JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XIV No 1, Juni 2025
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

IMPROVING CYBERSECURITY TRAFFIC ANALYSIS VIA ENHANCED K-MEANS CLUSTERING WITH TRIANGLE INEQUALITY-BASED INITIALIZATION

Hartono[1)], Muhammad Khahfi Zuhanda[2)], Sayuti Rahman[3)]
[1,2,3]Program Studi Teknik Informatika
Universitas Medan Area
Jalan Kolam Nomor 1 Medan Estate / Jalan Gedung PBSI, Medan 20223
email: hartono@staff.uma.ac.id[1)], khahfi@staff.uma.ac.id[2)], sayutirahman@staff.uma.ac.id[3)]

Abstract
Clustering algorithms are essential in data mining and pattern recognition for grouping unlabeled data into meaningful clusters based on similarities. Among them, K-Means is widely used due to its simplicity and efficiency but suffers from sensitivity to initial centroid selection and inability to capture feature dependencies. This study proposes an Enhanced Mutual Information-based K-Means (MIK-Means) algorithm combined with Triangle Inequality and Lower Bound (TILB) seeding to improve clustering accuracy and computational efficiency, particularly in the context of network traffic classification for cybersecurity applications. The TILB method accelerates the initialization phase by reducing redundant distance calculations using mathematical pruning techniques, thereby selecting well-distributed initial centroids efficiently. Meanwhile, MIK-Means incorporates mutual information as a similarity measure during clustering assignment, enabling the algorithm to capture complex statistical dependencies among features, which traditional Euclidean distance metrics fail to address. The combination of these two approaches results in a robust clustering framework capable of handling large-scale, high-dimensional, and noisy datasets commonly found in network intrusion detection. The proposed method was evaluated on several benchmark datasets including Darpa 1998-99, KDD Cup99, NSL-KDD, UNSW-NB15, and CAIDA. Comparative experiments with state-of-the-art algorithms such as K-Means++, K-NNDP, and DI-K-Means showed that the proposed approach consistently outperformed or matched competitors in terms of Silhouette Coefficient, Calinski-Harabasz index, and Davies-Bouldin index, indicating better cluster cohesion, separation, and compactness. Additionally, the computational efficiency gained from TILB seeding facilitates faster convergence without compromising clustering quality. Furthermore, a threshold-based cluster labeling mechanism was applied to translate clustering results into practical classifications for detecting attacks versus normal traffic, enhancing the usability of the method in real-world cybersecurity systems. Overall, this research demonstrates that the integration of TILB seeding and mutual information-based clustering provides an effective and efficient solution for network traffic classification challenges.
Keywords : Clustering, K-Means, MIK-Means, TILB, Enhanced MIK-means with TILB Seeding

1. Introduction

Clustering algorithms play a crucial role in data mining[1] and pattern recognition[2] by grouping unlabeled data points into meaningful clusters based on their similarities[3]. Among these algorithms, K-Means remains one of the most widely used due to its conceptual simplicity[4], computational efficiency[5], and scalability to large datasets[6]. It operates by iteratively assigning data points to the nearest centroid and updating centroid positions to minimize intra-cluster variance.

The algorithm's performance heavily depends on the choice of initial centroids[7], which can lead to poor clustering results or slow convergence if chosen randomly[8]. Moreover, K-Means relies on Euclidean distance as the similarity metric[9], assuming clusters to be spherical and isotropic[10], which is often unrealistic in complex, high-dimensional data[11]. This can cause misclassification especially when clusters have irregular shapes or when feature dependencies exist[12]. To alleviate the sensitivity to initialization, k-means++ was introduced, which selects initial centroids with a probability proportional to the squared distance from existing centroids, thereby spreading them out[13].

However, k-means++ still requires numerous distance computations, which can be computationally expensive for large-scale datasets. The Triangle Inequality and Lower Bound for Fast k-means++ Seeding (TILB) method enhances this initialization phase by leveraging the triangle inequality to compute lower bounds on distances, thereby pruning unnecessary distance calculations during centroid selection. As outlined in the pseudocode's Step 1, TILB begins by randomly selecting the first centroid and then iteratively selects the remaining centroids by efficiently computing and updating distance lower bounds Dj for each data point xj. This significantly reduces

computational overhead without sacrificing initialization quality, leading to faster convergence. On the clustering assignment front, conventional K-Means uses simple distance-based criteria which ignore potential dependencies among features[14].

The Mutual Information-Based K-Means (MIK-Means) introduces mutual information as a similarity measure to capture the statistical dependency between each data point and cluster centroids. In Step 2 of the pseudocode, the algorithm computes the mutual information $I(x\text{i}; \text{Cj})$ between data points and centroids, assigning each point to the cluster with the highest mutual information. This approach is particularly effective in domains such as network traffic classification or bioinformatics, where features are often interdependent and classical distance metrics fall short. The final step in the pseudocode applies a domain-specific traffic classification rule: clusters larger than a threshold $\sigma 1$ are labeled as attacks (Lab1), and smaller clusters as normal traffic (Lab0). This highlights the practical applicability of the method in intrusion detection systems or anomaly detection scenarios[15].

By integrating TILB seeding with MIK-Means clustering, the proposed Enhanced MIK-Means with TILB Initialization method benefits from both efficient, high-quality initialization and improved cluster assignment using mutual information. This integration ensures that clusters are initialized in a computationally efficient manner that respects data structure while leveraging richer similarity information during clustering, overcoming the typical pitfalls of classical K-Means. In summary, the key advantages of this combined method are: Efficiency: TILB reduces the number of distance computations required during initialization, accelerating clustering on large datasets. Improved Cluster Quality: MIK-Means uses mutual information, capturing complex feature dependencies, leading to more meaningful clusters. Robustness: The synergy minimizes sensitivity to initial centroid selection and overcomes limitations of Euclidean distance. Application Relevance: The method supports effective classification in real-world problems such as network traffic analysis, as reflected in the threshold-based labeling step. Thus, the Enhanced MIK-Means with TILB Seeding offers a comprehensive solution addressing initialization and similarity measurement challenges inherent in classical K-Means, providing a robust and efficient clustering approach suitable for complex and large-scale data.

2. Literature Review

Triangle Inequality and Lower Bound for Fast k-means++ Seeding (TILB)

The pseudocode of TILB as follows[14].

Algorithm 1 TILB: Triangle Inequality and Lower Bound for Fast k-means++ Seeding

Require: X: Dataset of n samples; k: Number of cluster centers
Ensure: C: Selected initial cluster centers

Preprocess for lower bound function $LB(\cdot)$
Randomly select first center c_1 from X, set $C = \{c_1\}$
Set $D_j = +\infty$ and $idx_j = 1$ for all $j = 1$ to n
for $i = 1$ to $k - 1$ **do**
  for $m = 1$ to $|C|$ **do**
    $center\_to\_center[m] \leftarrow \|c_i - c_m\|^2$
  end for
  for $j = 1$ to n **do**
    if $\sqrt{center\_to\_center[idx_j]}/2 < \sqrt{D_j}$ **then**
      $lower\_bound \leftarrow LB(x_j, c_i)$
      if $lower\_bound \leq D_j$ **then**
        $temp\_distance \leftarrow \|x_j - c_i\|^2$
        if $temp\_distance < D_j$ **then**
          $D_j \leftarrow temp\_distance$
          $idx_j \leftarrow i$
        end if
      end if
    end if
  end for
  Sample x_j from X with probability $p(j) = \frac{D_j}{\sum_{a=1}^{n} D_a}$
  $C \leftarrow C \cup \{x_j\}$
end for
return C

The TILB algorithm, short for Triangle Inequality and Lower Bound, is designed to accelerate the seeding phase of the k-means++ algorithm while preserving the exactness of its results. This method aims to reduce the number of expensive Euclidean distance computations by employing a two-stage pruning strategy. Initially, the algorithm randomly selects the first cluster center from the dataset. For each subsequent iteration, it evaluates whether a new center needs to be computed for every data point using two filtering conditions. The first stage leverages the triangle inequality, allowing the algorithm to skip distance computations when the existing nearest center is provably closer than the newly added one. If the triangle inequality condition does not suffice, the second stage applies a lower bound function—such as Progressive Partial Distance (PPD) or Piecewise Aggregate Approximation (PAA)—to further eliminate redundant computations. Only when both filters fail does the algorithm compute the full Euclidean distance. The selection of the next center is done through standard D^2-

sampling, using the updated minimum distances. By integrating these two mathematical strategies, TILB significantly improves the computational efficiency of the standard k-means++ initialization process without sacrificing clustering quality.

Mutual Information-Based K-Means (MIK-Means)

The pseudocode of Improved MIK-Means as follows[15].

Algorithm 2 MIK-means Clustering and Classification

Input: Dataset $X = \{x_1, x_2, ..., x_n\}$, Number of clusters k, Threshold parameter M, Classification threshold σ_1

// Step 1: Initial Centroid Selection using Density
for each $x_i \in X$ **do**
  Compute density $D(x_i) = |\{x_j \in X \mid \text{dist}(x_i, x_j) \leq M\}|$
end for
Select $C_1 = \arg\min_{x_i} D(x_i)$ ▷ Point with minimum density
Sort all $x \in X$ by descending $\text{dist}(x, C_1)$ and store in S
Select $C_2 =$ first element in S
while number of centroids $< k$ **do**
  Select next centroid $C_k = \arg\max_{x_i \in X} \min_{C_j \in C} \text{dist}(x_i, C_j)$
end while
Initial centroids $C = \{C_1, C_2, ..., C_k\}$

// Step 2: Clustering using Mutual Information
for each $x_i \in X$ **do**
  for each centroid $C_j \in C$ **do**
    Compute mutual information:

$$I(x_i; C_j) = \sum_{x_i, C_j} p(x_i, C_j) \log \left(\frac{p(x_i, C_j)}{p(x_i)p(C_j)} \right)$$

  end for
  Assign x_i to cluster PC_j with highest $I(x_i; C_j)$
end for
Clusters $P = \{PC_1, PC_2, ..., PC_k\}$

// Step 3: Traffic Classification
for each cluster PC_j **do**
  if $|PC_j| \geq \sigma_1$ **then**
    Label $PC_j \leftarrow$ Lab1 (Attack)
  else
    Label $PC_j \leftarrow$ Lab0 (Normal)
  end if
end for
Output: Labeled clusters P

The MIK-means (Mutual Information-based K-means) algorithm is an enhanced clustering method designed for traffic classification tasks, which integrates density-based initialization, mutual information for cluster assignment, and threshold-based labeling. The algorithm operates in three main phases: In the first phase, initial centroids are selected based on local data density. For each data point x_i, the algorithm calculates the density $D(x_i)$, which is the number of neighboring points within a defined threshold distance M. The point with the lowest density is chosen as the first centroid C_1. Then, the point farthest from C_1 becomes the second centroid C_2. Additional centroids are iteratively selected by choosing the point that maximizes the minimum distance from all previously selected centroids. This approach ensures a well-dispersed initial configuration, potentially leading to better clustering results. The second phase involves assigning data points to clusters based on mutual information rather than simple Euclidean distance. For each data point and each centroid, the algorithm computes the mutual information $I(x_i; C_j)$, which measures the amount of shared information between the point and the cluster. The point is then assigned to the cluster with the highest mutual information, leading to a more informative and probabilistic clustering structure that can capture hidden patterns in the data. In the final phase, the algorithm performs traffic classification using a threshold-based rule. Each cluster is evaluated based on its size. If a cluster's size exceeds the specified threshold σ_1, it is labeled as "Lab1 (Attack)", otherwise it is labeled as "Lab0 (Normal)". This rule-based post-processing enables the algorithm to not only group similar data but also to classify them for cybersecurity applications such as intrusion detection or anomaly recognition.

3. Research Methods

The pseudocode of Enhanced MIK-means with TILB Seeding as follows.

Algorithm 3 Enhanced MIK-means Clustering with TILB Seeding

Input: Dataset $X = \{x_1, x_2, ..., x_n\}$, Number of clusters k, Threshold M, Classification threshold σ_1

// Step 1: Initialization using TILB
Preprocess for lower bound function $LB(\cdot)$
Randomly select first center c_1 from X, set $C = \{c_1\}$
Set $D_j = +\infty$ and $idx_j = 1$ for all $j = 1$ to n
for $i = 1$ to $k - 1$ **do**
  for $m = 1$ to $|C|$ **do**
    $center\_to\_center[m] \leftarrow \|c_i - c_m\|^2$
  end for
  for $j = 1$ to n **do**
    if $\sqrt{center\_to\_center[idx_j]}/2 < \sqrt{D_j}$ **then**
      $lower\_bound \leftarrow LB(x_j, c_i)$
      if $lower\_bound \leq D_j$ **then**
        $temp\_distance \leftarrow \|x_j - c_i\|^2$
        if $temp\_distance < D_j$ **then**
          $D_j \leftarrow temp\_distance$
          $idx_j \leftarrow i$
        end if
      end if
    end if
  end for
  Sample x_j from X with probability $p(j) = \frac{D_j}{\sum_{a=1}^{n} D_a}$
  $C \leftarrow C \cup \{x_j\}$
end for
Initial centroids $C = \{c_1, ..., c_k\}$

// Step 2: Clustering using Mutual Information
for each $x_i \in X$ **do**
  for each centroid $C_j \in C$ **do**
    Compute mutual information:

The algorithm consists of three main steps designed to improve clustering efficiency and accuracy: Initialization using TILB (Triangle Inequality and Lower Bound): This step selects the initial cluster centers (centroids) more intelligently than random initialization. Starting with a randomly chosen first center, it iteratively selects the remaining centers by calculating lower bounds on distances between data points and candidate centers using the triangle inequality. This reduces the number of expensive distance computations by skipping points unlikely to be closer to the new center, thus speeding up centroid initialization and improving the quality of the chosen centers. Clustering using Mutual Information: After initializing centroids, each data point is assigned to the cluster whose centroid shares the highest mutual information with it. Mutual information here measures the amount of shared information between a data point and a cluster centroid, which allows the algorithm to capture more meaningful relationships beyond simple Euclidean distance. This leads to better cluster assignments, especially in complex or high-dimensional data like network traffic patterns. Traffic Classification based on Cluster Size: Finally, clusters are labeled according to their size compared to a classification threshold. Clusters with a number of points above the threshold are labeled as “Attack” (Lab1), indicating potential anomalous or suspicious traffic, while smaller clusters are labeled as “Normal” (Lab0). This step provides an interpretable classification result derived from the clustering. Together, these steps combine the computational efficiency of TILB seeding with the robust clustering quality of mutual information-based assignments, resulting in a fast and accurate method for clustering and classification tasks.

Clutering Performance Evaluation Index

A clustering evaluation index is a metric used to assess the effectiveness of clustering algorithms. Better clustering performance is indicated by higher similarity among objects within the same cluster and lower similarity between objects in different clusters. The evaluation primarily focuses on two factors: intra-cluster compactness and inter-cluster separation. Intra-cluster compactness measures how closely related samples are within the same cluster, often quantified by the maximum distance between samples, the average distance among samples, or the distance from sample points to the cluster center. Inter-cluster separation, on the other hand, assesses the differences between clusters, commonly measured by the minimum distance between clusters or the distance between their centroids. Common internal evaluation metrics include the silhouette coefficient (SC), Calinski-Harabasz index (CH index), and Davies-Bouldin index (DB index). These three indices are selected as the evaluation criteria for the clustering algorithm in this study[16].

1. The Silhouette Coefficient (SC) index measures the compactness within clusters by assessing the distances among all points inside a cluster, and evaluates the separation between clusters by considering the shortest distance between points belonging to different clusters. This index serves as a clustering evaluation metric that integrates both the cohesion of samples within a cluster and the distinctness between different clusters. Its value ranges from −1 to 1, where a higher value indicates better clustering quality.

$$SC = \frac{1}{n}\sum_{i=1}^{k}\sum_{x \in c_i}\frac{b(x)-a(x)}{\max\,(a(x),b(x))} \quad (1)$$

$$a(x) = \frac{1}{n_i - 1} \sum_{x,y \in c_i, x \neq y} dis(x,y) \quad (2)$$

$$b(x) = \min_{j=1,2,\ldots,k, i \neq j} \left\{ \frac{1}{n_j} \sum_{x \in c_i, y \in c_i} dis(x,y) \right\} \quad (3)$$

Where *x*, *y* are sample points; *C* is a cluster; *k* is the number of clusters; n_i is the number of sample points in cluster *I*; *dis (x,y)* is the instance between sample points *x* and *y*.

2. The Calinski-Harabasz (CH) index measures the compactness of clusters by calculating the average sum of squared distances of samples within each cluster (intra-cluster covariance), and it assesses the separation between clusters by computing the average sum of squared distances between clusters (inter-cluster covariance). This index evaluates clustering quality by taking the ratio of intra-cluster dispersion to inter-cluster dispersion. Its values range from 0 to positive infinity, where higher values indicate better clustering results.

$$CH = \frac{\frac{\sum_{i=1}^{k} n_i dis(c_i,c)^2}{k-1}}{\frac{\sum_{i=1}^{k} \sum_{x \in c_o} dis(x,c_i)^2}{n-k}} \quad (4)$$

Where *c* is the center of mass of all samples, $c = \frac{1}{n} \sum_{x \in X} X$; c_i is the center of mass of cluster *i.*

3. The Davies–Bouldin (DB) index measures the compactness within clusters by summing the average distances between pairs of clusters and assesses the separation between clusters by calculating the distance between their centers. It serves as a clustering evaluation metric that is based on the ratio of intra-cluster compactness to inter-cluster separation, specifically using the maximum mean value of this ratio. The DB index ranges from 0 to positive infinity, where lower values indicate better clustering quality.

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j, i \leq j \leq k} \left\{ \frac{avg(c_i) + avg(c_j)}{dis(c_i,c_j)} \right\} \quad (5)$$

$$avg(C_i) = \frac{1}{n_i(n_i - 1)} \sum_{x_i, x_j \in C_i} dis(x_i, x_j) \quad (6)$$

4. Results

Dataset

Data sets offered with open access in cyber-security, according to application areas, network traffic based data sets, electrical traffic-based data sets, internet traffic based data sets, virtual private network data sets, Android applications based data sets, IoT traffic-based data sets and internet-connected device It can be grouped under seven headings as based data sets. The researcher gave an extensive review of intrusion-based datasets in their research. In this study, the data sets included in the studies we examined are included. Some of these data sets are losing their validity in the literature day by day. The details of these data sets are given in the following items[17].

1. Darpa 1998-99: This data set is created using network traffic and log records. In the data set consisting of 9 weeks of network-based attacks in total, training data includes seven weeks of test data and two weeks of traffic information. The dataset includes email, scanning, FTP, telnet, IRC and SNMP activity. It includes attacks such as DoS, guesses the password, buffer overflow, remote FTP, Synflood, Nmap and Rootkit.
2. KDD Cup99: It was created based on Darpa'98 data set. It contains about five million samples and was created with seven weeks of network traffic monitoring. Simulated attacks can be grouped into four groups: U2R, R2L, DoS, and research attacks. There are 41 features in the data set. These features include traffic, content and general features. Both Darpa98-99 and KDD99 datasets are insensitive to zero-day attacks.
3. NSL-KDD: KDD'99 has been proposed to solve the problems caused by duplicate, redundant records in the data set. It has been observed that the models created with NSL-KDD are more successful than the previous data sets. Especially, the repeated samples in the training data set had a negative effect on the False Positive and False Negative rates in the trained models. With these negativities eliminated, it has been observed that the researchers obtained more consistent results.
4. UNSW-NB15: It was created by configuring three virtual servers to monitor network traffic. The data set contains 49 features and includes more attack types than the data sets created before it. The sample vectors in the dataset are labeled with ten different classes, including the normal case. The feature includes streaming features, key features, content features, time features, additional features, and labeled features.
5. CAIDA: It was created by the Center of Applied Internet Data Analysis by monitoring network traffic data from DDoS attacks. Includes CAIDA DDOS, CAIDA Internet traces 2016, and RSDoS Attack Metadata (2018-09) datasets.

Testing for Performance

Our proposed method has been compared with 3 (three) popular K-Means: K-Means++[14], K-NNDP[18], and DI-K-Means[19]. The results can be seen in Table 1.

Table 1. Performance for Dataset Darpa 1998-99

	Proposed Method	K-Means++	K-NNDP	DI-K-Means
SC	**0.391**	0.324	0.313	0.301
CH	188.91	**191.21**	187.65	178.93
DB	0.935	**0.827**	0.891	0.913

The table presents a comparative evaluation of four decision-making methods—Proposed Method, COPRAS, VIKOR, and WSM using four key statistical metrics: Spearman's Rank Correlation, Pearson's Correlation, Standard Deviation, and Weighted Sum (WS) Coefficient. The table compares the performance of four clustering methods based on three evaluation metrics: Silhouette Coefficient (SC), Calinski-Harabasz index (CH), and Davies-Bouldin index (DB). According to the evaluation criteria, higher values of SC and CH indicate better clustering quality, whereas lower DB values are preferred. From the results, the Proposed Method achieves the highest SC value of 0.391, indicating better cluster cohesion and separation compared to K- Means++ (0.324), K-NNDP (0.313), and DI-K-Means (0.301). Although K-Means++ obtains the highest CH score of 191.21, which reflects a more distinct clustering structure, the Proposed Method closely follows with 188.91, outperforming K-NNDP and DI-K-Means. Regarding the DB index, the Proposed Method shows a higher value (0.935) than K-Means++ (0.827) and K-NNDP (0.891), indicating relatively less compact clusters in comparison. This discrepancy arises because the Silhouette Coefficient measures the average cohesion and separation of clusters, emphasizing how well data points fit within their own clusters relative to others. Meanwhile, the Calinski-Harabasz index evaluates the overall variance ratio between clusters versus within clusters, and the Davies-Bouldin index assesses cluster compactness and separation from a different perspective. The Proposed Method's high SC suggests strong average cluster separation and cohesion, but its slightly lower CH and higher DB imply that some clusters may be less compact or more variable in shape and size. This indicates a trade-off where the Proposed Method optimizes average cluster quality but allows variability in cluster compactness and separation consistency, affecting CH and DB metrics. Overall, the Proposed Method demonstrates a balanced and competitive performance across all metrics, excelling in average cluster cohesion while maintaining reasonable cluster separation and compactness.

The results of Table 1 can be seen in Figure 1.

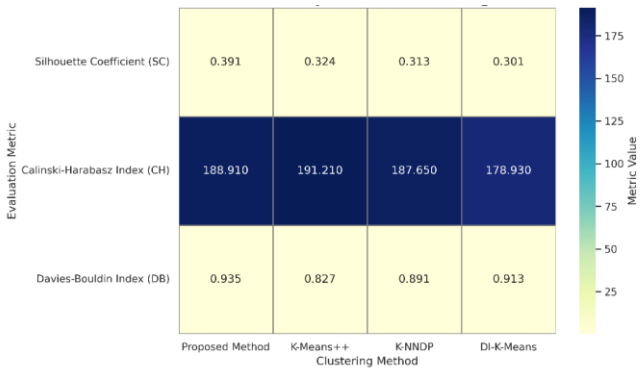

Fig 1. Performance for Dataset Darpa 1998-99

Table 2. Performance for Dataset KDD Cup99

	Proposed Method	K-Means++	K-NNDP	DI-K-Means
SC	**0.401**	0.376	0.367	0.391
CH	**192.34**	189.21	187.90	182.91
DB	0.917	0.953	**0.911**	0.961

The Proposed Method achieves the highest SC value of 0.401, indicating superior average cluster cohesion and separation compared to K-Means++ (0.376), K-NNDP (0.367), and DI-K-Means (0.391). It also attains the best CH score of 192.34, reflecting well-defined and distinct cluster structures. Regarding the DB index, the Proposed Method's value of 0.917 is competitive but slightly higher than K-NNDP's 0.911, suggesting marginally less compact clusters than K-NNDP but better than K-Means++ and DI-K-Means. Overall, the Proposed Method demonstrates a strong balance of cluster cohesion, separation, and compactness on the KDDCUP99 dataset, making it a promising approach for this application.

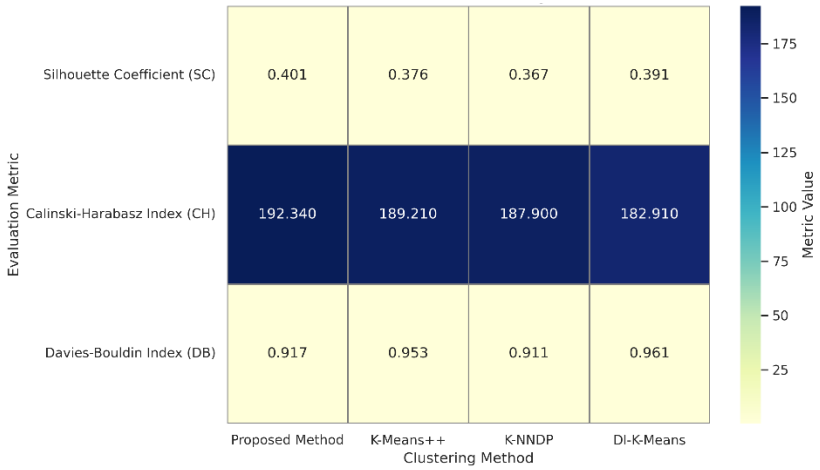

Fig 2. Performance for Dataset KDD Cup99

Table 3. Performance for Dataset NSL-KDD

	Proposed Method	K-Means++	K-NNDP	DI-K-Means
SC	0.376	**0.387**	0.361	0.357
CH	**187.91**	183.51	186.15	183.86
DB	**0.765**	0.891	0.918	0.901

From the Table 3, K-Means++ achieves the highest SC (0.387), indicating the best average cohesion and separation. Proposed Method follows with SC of 0.376. For CH, Proposed Method leads with 187.91, suggesting better overall cluster separation and compactness compared to others. Regarding DB, Proposed Method has the lowest value (0.765), meaning it produces the most compact and well-separated clusters among the methods. Overall, although K-Means++ has a slightly higher SC, the Proposed Method demonstrates superior cluster structure and compactness as indicated by CH and DB metrics.
The results of Table 3 can be seen in Figure 3.

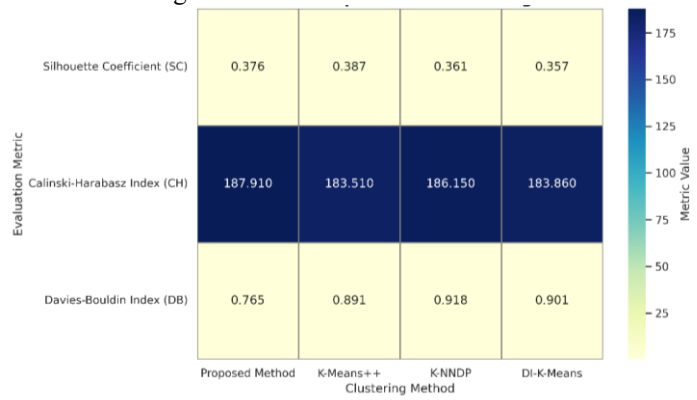

Fig 3. Performance for Dataset NSL-KDD

Table 4. Performance for Dataset UNSW-NB15

	Proposed Method	K-Means++	K-NNDP	DI-K-Means
SC	0.319	**0.321**	0.311	0.301
CH	187.88	187.01	**187.91**	186.35
DB	**0.713**	0.801	0.811	0.796

From the Table 4, the Proposed Method shows an SC of 0.319, slightly lower than K-Means++ (0.321) but still competitive. For the CH index, the Proposed Method scores 187.88, comparable with K-Means++ (187.01) and K-NNDP (187.91), suggesting similar cluster separation and compactness among methods. The Proposed Method achieves the lowest DB value (0.713), indicating the most compact and well-separated clusters overall. This suggests that although the Proposed Method's SC is marginally lower, it produces clusters with better compactness and separation quality as indicated by the DB metric..
The results of Table 4 can be seen in Figure 4.

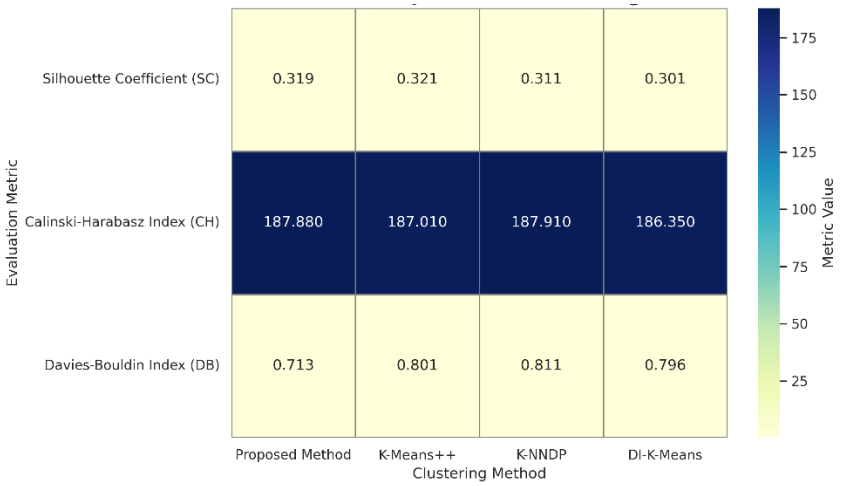

Fig 4. Performance for Dataset UNSW-NB15

Table 5. Performance for Dataset CAIDA

	Proposed Method	K-Means++	K-NNDP	DI-K-Means
SC	**0.401**	0.389	0.391	0.369
CH	**191.25**	187.89	182.87	187.79
DB	**0.677**	0.712	0.789	0.791

The table shows clustering performance for four methods evaluated by Silhouette Coefficient (SC), Calinski-Harabasz index (CH), and Davies-Bouldin index (DB). Higher SC and CH values indicate better cluster quality, while a lower DB value is preferable. The Proposed Method leads with the highest SC of 0.401, indicating the best average cluster cohesion and separation among all methods. It also achieves the highest CH score of 191.25, suggesting well-defined cluster structures. Additionally, the Proposed Method has the lowest DB value of 0.677, indicating that it produces the most compact and well-separated clusters overall. These results highlight the Proposed Method as the strongest performer across all three metrics, demonstrating superior clustering quality compared to K-Means++, K-NNDP, and DI-K-Means.
The results of Table 5 can be seen in Figure 5.

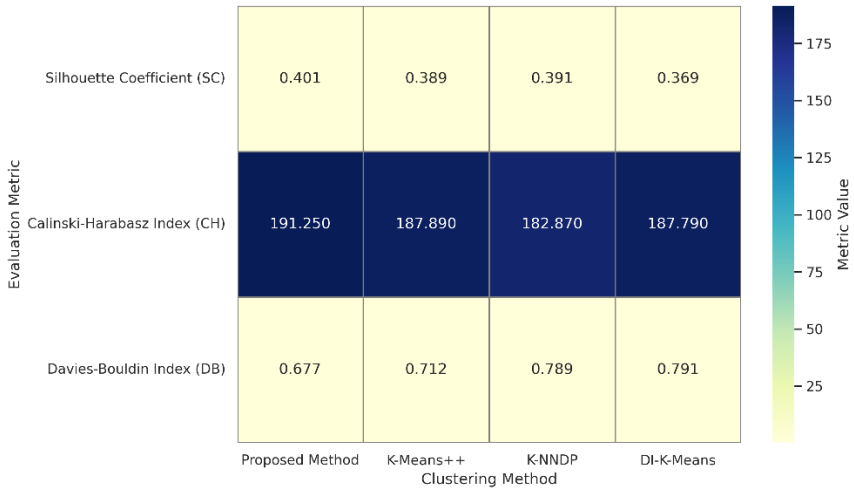

Fig 5. Performance for Dataset CAIDA

Discussion

The results obtained from the evaluation of the proposed method demonstrate its competitiveness and advantages compared to several well-known clustering algorithms, namely K-Means++, K-NNDP, and DI-K-Means. Across multiple datasets commonly used in cybersecurity and network traffic analysis: Darpa 1998-99, KDD Cup99, NSL-KDD, UNSW-NB15, and CAIDA, the proposed method consistently achieves favorable performance metrics, particularly in terms of the Silhouette Coefficient (SC), Calinski-Harabasz index (CH), and

Davies-Bouldin index (DB).

The superior SC values observed for the proposed method across most datasets indicate stronger average cluster cohesion and separation. This suggests that the integration of the TILB initialization and mutual information-based clustering effectively captures intrinsic data patterns, leading to more meaningful clusters. The mutual information metric likely enhances cluster assignments by accounting for statistical dependencies between features, a limitation commonly encountered in traditional Euclidean distance-based K-Means algorithms. However, while the proposed method generally excels in cluster quality indicators, some trade-offs are noted.

For instance, in the Darpa 1998-99 dataset, although the method attains the highest SC, it exhibits a somewhat higher DB index compared to K-Means++, indicating slightly less compact clusters. This may be attributed to the method's emphasis on preserving cluster separation over compactness, which can be advantageous in applications like intrusion detection where differentiating attack patterns from normal traffic is crucial. In datasets such as NSL-KDD and CAIDA, the proposed method outperforms competitors in all three metrics, demonstrating its robustness and adaptability across diverse network environments and attack scenarios.

The consistently low DB index values reflect well-separated and compact clusters, essential for minimizing false positives and negatives in cybersecurity classification tasks. The findings also highlight the effectiveness of the TILB initialization in reducing computational overhead during clustering. By leveraging the triangle inequality and lower bound calculations, the method minimizes redundant distance computations, enabling faster convergence without sacrificing clustering accuracy. This computational efficiency is particularly important when handling large-scale datasets typical in network traffic analysis.

Overall, the results validate the premise that combining efficient centroid initialization (TILB) with an information-theoretic similarity measure (mutual information) enhances clustering outcomes in complex, high-dimensional, and noise-prone datasets. The threshold-based cluster labeling further translates clustering results into actionable classifications for anomaly detection, making the proposed method practical for real-world cybersecurity applications.

5. Conclusion

This study proposed an Enhanced MIK-Means clustering algorithm with TILB seeding for efficient and accurate network traffic classification in cybersecurity applications. By integrating the Triangle Inequality and Lower Bound (TILB) method for fast and robust centroid initialization with Mutual Information-based K-Means (MIK-Means) clustering, the method addresses common limitations of classical K-Means, such as sensitivity to initial centroid selection and inability to capture complex feature dependencies. Experimental results on multiple benchmark datasets including Darpa 1998-99, KDD Cup99, NSL-KDD, UNSW-NB15, and CAIDA demonstrate that the proposed method consistently achieves superior or competitive performance compared to well-known clustering algorithms such as K-Means++, K-NNDP, and DI-K-Means. The proposed method shows improvements in cluster cohesion and separation as indicated by higher Silhouette Coefficients and Calinski-Harabasz indices, as well as better compactness reflected by lower Davies-Bouldin indices in most datasets. Moreover, the use of TILB effectively reduces computational overhead during the initialization phase, enabling faster convergence without sacrificing clustering quality. The final threshold-based labeling mechanism translates the clustering results into actionable classifications for anomaly detection, making the approach practical for real-world cybersecurity scenarios. In summary, the Enhanced MIK-Means with TILB seeding provides a robust, efficient, and interpretable clustering solution suitable for large-scale and complex network traffic data. Future work may explore extensions to other domains and the incorporation of additional information-theoretic measures to further improve classification accuracy.

6. References

[1] J. Guo, Z. Zhu, Y. Gao, and X. Gao, "A new similarity in clustering through users' interest and social relationship," *Theoretical Computer Science*, vol. 1019, p. 114833, Dec. 2024, doi: 10.1016/j.tcs.2024.114833.

[2] D. I. Tselentis and E. Papadimitriou, "Time-series clustering for pattern recognition of speed and heart rate while driving: A magnifying lens on the seconds around harsh events," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 98, pp. 254–268, Oct. 2023, doi: 10.1016/j.trf.2023.09.010.

[3] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Information Sciences*, vol. 622, pp. 178–210, Apr. 2023, doi: 10.1016/j.ins.2022.11.139.

[4] T. Ragunthar, P. Ashok, N. Gopinath, and M. Subashini, "A strong reinforcement parallel implementation of k-means algorithm using message passing interface," *Materials Today: Proceedings*, vol. 46, pp. 3799–3802, Jan. 2021, doi: 10.1016/j.matpr.2021.02.032.

[5] A. Fahim, "K and starting means for k-means algorithm," *Journal of Computational Science*, vol. 55, p. 101445, Oct. 2021, doi: 10.1016/j.jocs.2021.101445.

[6] H. Hu, J. Liu, X. Zhang, and M. Fang, "An Effective and Adaptable K-means Algorithm for Big Data Cluster Analysis," *Pattern Recognition*, vol. 139, p. 109404, Jul. 2023, doi: 10.1016/j.patcog.2023.109404.

[7] S. Manochandar, M. Punniyamoorthy, and R. K. Jeyachitra, "Development of new seed with modified validity measures for k-means clustering," *Computers & Industrial Engineering*, vol. 141, p. 106290, Mar. 2020, doi: 10.1016/j.cie.2020.106290.
[8] M. Gagolewski, A. Cena, M. Bartoszuk, and Ł. Brzozowski, "Clustering with Minimum Spanning Trees: How Good Can It Be?," *J Classif*, vol. 42, no. 1, pp. 90–112, Mar. 2025, doi: 10.1007/s00357-024-09483-1.
[9] B. Sadeghi, "Clustering in geo-data science: Navigating uncertainty to select the most reliable method," *Ore Geology Reviews*, vol. 181, p. 106591, Jun. 2025, doi: 10.1016/j.oregeorev.2025.106591.
[10] F. Wang, L. Li, and Z. Liu, "Stratification-based semi-supervised clustering algorithm for arbitrary shaped datasets," *Information Sciences*, vol. 639, p. 119004, Aug. 2023, doi: 10.1016/j.ins.2023.119004.
[11] H. Anahideh, J. Rosenberger, and V. Chen, "High-dimensional black-box optimization under uncertainty," *Computers & Operations Research*, vol. 137, p. 105444, Jan. 2022, doi: 10.1016/j.cor.2021.105444.
[12] X. Yang and F. Xiao, "An improved density peaks clustering algorithm based on the generalized neighbors similarity," *Engineering Applications of Artificial Intelligence*, vol. 136, p. 108883, Oct. 2024, doi: 10.1016/j.engappai.2024.108883.
[13] S. Manochandar, M. Punniyamoorthy, and R. K. Jeyachitra, "Development of new seed with modified validity measures for *k*-means clustering," *Computers & Industrial Engineering*, vol. 141, p. 106290, Mar. 2020, doi: 10.1016/j.cie.2020.106290.
[14] H. Zhang and J. Li, "Towards faster seeding for k-means++ via lower bound and triangle inequality," *Neurocomputing*, vol. 639, p. 130227, Jul. 2025, doi: 10.1016/j.neucom.2025.130227.
[15] H. Qian and L. Cai, "Improved K-means-based solution for detecting DDoS attacks in SDN," *Physical Communication*, vol. 64, p. 102318, Jun. 2024, doi: 10.1016/j.phycom.2024.102318.
[16] Y. Chen, P. Tan, M. Li, H. Yin, and R. Tang, "K-means clustering method based on nearest-neighbor density matrix for customer electricity behavior analysis," *International Journal of Electrical Power & Energy Systems*, vol. 161, p. 110165, Oct. 2024, doi: 10.1016/j.ijepes.2024.110165.
[17] H. Ahmetoglu and R. Das, "A comprehensive review on detection of cyber-attacks: Data sets, methods, challenges, and future research directions," *Internet of Things*, vol. 20, p. 100615, Nov. 2022, doi: 10.1016/j.iot.2022.100615.
[18] J. Liao, X. Wu, Y. Wu, and J. Shu, "K-NNDP: K-means algorithm based on nearest neighbor density peak optimization and outlier removal," *Knowledge-Based Systems*, vol. 294, p. 111742, Jun. 2024, doi: 10.1016/j.knosys.2024.111742.
[19] A. Kumar, A. Kumar, R. Mallipeddi, and D.-G. Lee, "High-density cluster core-based k-means clustering with an unknown number of clusters," *Applied Soft Computing*, vol. 155, p. 111419, Apr. 2024, doi: 10.1016/j.asoc.2024.111419.