Composite Node Information - Variance Inflation Factor (CNI-VIF), tailored for graph databases, which particularly focuses on network traffic datasets, is proposed. Traditional feature selection methods often fail to adequately capture the complex interrelationships in graph data. The proposed method incorporates Composite Node Information (CNI), an aggregate of Betweenness, Closeness, and Degree centrality, into the VIF framework to address these limitations. By integrating CNI, the proposed method not only improves the selection of graph-based features but also achieves dimensionality reduction and decreased computation time, making the feature selection process more efficient. Experiments conducted on CTU-13, IoT-23, and NCC-2 datasets demonstrate that CNI-VIF significantly outperforms traditional methods by effectively selecting graph-based features, thus enhancing the performance of machine learning models. Specifically, the Random Forest algorithm shows exceptional results among all feature selection techniques, with CNI-VIF yielding the best performance overall. The results indicate that CNI-VIF is particularly effective for graph databases, offering a robust and efficient feature selection mechanism that enhances model computation and predictive accuracy.*

Keywords *- CNI, CNI-VIF, Graph database, Feature selection, VIF.*

1. Introduction

In today's data-driven world, the absolute volume and data complexity present substantial challenges for analysis and decision-making. Social networks, e-commerce websites, online applications, communication systems, and other technologies generate "big data", which is complex, valuable, structured, and unstructured. Processing big datasets with high-dimensional feature space is a crucial problem. Nonetheless, a large number of features are frequently present in the training data of real-world classification applications [1].

They might contain some redundant or unnecessary features, which would lower the performance of the resulting classifier as well as the training efficiency [2]. One of the most crucial methods for eliminating extraneous or irrelevant features from the original feature collection is dimensionality reduction. Feature selection and dimensionality reduction have become crucial techniques to address these challenges, particularly in the context of machine learning and data science [3, 4].

These methods aim to identify the most relevant features from large datasets, reducing the dimensionality while retaining the essential information needed for accurate and efficient modelling. This not only enhances the performance of predictive models but also reduces computational costs, storage requirements, and the risk of overfitting. Dimensionality reduction techniques can improve application method performance, reduce computational costs, and prevent overfitting issues when processing high-dimensional information.

The two primary approaches of dimensionality reduction used today are extracting features and selecting required features [4, 5]. Feature extraction helps to reduce dimensionality by converting the original feature space into a compact space. Feature selection eliminates superfluous or unnecessary characteristics when selecting a subset of the features compared to feature extraction. Four main categories can be used to categorize feature selection: filter, wrapper, hybrid, and embedded techniques [6, 7].

Wrapper techniques have a high processing cost despite being able to produce good classification results. Filter techniques are widely employed in practical applications due to their effortless and practical computations. The greatest elements from filter and wrapper approaches are combined in hybrid strategies. Embedded techniques use feature selection during the learning algorithm training phase to find the optimal feature subset. Filter techniques that are simpler to build, more generalizable, and independent of the learning model are more suitable for processing high-dimensional data than other approaches.

Graph databases [8], representing data as interconnected nodes and edges, are increasingly used in various domains. In network traffic, relationships and interactions between IP addresses, devices, or nodes resemble a graph structure, where nodes represent entities (such as devices or IPs), and edges represent interactions (like connections or traffic flows). Graph databases can directly model these relationships, making it easier to visualize and analyse the network's structure. When it comes to storing and retrieving data, graph databases perform best in scenarios where the majority of the data is linked, such as social media, geolocation, networks, cybersecurity and biomedical data [9, 26].

Traditional feature selection techniques often fall short in such databases as they do not fully leverage the structural information inherent in graph data. Features derived from graph properties can be used to get insights such as the importance and influence of nodes within the network. However, integrating these graph-specific features with conventional feature selection methods remains challenging. Researchers focused on developing graph-based feature selection techniques, which involve projecting intricate multi-way feature interactions into a feature graph and using various graph-theoretic concepts to choose final feature subsets, which have proven superior to the classic feature selection methods [9, 10].

Traditional feature selection methods do not account for graph-specific features like centrality measures, community structures, or connectivity patterns, which are crucial for understanding the underlying graph topology. Applying existing feature selection techniques to large-scale graph databases can be computationally intensive, especially if the graph has a high degree of connectivity and numerous features. This necessitates the development of modified techniques that incorporate graph-specific features and account for the complex relationships inherent in graph data. When compared to static feature graphs, the dynamic feature graph that was created was able to yield feature subsets of greater quality and more closely resemble the ideal feature graph [11].

An enhanced approach called Composite Node Information-Variance Inflation Factor (CNI-VIF) is proposed to bridge this gap, integrating graph-based features with the traditional Variance Inflation Factor (VIF) framework. By incorporating Composite Node Information (CNI) derived from centrality measures such as betweenness, closeness, and degree centrality, CNI-VIF aims to provide a more comprehensive and interpretable measure of multicollinearity in graph databases. This approach not only enhances feature selection but also improves the overall performance and efficiency of machine learning models applied to graph-based data.

The methodology used in this paper explores the effectiveness of the CNI-VIF approach by comparing it with traditional VIF, Principal Component Analysis (PCA), and Recursive Feature Elimination (RFE) methods. Experiments are conducted using three benchmark network traffic datasets: CTU-13 [12], IoT-23 [13], and NCC-2 [14] to measure the performance of the proposed method. This study underscores the importance of incorporating graph-specific characteristics in feature selection methodologies and positions CNI-VIF as a powerful tool for enhancing analysis and decision-making in graph databases. To be precise, the key contributions in this paper can be abridged as follows:

1) An aggregate graph-based feature called “CNI” is introduced, which contains an average of three important centralities in a network.
2) Based on the “CNI”, the original VIF feature selection algorithm is modified to make it more appropriate for graph databases.
3) The efficacy of the proposed CNI-VIF is tested on three network traffic data sets of varying dimensions. The results demonstrate that CNI-VIF consistently selects more relevant features and achieves superior model accuracy and computational efficiency.

The remainder of the paper is systematized as follows. The preliminary feature selection approaches, and the associated work on graph-based feature selection algorithms are specified in Section II. Traditional VIF, centrality measures and the proposed CNI-VIF are accessible in Section III. Then, in Section IV, we analytically verify the efficacy of the CNI-VIF by comparing it with the three latest feature selection algorithms with respect to graph-based feature selection, various performance metrics and running time. Finally, in Section V, a conclusion and a deliberate discussion of future work is provided.

2. Literature Review

The technique of removing unnecessary and significant attributes from a dataset to expedite processing is known as feature selection. Traditional feature selection techniques can be separated into behaviours for node-level, edge-level, and graph-level feature extraction [23]. Information such as the density of the node's neighbourhood cluster and node centrality can all be encoded by node-level attributes. Graph-level features can be used to wrench out information through counting the presence of various trivial subgraph structures, iterative neighbourhood aggregation or accumulating statistics/features from all nodes inside a graph. Edge-level feature techniques extract features by counting the number of paths of all lengths between two nodes or the count of neighbours that two nodes share. Graph-based feature selection methods proved superior to existing feature selection methods due to the capability of considering one-way or two-way relations among the features.

At present, the graph-based feature selection techniques can be broadly divided into various classes, including graph-lasso-based [15, 17], graph-clustering-based [16] and evolutionary computation graph-based [18, 19]. Moreover, these approaches can be applied in a supervised, unsupervised and hybrid manner.

The research [15] proposed a Graph Regularized Feature Selection with Data Reconstruction (GRFS) approach for feature selection. The approach formulates the feature selection problem in an unsupervised manner with the view of data reconstruction, where the objective is to choose features that maintain the discriminant information and similarity with the original data space. The approach uses a joint framework that integrates data reconstruction and graph regularization, and to solve the optimization problem, it uses a gradient method. The authors claimed that the proposed method can be stretched to supervised feature selection by incorporating label information into the graph regularization.

The authors [16] note that the number of subgraph features can be extreme depending on the threshold of the frequent pattern mining algorithm. The Incremental Subgraph Feature Selection (ISF) algorithm leads classifiers to generate long-pattern subgraph features to avoid bias towards short-patterns that form a sequence of primal-dual solutions that shrink the dual gap and render an improved solution towards the optimum. The Incremental Subgraph Join Feature Selection (ISJF) algorithm helps classifiers to generate long-pattern subgraph features. The study also discusses the limitations of existing graph classification methods and proposes a max-margin graph classifier using the proposed algorithms.

The authors [17] proposed an innovative feature selection method, Dual-graph regularized Feature Selection Clustering (DFSC), which maintains local geometrical structure by building neighbourhood graphs in feature and data spaces using the self-representation property. The authors also analyse the sensitivity of DFSC to parameters and compare it with co-clustering algorithms on the COIL20 dataset, showing that DFSC attains the best clustering results by utilizing both data and feature manifolds and selecting the most effective features. The effectiveness of DFSC is further validated using the "Ionosphere" dataset, revealing that the coefficients of the actual features are significantly greater than those of new features in the coefficient's matrix P.

Zhiwei Hu et al. [20] and T. B. Mudiyanselage [21] proposed graph-based feature selection approaches. [20] is built on the concept of feature clusters and uses a graph structure to represent the correlation between features and labels. At the same time, [21] aims to address the limitations of existing methods by considering feature dependencies and performing well in high-dimensional feature spaces. Future work includes investigating the similarity criteria between continuous value attributes and discrete value attributes, as well as refining the method to reconcile the conflict between high-dimensional data and computational complexity.

The study [22] employs a graph kernel-based Structured Feature Selection (gk-SFS) technique for brain disease classification using Functional Connectivity Networks (FCNs) constructed on resting-state functional Magnetic Resonance Imaging (rs-fMRI) data. The authors also investigate the effects of regularization parameters and thresholds on the performance of the projected method and validate the identified brain regions with previous studies.

Giorgio Roffo et al. [7] present a framework for filtering feature selection that handles relevance and redundancy principles by considering a subset of features as a path in a graph, wherein a node represents a feature, and an edge denotes interactions among features. The framework is evaluated on eleven different publicly available datasets to study the pros and cons of the unsupervised and supervised Inf-FS.

Afnan Alharbi et al. [6] worked on CTU-13 and IoT-23 datasets for graph-based botnet detection. The authors used supervised machine learning algorithms to evaluate proposed methods with filter-based feature evaluation metrics. This approach detects numerous botnet families with an increased botnet detection rate in reduced time. The authors suggest using attribute features along with structural features of the network for better botnet detection.

The research [23] aims to propose a graph-based feature selection method, GBFS-SND, which constructs a dynamic feature graph and utilizes a multi-objective evolutionary technique to optimize its structure and nodes, leading to a high-quality feature subset. The paper also investigates the efficacy of GBFS-SND on twelve real-world datasets and compares it with existing algorithms, demonstrating its superiority in terms of accuracy and selected features.

The study [24] aims to address the challenges of modelling high-dimensional spectral data with the help of the Mutual Information-Variance Inflation Factor (MI-VIF). While [16, 20] focus on the structure of the network, [24] does not consider it. Ling Zheng et al. [25], Fan Cheng et al. [23]

and Consolata Gakii et al. [26] aim to propose a graph-based framework for feature selection in real-world datasets, addressing the limitations of traditional FS methods. The authors identify several potential areas for future research, including investigating other search mechanisms for feature selection, improving feature grouping strategies, and applying the method to deep features. These opportunities aim to enhance the performance and flexibility of the proposed feature grouping framework. A detailed summary of the literature is presented in Table 1.

Table 1. Summary of literature review

Literature	Feature Selection Approach	Dataset(s)	Contribution	Gap
[15]	Graph-clustering based (Unsupervised)	TDT2, Routers document corpora	Features are chosen based on how well they maintain discriminant information and similarity in the original data space.	The algorithms cannot handle high dimensional sub-graph feature space and more complex graph structures.
[16]	Graph-based (Supervised)	Albert-Barabasi, Forest Fire, Small World, Erdos-Renyi, DBLP, MemeTracker	A primal-dual incremental subgraph feature selection algorithm (ISF) and a subgraph join feature selection algorithm (ISJF) are proposed.	
[17]	Graph lasso-based (Unsupervised)	Umist, Dbworld_bodies, Isolet, Sonar, ORL, BC, Ionosphere, Dbworld_bodies	By maintaining local geometrical information in both the feature spaces and the data, DFSC efficiently chooses the best representative features for clustering.	DFSC removes related features after the feature selection process to avoid redundancy.
[20]	Graph-based (Supervised)	Wine, Dermatology, Sonar, Wdbc, Parkinsons, Ionosphere, Lung, Hill Valley	The correlation between features and labels filters out weakly correlated features, and the remaining features are used to construct a graph.	The algorithm cannot handle high-dimensional data.
[21]	Graph-based (Supervised + Unsupervised)	COIL20, ORL	The method constructs a graph based on feature similarity and uses the Markov chain process to calculate each feature's score.	New similarity calculation methods are required to better represent the original feature space and find the optimal feature set.
[22]	Graph-Kernel based (Supervised)	ADNI, ADHD-200	The structural information of networks is preserved, and the learning performance is improved by taking advantage of the local-to-global structural information.	The methods can be integrated with other machine learning algorithms for enhanced classification.
[7]	Graph-based (Supervised + Unsupervised)	Colon, Lymphoma, Leukemia, Lung, Prostate	The method creates pathways of variable lengths that eventually grow to infinity by using the properties of a power series of matrices and depending on the principles of Markov chains.	

[6]	Graph-based (Supervised)	CTU-13, IoT-23	Using consistency, correlation and information, feature evaluation is derived.	Only structural features of the network are considered.
[26]	Graph-based (Supervised)	GSE60052, GSE81089	Graph-based feature selection is used along with association miming to find associations between features in RNAseq data.	
[23]	Graph-based (Supervised)	12 datasets	High-quality feature subset was obtained by dynamically adjusting the feature graph's node and structure.	The candidate graph's graph structure is modified by eliminating a few links.
[24]	Graph-based (Supervised)	Tea dataset, Diesel fuels dataset	The objective is to maximize the correlation between independent variables and the response variable while minimizing collinearity amongst selected variables.	The mutual Information-Variance Inflation Factor algorithm does not focus on the graph's structure.
[25]	Graph-based (Supervised)	20 datasets	The framework constructs a graph based on feature interactions and uses Kruskal's algorithm to build a minimum spanning tree, which is then used to form feature groups.	Feature grouping strategies can be improved.

3. Methodology

When working with datasets enriched with graph-based features, traditional methods for calculating VIF can be adapted to better capture the complex relationships inherent in graph data. While traditional VIF can detect multicollinearity, it does not consider the structural properties of the graph, which can be critical in network data. The Modified VIF, referred to as CNI-VIF, aims to incorporate graph-based metrics to provide a more nuanced understanding of multicollinearity.

3.1. Traditional VIF

VIF [27] is a metric that illustrates the extent to which multicollinearity in the data increases the variance of an estimated regression coefficient. High VIF values can indicate multicollinearity but might also lead to overfitting if used excessively for feature elimination. Despite its effectiveness in traditional data structures, VIF does not inherently account for the interconnected nature of graph data. Traditional VIF ignores the contextual information provided by graph-based features, which can be crucial in detecting patterns like botnet behaviour in network data.

The traditional VIF for a feature X_i is calculated to assess how much the variance of its estimated coefficient is increased due to multicollinearity with other features.

The steps to calculate VIF are:

- Fit a regression model where X_i is the dependent variable, and all other features are the independent variables.
- Compute R^2 for this regression, denoted as R_i^2.
- Calculate VIF using:

$$\mathrm{VIFi} = \frac{1}{1-\mathrm{Ri}^2} \tag{1}$$

This formula measures the degree of multicollinearity; a higher VIF indicates greater collinearity. The results of regression are not invalidated by greater values of the VIF. Greater values indicate the removal of one or more independent variables or the need to combine them into a single index. So, Composite Node Information (CNI) is considered a single variable.

3.2. Centrality Measures

When graph theory is used for network analysis, centrality measures play an important role as they capture the importance of nodes in terms of connections, communications, and relationships [6, 8]. Hence, these measures can be helpful to discriminate between normal nodes and bots. Some important centrality measures are explained below:

3.2.1. Betweenness Centrality (BC)

The Betweenness Centrality of any node indicates the number of times that node resides on the shortest path through others. As stated differently, BC indicates the frequency with which a node spans the shortest path between two other vertices. A high BC count simply indicates that a particular node embraces authority over different clusters in a network or that both nodes are on the border of both clusters.

3.2.2. Closeness Centrality (CC)

This measure calculates the shortest paths between all nodes and then assigns each node a score based on its sum of shortest paths. In a densely connected network, CC can assist in identifying quality "broadcasters."

3.2.3. Degree Centrality (DC)

Degree Centrality measures how well a node is connected. It assigns a score based on the node's communication with others in the network. DC is used to find popular connected nodes, nodes that hold the greatest information or nodes that can rapidly connect with the broader network.

3.2.4. Eigen Centrality (EC)

In addition, EC considers a node's degree of connectivity, the number of hyperlinks that connect it to other nodes, and so on through the network.

3.2.5. PageRank (PR)

An additional method of assigning a score to nodes based on their connections and their neighbours is PageRank, a variation of EC. The distinction is that PageRank considers both the direction and weight of connections, meaning that links can only carry varying degrees of influence in one direction.

3.3. Composite Node Information (CNI)

In this research, the focus is on graph-based network traffic data. So, to enhance feature selection in network data, we calculate the Composite Node Information (CNI) as an aggregate measure of three key centrality metrics: Betweenness Centrality (BC), Closeness Centrality (CC) and Degree Centrality (DC). These metrics provide a holistic view of a node's significance within the network.

$$CNI = \frac{BC+CC+DC}{3} \qquad (2)$$

This adjustment reflects the added complexity or collinearity introduced by the network structure.

In a network context, interactions between nodes (e.g., data packets sent from a source IP to a destination IP) can be characterized by the properties of both participating nodes. Incorporating separate CNI values for both source and destination IPs would effectively double the number of features related to CNI. This could increase the complexity of the model and the computational burden, especially in large datasets.

By using the average_CNI ($\overline{CNI}$), the dimensionality of the entire feature space is reduced, simplifying the model while still capturing the essential characteristics of the node interactions. It's useful to combine the information from both ends of the connection to provide a comprehensive representation of these interactions.

The $\overline{CNI}$ serves this purpose by aggregating the CNI values of the source and destination nodes into a single metric, reflecting the overall influence or importance of the interaction within the network. $\overline{CNI}$ captures the centrality or importance of nodes (features) within the graph, indicating how influential a node is within the network.

3.4. CNI-VIF for Graph Databases

The traditional VIF is modified by integrating the $\overline{CNI}$, scaled by a parameter α:

$$CNI - VIFi = \frac{1}{1-Normalised(Ri^2 + \alpha * \overline{CNI})} \qquad (3)$$

Where:

- R_i^2 is the unadjusted coefficient of determination for regressing the ith independent variable on the remaining ones.
- $\overline{\text{CNI}}$ is the mean value of the average_CNI for the dataset.
- α is a parameter that adjusts the influence of the graph-based feature.

To ensure that CNI-VIF values are on a comparable scale and to prevent any undue influence of outliers, we normalize $(Ri^2 + \alpha * \overline{CNI})$ using Min-Max normalization. Since both Ri^2 and $\overline{CNI}$ are in [0,1], their sum, scaled by non-negative factor α, is also bounded.

Explicitly:

$$0 \leq Normalised(Ri^2 + \alpha * \overline{CNI}) < 1$$

The strict inequality <1 ensures that the denominator of the CNI-VIF equation:

$$1 - Normalised(Ri^2 + \alpha * \overline{CNI})$$

Is always positive and never zero. This positive value ensures that the denominator does not approach zero, thus preventing the CNI-VIF from becoming infinite or undefined. Since the denominator is strictly positive, the CNI-VIF is finite.

```
Algorithm for CNI-VIF:
Input    X: DataFrame containing all features,
         including graph-based features (average_CNI),
         α: Parameter controlling the weight of
         average_CNI.
Output   CNI-VIF_i for each feature X_i, Selected features
Steps
  1:     Calculate the average centrality (average_CNI)
         across the dataset.
  2:     Calculate $(Ri^2 + \alpha * \overline{CNI})$ , scaled by the
         parameter $\alpha$.
  3:     If the component $(Ri^2 + \alpha * \overline{CNI})$ > 1, then
         normalize this component to ensure it is within
         [0,1].
  4:     Repeat Step 1 – Step 3 for each feature X_i.
  5:     Return CNI-VIF_i for each feature X_i.
```

Let's consider an example where the betweenness centrality feature is added in the CTU-13 dataset. When this feature is added to the dataset, as the source and destination IPS are provided, which represent nodes in a graph database, it is necessary to calculate betweenness centrality for both of them.

The feature "orig_betweenness" represents betweenness centrality for source ip. After applying VIF and CNI-VIF to the CTU-13 dataset, the following values are obtained as the output:

VIF : inf (infinite)
CNI-VIF : 9.893454

If the traditional VIF value for a feature (in this case, orig_betweenness) is infinite, it indicates perfect multicollinearity among the predictor variables. This means that orig_betweenness can be perfectly predicted from the other features in the dataset, and the feature is eliminated. An infinite VIF value typically occurs when the coefficient of determination Ri^2 for the regression of orig_betweenness on the other features is 1.0.

When α is assumed as 1, a CNI-VIF value of 9.893454 indicates high, but not perfect, multicollinearity when accounting for both traditional and graph-based features. The substantial but finite CNI-VIF suggests that orig_betweenness is still strongly influenced by other features, but the inclusion of CNI has alleviated the severity of perfect multicollinearity, providing a more nuanced understanding of feature relationships to sustain the feature. In practice, features like "orig_betweenness" might show a high CNI-VIF (indicating multicollinearity with other graph features) while exhibiting a non-significant traditional VIF. This showcases CNI-VIF's ability to detect dependencies missed by VIF, particularly those relevant to the graph's structural context.

4. Experiments

Here, the performance of the proposed CNI-VIF algorithm is analytically validated by comparing it with several benchmark algorithms. The network traffic datasets and benchmark algorithms used in the experiments are introduced (Section 4.1). Two data frames for every dataset are used here:

1) one with centrality features of graph to show the effectiveness of our proposed method on graph-based datasets, and
2) another with our composite parameter "CNI" instead of these centrality features to show reduced dimensionality and reduced computation time.

Then, the results after comparing the proposed algorithm and benchmark algorithms are outlined with the help of evaluation metrics such as accuracy, precision, recall and F1 score (Section 4.2). Additionally, to prove the dominance of the proposed method, all the listed techniques on the data sets of varying dimensions are compared with respect to the number of selected features, including graph features, and the running time (Section 4.2).

4.1. Datasets

The proposed algorithm is tested on three datasets containing network traffic for botnet detection. Graph databases can directly model these relationships, making it easier to visualize and analyse the network's structure. The experiments are conducted to,

1) test the proposed method on network traffic datasets to ensure dimensionality reduction with the introduction of composite variable "CNI";
2) compare CNI-VIF algorithm with other feature selection methods to validate the performance in terms of performance metrics;
3) compare CNI-VIF algorithm with listed feature selection methods with respect to execution time; and
4) Additionally, to show the pre-eminence of the proposed method, all the listed algorithms are compared on the data sets of varying dimensions.

The code and data sets are available online [28]. The algorithms and their evaluation are implemented in Python with the help of various ML and DL methods. All experiments are executed on a GPU with Nvdia GeForce Trx 4080 and 16GB memory.

To test the efficiency of the CNI_VIF approach, CTU-13, IoT-23 and NCC-2 datasets are used. CTU-13 contains PCAP files from infected ips formed by CTU University. Thirteen scenarios comprising both malicious and benign traffic from different botnet families are included in the CTU-13 dataset. The IoT-23 dataset is designed for machine learning research on IoT-based cyber-attacks.

The NCC-2 dataset was released in 2022 with the goal of capturing both periodic attacks on NCC [29] and irregular attacks on CTU-13. Various botnet tools are used in all three datasets to contain botnet traffic. Thus, the tuples in the datasets contain normal, background, and botnet traffic. The datasets are available for free download on the official websites and have been used in several research studies and competitions.

Table 2. List of datasets

Dataset	Size	No. of Features	No. of Nodes	Edges
CTU-13	1.9 GB	17	52749	801132
IoT-23	21 GB	21	1097904	1446639
NCC-2	866 MB	18	8559	997757

The detailed statistics of the data sets are provided in Table 2. Table 2 shows that the feature count in these datasets varies from 17 to 21. For every data set, the data is split randomly into two sets: 30% is used as the test set and 70% is used as the training set. The Accuracy, Precision, Recall, F1 score and feature subset (how many graph-based features are still sustained) after feature selection and running time of algorithms on the test data are reported as the experimental outcomes.

4.2. Experimental Setup and Results

Experiments are being carried out to determine whether the CNI-VIF approach is effective. The performance of CNI-VIF is compared with the three latest feature selection techniques, which are VIF, PCA [30], and RFE [31]. VIF is a measure used to detect multicollinearity among predictor variables in a regression model.

PCA reduces dimensional space by transforming a large set of variables into a smaller set of uncorrelated components. It is widely used in machine learning and data analysis to reduce the dimensionality of data. RFE is a feature selection method that recursively removes the least significant features based on model performance, aiming to select the important set of features.

Based on the listed data sets, CNI-VIF is compared with three algorithms that consider only graph-based features that focus on centrality. As these datasets mainly contain network traffic, three graph-based features are considered: BC, CC, and DC.

All the datasets contain source and destination ips, which are further converted into nodes, so six centralities are added as graph-based features for BC, CC and DC, respectively, namely:

- orig_betweenness,
- resp_betweenness,
- orig_closeness,
- resp_closeness,
- orig_degree,
- resp_degree.

Table 3. Feature selection obtained by all comparison algorithms for CTU-13

Sr. No.	Features		Feature Selection Algorithms			
			CNI-VIF	VIF	PCA	RFE
1	Original Features	Dur	√	√	√	√
2		Proto				
3		SrcAddr	√			
4		Sport	√	√	√	√
5		Dir				
6		DstAddr	√	√	√	
7		Dport	√			
8		State	√	√		
9		sTos				
10		dTos				
11		TotPkts			√	
12		TotBytes			√	√
13		SrcBytes	√	√		
14		Label	√		√	
15		Train				
16		StartTime	√	√	√	√
17		ActivityLabel	√	√	√	√
18	Graph-Based Features	orig_degree	√			
19		resp_degree	√			
20		orig_closeness	√			
21		resp_closeness	√			
22		orig_betweenness	√			
23		resp_betweenness	√			

Table 4. Feature selection obtained by all comparison algorithms for IoT-23

Sr. No.	Features		Feature Selection Algorithms			
			CNI-VIF	VIF	PCA	RFE
1	Original Features	Ts				
2		uid				
3		id.orig_h	√	√	√	√
4		id.orig_p	√	√	√	√
5		id.resp_h	√	√	√	√
6		id.resp_p	√	√	√	√
7		proto	√	√	√	√
8		service	√			
9		duration	√	√	√	
10		orig_bytes	√	√		
11		resp_bytes	√	√		
12		conn_state				
13		local_orig				
14		local_resp				
15		missed_bytes				
16		history	√	√	√	√
17		orig_pkts			√	
18		orig_ip_bytes				
19		resp_pkts				
20		resp_ip_bytes				
21		ActivityLabel	√	√	√	√
22	Graph-Based Features	orig_degree	√	√	√	
23		resp_degree	√			
24		orig_closeness	√			
25		resp_closeness	√			
26		orig_betweenness				
27		resp_betweenness				

Table 5. Feature selection obtained by all comparison algorithms for NCC-2

Sr. No.	Features		Feature Selection Algorithms			
			CNI-VIF	VIF	PCA	RFE
1	Original Features	StartTime	√	√	√	
2		Dur	√	√	√	√
3		Proto				
4		SrcAddr	√			
5		Sport	√	√	√	√
6		Dir				
7		DstAddr	√	√	√	
8		Dport	√			
9		State	√	√		
10		sTos				
11		dTos				
12		TotPkts			√	
13		TotBytes			√	√
14		SrcBytes	√	√		
15		Label	√		√	
16		ActivityLabel				
17		BotnetName				
18		SensorId				
19	Graph-Based Features	orig_degree	√			√
20		resp_degree	√			
21		orig_closeness	√			√
22		resp_closeness	√			
23		orig_betweenness	√			
24		resp_betweenness	√			

The proposed algorithm is tested on these data frames and the outcomes are reported from the following parts. Tables 3, 4 and 5 give an idea about the feature selection for the CTU-13, IoT-23, and NCC-2 datasets, respectively. The selected features are marked '√' as shown below. Specifically, our focus is on whether or not graph features are selected after feature selection.

From Tables 3, 4 and 5, it can be observed that the CNI-VIF method significantly outperforms for selecting graph-based features for all the datasets used here and becomes the most eligible feature selection method to apply for graph databases.

Other feature selection methods fail to select most graph-based features and do not show their applicability to graph-based databases. Since CNI-VIF selects most of the added graph-based features, the number of features selected is always greater in comparison with VIF, PCA and RFE, which also increases the computation time of the algorithm.

To reduce the effect of this added dimensionality, CNI is used as an aggregate of BC, CC, and DC. Again, as we work on network traffic, CNI will be calculated for Source and Destination ips. We calculate an average of CNI of Sip and Dip to have a single composite feature.

Table 6 is simply an extension for Tables 3, 4 and 5, where all graph-based features are now replaced with CNI. Table 6 gives a clear idea about CNI feature selection using feature selection techniques for the datasets.

Table 6. $\overline{\text{CNI}}$ Selection obtained by all comparison algorithms for all the datasets used

Dataset	Feature	Feature selection algorithms			
		CNI-VIF	VIF	PCA	RFE
CTU-13	CNI	√			
IoT-23	CNI	√	√		
NCC-2	CNI	√			

Clearly, from Table 6, the CNI-VIF algorithm demonstrates a superior capability to select graph-based features compared to other feature selection methods like VIF, PCA, and RFE. Specifically, the CNI feature was consistently selected by the CNI-VIF algorithm across all datasets, highlighting its effectiveness in recognizing the importance of graph-specific attributes.

In contrast, the VIF method only identified the CNI feature in the IoT-23 dataset, and PCA and RFE failed to select CNI across all datasets. This indicates that traditional feature selection techniques may overlook essential graph-based features, potentially leading to a loss of critical information.

Table 7. Running time(s) in seconds obtained by CNI-VIF and VIF

Dataset	Feature Selection Algorithms	
	CNI-VIF	VIF
CTU-13	0.50	0.58
IoT-23	1.64	2.40
NCC-2	0.42	0.59

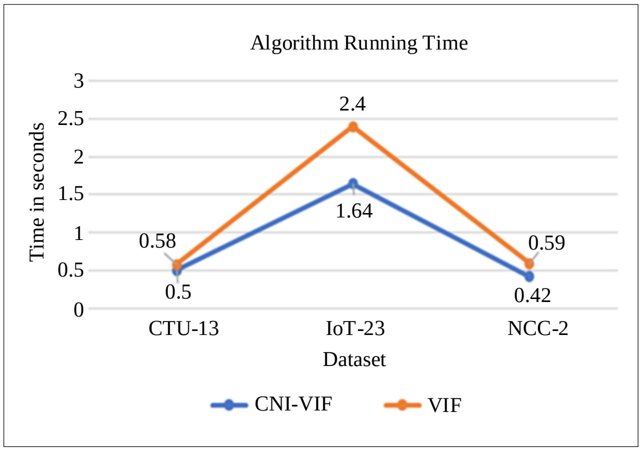

Fig. 1 Comparison of algorithm running time for CNI-VIF and VIF

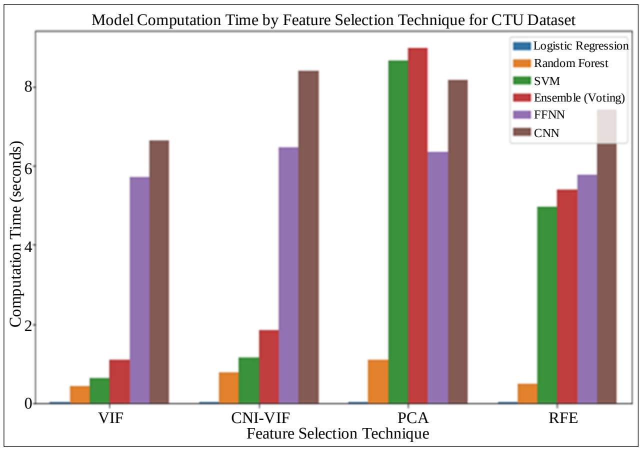

Fig. 2 Model computation time obtained by all comparison algorithms for CTU-13

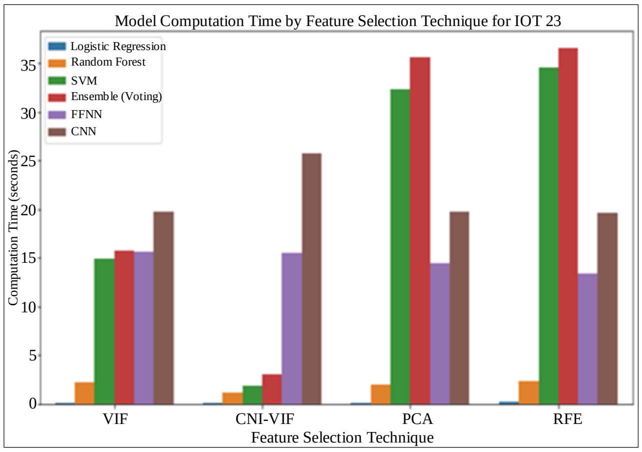

Fig. 3 Model computation time obtained by all comparison algorithms for IoT-23

The running times presented in Table 7 and Figure 1 demonstrate that the CNI-VIF algorithm is consistently more efficient than the traditional VIF method across all tested datasets. These results indicate that incorporating CNI enhances the selection of relevant graph-based features and contributes to a more efficient computational process.The reduced running times associated with CNI-VIF can be attributed to its ability to streamline the feature selection process by effectively capturing essential graph-based features, thus avoiding unnecessary computations.

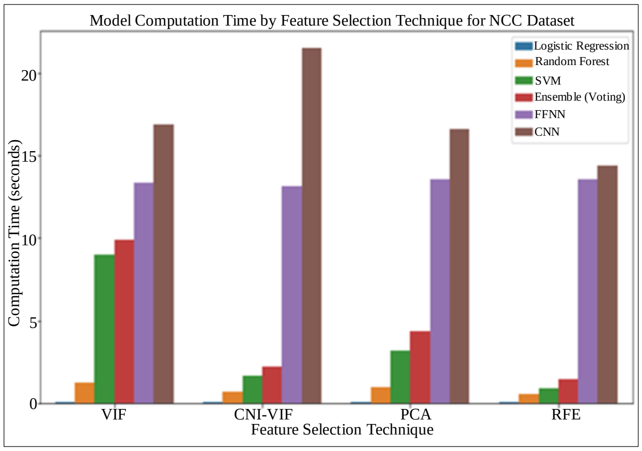

Fig. 4 Model computation time obtained by all comparison algorithms for NCC-2

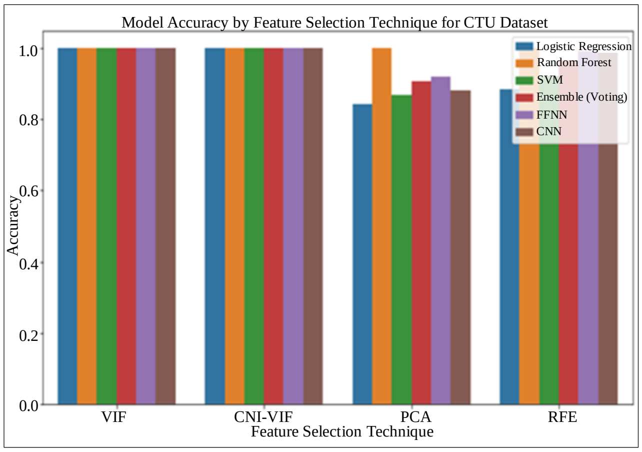

Fig. 5 Accuracy obtained by all comparison algorithms for CTU-13

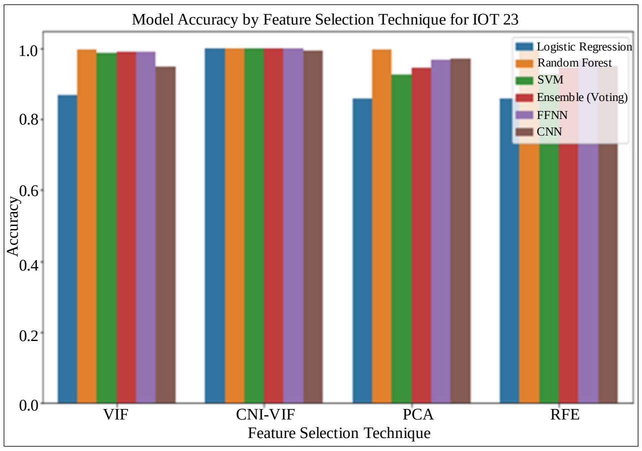

Fig. 6 Accuracy obtained by all comparison algorithms for IoT-23

The model computation time is also evaluated using the listed feature selection techniques, and the results can be seen in Figures 2, 3 and 4. The models include Logistic Regression, Random Forest, SVM, Ensemble (Voting), Convolutional Neural Network (CNN), and Feed-Forward Neural Network (FFNN). CNI-VIF clearly outperforms the rest of the three algorithms.

To evaluate the performance of the proposed method in contrast with the rest of the feature selection methods, the listed feature selection techniques are tested using popular machine learning models. The models such as Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Convolutional Neural Network (CNN), Feed-Forward Neural Network (FFNN), and Ensemble methods with soft voting are used here.

Table 8 describes accuracy, precision, recall and F1-score for all the evaluated models and the best performances are marked in bold. Figures 5, 6 and 7 denote these performances in graphical format. To maintain consistency, for values greater than 0.5, ceiling values are considered and for values less than 0.5, floor values are considered.

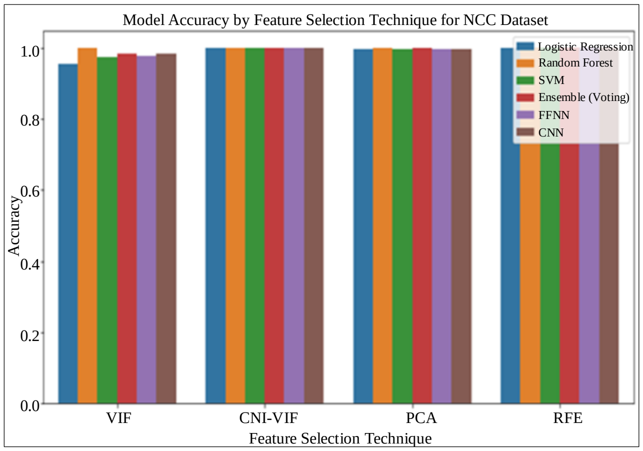

Fig. 7 Accuracy obtained by all comparison algorithms for NCC-2

Table 8 presents the accuracy (Ac), precision (Pr), recall (Re), and F1 score (F1) metrics for various machine learning models using different feature selection algorithms across three datasets. Across all datasets and models, the CNI-VIF approach consistently achieves perfect or near-perfect scores in all performance metrics. This indicates that CNI-VIF not only excels in selecting relevant features but also enhances model accuracy, precision, recall, and F1 scores, making it the most effective feature selection method among those tested. For instance, while VIF and PCA models show relatively high performance, they do not consistently achieve perfect scores, particularly in precision and recall.

Table 8 shows that the CNI-VIF method performs exceptionally well across a range of models, including LR, RF, SVM, CNN, FFNN, and Ensemble. Notably, models like RF and CNN often achieve perfect scores, further emphasizing the robustness of CNI-VIF in selecting features that enhance model performance. The results clearly demonstrate that the CNI-VIF method is superior to traditional feature selection methods for graph databases. CNI-VIF ensures the selection of relevant features and significantly enhances the performance metrics of various machine learning models. Clearly, the Random-Forest algorithm outperforms irrespective of the feature selection approach.

Table 8. Accuracy, precision, recall and F1 score obtained by all comparison algorithms

Dataset	Model	Feature Selection Algorithms															
		CNI-VIF				VIF				PCA				RFE			
		Performance Metrics															
		Ac	Pr	Re	F1	Ac	Pr	Re	F1	Ac	Pr	Re	F1	Ac	Pr	Re	F1
CTU-13	LR	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	87	83	77	80	89	87	78	82
	RF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	99	99	99	99	99	99	1	99
	SVM	99.8	1	99.5	99.8	**1**	**1**	**1**	**1**	85	88	65	75	92	98	78	87
	FFNN	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	86	88	67	78	95	97	88	92
	CNN	99.8	1	99.5	99.8	67	62	61	11	86	87	68	77	98	97	96	96
	Ensemble	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	90	91	78	84	96	99	87	93
IoT-23	LR	99	99	1	99	99	99	1	99	86	86	99	92	86	86	99	92
	RF	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	99	99	99	99	99	99	99	99
	SVM	99	1	99	99	99	1	99	99	96	98	97	97	97	98	98	98
	FFNN	99	99	1	99	99	99	1	99	99	99	99	99	99	99	99	99
	CNN	99	99	1	99	99	99	1	99	97	99	98	98	99	98	1	99
	Ensemble	99	99	1	99	99	99	1	99	98	98	1	99	99	98	1	99
NCC-2	LR	97.5	76	63	69	96	70	38	49	97	74	58	65	**1**	**1**	**1**	**1**
	RF	**1**	**1**	**1**	**1**	98	85	68	75	99.9	1	98	99	**1**	**1**	**1**	**1**
	SVM	98.8	91	80	84	97	78	48	59	98.8	92	80	86	99	1	98	99
	FFNN	99	88	87	87	95	65	58	61	99	92	85	89	**1**	**1**	**1**	**1**
	CNN	98.9	81	82	82	97	71	62	66	98.9	88	88	88	**1**	**1**	**1**	**1**
	Ensemble	98.8	91	82	86	97	78	55	64	98.8	94	81	88	**1**	**1**	**1**	**1**

5. Conclusion and Future Work

The results validate the efficacy of the CNI-VIF approach in feature selection for graph databases. By integrating Composite Node Information (CNI) with traditional VIF, the proposed method successfully identifies and retains crucial graph-based features, significantly improving the performance of various machine learning models.

Compared to traditional methods such as VIF, PCA, and RFE, CNI-VIF consistently selects more relevant features and validates superior performance regarding model accuracy and computational efficiency. This work introduces CNI-VIF, an enhanced feature selection method incorporating graph-based features such as centrality measures. By better capturing the importance and influence of nodes within the graph, CNI-VIF provides a more interpretable measure of multicollinearity that considers both traditional predictors and graph-specific features, offering deeper insights into the data. The Random Forest algorithm achieves the highest performance metrics with CNI-VIF-selected features, underscoring the robustness of our approach. This study underscores the importance of incorporating graph-specific characteristics in feature selection methodologies and positions CNI-VIF as a powerful tool for enhancing analysis and decision-making in graph databases.

CNI-VIF is a powerful feature selection method, particularly suited for graph-based datasets and applications, but it has some limitations. Calculating CNI values across large and highly interconnected graphs can be computationally intensive, especially for real-time applications. CNI-VIF relies heavily on the structure and accuracy of the graph database. If the graph structure has missing or noisy edges or nodes, it may misrepresent node importance, affecting feature selection and model performance. Since CNI-VIF leverages graph-specific centrality measures, its application and effectiveness may be reduced for non-graph or non-relational data.

Future research can explore further enhancements to the CNI-VIF methodology, such as integrating additional graph-based features or optimizing the selection of α for different datasets. Additionally, extending the CNI-VIF approach to other types of graph databases and applying it to real-time data analysis scenarios could provide further validation of its effectiveness and versatility. Investigating the scalability of CNI-VIF for larger graphs and diverse applications in cybersecurity, social network analysis, and bioinformatics presents promising avenues for future exploration.

References

[1] Bing Xue et al., “A Survey on Evolutionary Computation Approaches to Feature Selection,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606-626, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Pablo A. Estevez et al., “Normalized Mutual Information Feature Selection,” *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189-201, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[3] Weikuan Jia et al., “Feature Dimensionality Reduction: A Review,” *Complex & Intelligent Systems*, vol. 8, pp. 2663-2693, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xiaoping Li, Yadi Wang, and Rubén Ruiz, “A Survey on Sparse Learning Models for Feature Selection,” *IEEE Transactions on Cybernetics*, vol. 52, no. 3, pp. 1642-1660, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Guanglu Sun et al., “Feature Selection for IoT Based on Maximal Information Coefficient,” *Future Generation Computer Systems*, vol. 89, pp. 606-616, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Afnan Alharbi, and Khalid Alsubhi, “Botnet Detection Approach Using Graph-Based Machine Learning,” *IEEE Access*, vol. 9, pp. 99166-99180, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Giorgio Roffo et al., “Infinite Feature Selection: A Graph-Based Feature Filtering Approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4396-4410, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Santiago Timón-Reina, Mariano Rincón, and Rafael Martínez-Tomás, “An Overview of Graph Databases and their Applications in the Biomedical Domain,” *Database The Journal of Biological Databases and Curation*, vol. 2021, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Gonzalo Cerruela-García, José Manuel Cuevas-Muñoz, and Nicolás García-Pedrajas, “Graph-Based Feature Selection Approach for Molecular Activity Prediction,” *Journal of Chemical Information and Modeling*, vol. 62, no. 7, pp. 1618-1632, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ronghua Shang et al., “Sparse and Low-Redundant Subspace Learning-Based Dual-Graph Regularized Robust Feature Selection,” *Knowledge-Based Systems*, vol. 187, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Adnan Yazici, and Ezgi Taşkomaz, “BF-BigGraph: An Efficient Subgraph Isomorphism Approach Using Machine Learning for Big Graph Databases,” *Information Systems*, vol. 124, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] S. García et al., “An Empirical Comparison of Botnet Detection Methods,” *Computers & Security*, vol. 45, pp. 100-123, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[13] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga, *A Labeled Dataset with Malicious and Benign IoT Network Traffic*, Aposemat IoT-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] M. Aidiel Rachman Putra, Tohari Ahmad, and Dandy Pramana Hostiadi, *NCC-2 Dataset: Simultaneous Botnet Dataset*, Mendeley Data, Version 2, 2022. [CrossRef] [Publisher Link]

[15] Zhou Zhao et al., “Graph Regularized Feature Selection with Data Reconstruction,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 3, pp. 689-700, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Haishuai Wang et al., “Incremental Subgraph Feature Selection for Graph Classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 128-142, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ronghua Shang et al., “Self-Representation Based Dual-Graph Regularized Feature Selection Clustering,” *Neurocomputing*, vol. 171, pp. 1242-1253, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Sina Tabakhi, Parham Moradi, and Fardin Akhlaghian, “An Unsupervised Feature Selection Algorithm Based on Ant Colony Optimization,” *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 112-123, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[19] Hojat Ghimatgar et al., “An Improved Feature Selection Algorithm Based on Graph Clustering and Ant Colony Optimization,” *Knowledge-Based Systems*, vol. 159, pp. 270-285, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zhiwei Hu et al., “Feature Selection Based on Graph Structure,” *Combinatorial Optimization and Applications*, pp. 289-302, 2019. [CrossRef] [Publisher Link]

[21] Thosini Bamunu Mudiyanselage, and Yanqing Zhang, “Feature Selection with Graph Mining Technology,” *Big Data Mining and Analytics*, vol. 2, no. 2, pp. 73-82, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[22] Mi Wang et al., “Graph-Kernel Based Structured Feature Selection for Brain Disease Classification Using Functional Connectivity Networks,” *IEEE Access*, vol. 7, pp. 35001-35011, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[23] Fan Cheng et al., “Graph-Based Feature Selection in Classification: Structure and Node Dynamic Mechanisms,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 4, pp. 1314-1328, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Jiehong Cheng et al., “A Variable Selection Method Based on Mutual Information and Variance Inflation Factor,” *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 268, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ling Zheng et al., “Feature Grouping and Selection: A Graph-Based Approach,” *Information Sciences*, vol. 546, pp. 1256-1272, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Consolata Gakii, Paul O. Mireji, and Richard Rimiru, “Graph Based Feature Selection for Reduction of Dimensionality in Next-Generation RNA Sequencing Datasets,” *Algorithms*, vol. 15, no. 1, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Robert M. O’brien, “A Caution Regarding Rules of Thumb for Variance Inflation Factors,” *Quality & Quantity*, vol. 41, pp. 673-690, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[28] Anagha280883/CNI-VIF, 2024. [Online]. Available: https://github.com/Anagha280883/CNI-VIF/tree/main

[29] Dandy Pramana Hostiadi, and Tohari Ahmad, “Dataset for Botnet Group Activity with Adaptive Generator,” *Data in Brief*, vol. 38, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Ian T. Jolliffe, and Jorge Cadima, “Principal Component Analysis: A Review and Recent Developments,” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[31] Arif Mudi Priyatno, and Triyanna Widiyaningtyas, “A Systematic Literature Review: Recursive Feature Elimination Algorithms,” *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, no. 2, pp. 196-207, 2024. [CrossRef] [Google Scholar] [Publisher Link]