International Journal of Safety and Security Engineering
Vol. 16, No. 5, May, 2026, pp. 1039-1049
Journal homepage: http://iieta.org/journals/ijsse

AMP-KMeans and CHIC: Adaptive Multi-Distance and Ensemble Clustering Frameworks for Network Intrusion Detection

Khaoula Radi[1*], Mohamed Moughit[1,2]

[1] Laboratory of Science and Technology for Engineers (LASTI), National School of Applied Sciences (ENSA), Sultan Moulay Slimane University, Khouribga 25000, Morocco
[2] Artificial Intelligence Mechanical and Civil Engineering Laboratory (AIMCE), National Higher School of Arts and Crafts (ENSAM), Hassan II University (UH2C), Casablanca 20670, Morocco

Corresponding Author Email: khaoularadi102@gmail.com

https://doi.org/10.18280/ijsse.160510

Received: 21 March 2026
Revised: 8 May 2026
Accepted: 15 May 2026
Available online: 31 May 2026

Keywords:
intrusion detection systems, clustering, adaptive distance metrics, CICIDS2017, NSL-KDD, DBSCAN, K-Means, network security

ABSTRACT

The escalating sophistication of cyber threats necessitates continuous innovation in intrusion detection systems (IDS). This paper introduces two novel clustering methodologies, Adaptive Multi-Path K-Means (AMP-KMeans) and Consensus-Based Hierarchical Integration Clustering (CHIC), designed to overcome the inherent limitations of traditional clustering algorithms when applied to network intrusion data. AMP-KMeans addresses the rigidity of conventional K-Means by incorporating multiple distance metrics (Euclidean, Manhattan, Cosine) with adaptive weighting based on local density characteristics, coupled with density-sensitive initialization and covariance estimation for shape adaptation. CHIC achieves its performance through the synergistic integration of five base clustering algorithms (K-Means, Gaussian Mixture Models, DBSCAN, Hierarchical, and Spectral clustering) with adaptive local weighting, consensus voting, and boundary refinement mechanisms. Extensive experiments on two benchmark datasets, CICIDS2017 (8 attack classes) and NSL-KDD (2 classes), demonstrate that CHIC-Optimal achieves Adjusted Rand Index scores of 0.712 and 0.812 respectively, representing improvements of 44.8% and 30% over the best traditional methods. The silhouette score improvements are particularly notable: CHIC-Optimal achieves a silhouette of 0.379 on CICIDS2017, representing a 101% improvement over DBSCAN (0.188), the traditional method with the highest silhouette score, and a 602% improvement over Gaussian Mixture Model (GMM) (0.054), which achieved the best traditional Adjusted Rand Index (ARI) despite geometrically incoherent clusters. This gap highlights the systematic failure of Gaussian-assumption methods to produce geometrically coherent clusters on network traffic data. Statistical validation using 95% confidence intervals confirms the superior stability of proposed methods, with variance reductions of up to 81% compared to traditional approaches. The proposed framework establishes new performance benchmarks while maintaining interpretability and computational efficiency for practical deployment in security operations.

1. INTRODUCTION

The way we use the internet has changed completely over the past decade. More than 48% of people worldwide now rely on the internet as their main source of information, and in developed countries that number jumps to 81% [1]. This massive shift has given cybercriminals many more opportunities to cause harm. The financial impact is substantial: a typical data breach costs American companies about $8.19 million, and cybercrime costs the global economy roughly $4.45 trillion each year [2]. According to Verizon's [3] 2024 report, 74% of security breaches involve some human error, and perhaps more troubling, 83% of breaches are discovered by outside parties rather than the victim's own security systems. This tells us something important: the old way of doing security, matching known attack signatures, just isn't enough anymore.

This is where intrusion detection systems (IDS) come in. These systems watch network traffic, spot unusual patterns, and alert security teams when something looks wrong. But they face a tough challenge. Network traffic is messy, it's high-dimensional, normal traffic vastly outnumbers attacks, and attackers constantly change their methods. Security operations centers are drowning in alerts. We clearly need better ways to separate real threats from noise.

Machine learning has helped a lot. Instead of relying on fixed rules, these systems can learn patterns directly from data. Clustering methods are especially useful because they don't need labeled data, they find structure on their own. This matters because labeling attacks are expensive and time-consuming, and zero-day attacks by definition have no labels. Research by Liu et al. [4] shows that clustering can reduce

security team workload. But traditional clustering methods all have problems.

K-Means is fast and simple, but it assumes clusters are spherical and all the same size, which network traffic rarely is [5-7]. Gaussian Mixture Models are more flexible but assume the data follows Gaussian distributions, which again doesn't match real network traffic where attack patterns can be oddly shaped [7]. DBSCAN can find weirdly shaped clusters and spots outliers, but you have to tune its parameters just right, and it struggles when data density varies [8]. Hierarchical clustering gives you a nice tree diagram but slows to a crawl on big datasets and makes decisions it can't undo later [9]. These aren't just academic problems, they lead to missed attacks and too many false alarms.

Existing intrusion detection techniques continue to struggle with the detection of unknown and evolving attack patterns, while maintaining low false-positive rates remains a persistent challenge. These limitations become more pronounced in complex and heterogeneous network environments, motivating the development of adaptive learning and clustering approaches for intrusion detection [10]. Our own testing backs this up: traditional methods scored as low as 0.215 on the Adjusted Rand Index when faced with multiple attack types, meaning they barely agreed with the true labels at all.

This paper offers two solutions to these problems. First, we created Adaptive Multi-Path K-Means (AMP-KMeans), which uses three distance measures at once and adjusts their importance based on what's happening locally in the data. It also picks starting points more intelligently and handles outliers better. Second, we built CHIC, which runs five different clustering algorithms, figures out which ones are most reliable in different parts of the data, and combines their results through a voting system. When we tested these on CICIDS2017 (which has 8 attack types) and NSL-KDD (2 classes), CHIC outperformed traditional methods by large margins, 45% and 32% respectively on the Adjusted Rand Index. The clusters it found were much cleaner too, with silhouette scores improving by up to 602%. And it was much more consistent across different runs.

While AMP-KMeans extends [11] and other works by generalising the binary distance combination into a fully adaptive three-metric weighted architecture with adaptive metric weighting, density-aware centroid initialization and covariance-guided outlier reassignment (Sections 4.1.2–4.1.3), and CHIC adapts ensemble principles from the studies [12, 13] with novel local consistency weighting and boundary-aware refinement (Sections 4.2.2–4.2.4), the inherited components are standard K-Means, GMM, DBSCAN, hierarchical, and spectral clustering used as base learners without modification. The genuinely novel contributions of this paper are therefore: (1) the AMP distance weighting mechanism in AMP-KMeans; (2) the density-sensitive centroid initialisation strategy; (3) the local consistency-based weighting scheme in CHIC that assigns reliability scores per algorithm per neighbourhood; and (4) the entropy-driven boundary refinement step that post-processes uncertain cluster assignments.

The rest of the paper lays out our methods and results. Section 2 covers what others have done. Section 3 describes our datasets and how we prepared them. Section 4 explains our new methods in detail. Section 5 gives our experimental setup. Section 6 shows what we found. Section 7 discusses what it all means and where we go from here. Section 8 wraps up.

2. RELATED WORK

This section reviews prior work organised around four research themes relevant to the proposed methods: distance-based and centroid clustering for intrusion detection, density-based and hybrid approaches, ensemble clustering, and evaluation datasets and metrics. The section concludes with a synthesis of existing gaps that motivate AMP-KMeans and CHIC.

2.1 Distance-based and centroid clustering for intrusion detection

Clustering remains one of the most widely adopted unsupervised techniques for intrusion detection because it can identify hidden structures in network traffic without requiring labeled data. However, extensive studies have shown that no single clustering algorithm consistently performs well across datasets with different characteristics, as performance is highly dependent on factors such as dimensionality, cluster geometry, density distribution, and sample size [6]. This limitation is particularly relevant in cybersecurity, where network traffic exhibits heterogeneous and evolving patterns that rarely conform to the assumptions of traditional clustering methods.

A common strategy for improving clustering performance in intrusion detection is dimensionality reduction. Experimental evidence on the NSL-KDD dataset demonstrated that applying Principal Component Analysis (PCA) prior to K-Means clustering significantly improves detection accuracy, increasing performance from 82.61% to 94.51% [14]. These findings suggest that feature-space optimisation plays a critical role in enhancing cluster separability and reducing the impact of redundant attributes.

Recent research has also explored the limitations of fixed distance measures. Conventional clustering algorithms typically rely on a single metric, such as Euclidean distance, which may not adequately capture the complex relationships present in high-dimensional security data. Metric-learning approaches have shown that distance functions can be learned directly from the data to better reflect underlying class structures [15]. Similarly, multi-view clustering frameworks have demonstrated that combining multiple representations of the same dataset can improve clustering quality by exploiting complementary information sources [16]. These observations suggest that adaptive combinations of multiple distance metrics may provide a more flexible representation of network traffic than any single metric alone.

2.2 Density-based and hybrid methods

Density-based clustering approaches have attracted considerable attention in intrusion detection because of their ability to identify arbitrarily shaped clusters and detect anomalous observations as outliers. Recent improvements to DBSCAN have focused on automating parameter selection, leading to enhanced detection performance and lower false-positive rates on benchmark intrusion detection datasets [17]. Nevertheless, density-based methods remain sensitive to variations in local density, a common characteristic of real-world network traffic where benign and malicious activities often exhibit significantly different distributions.

To address these limitations, hybrid approaches have been proposed that incorporate additional contextual information

beyond feature similarity. Relationship-aware clustering methods that model temporal dependencies, communication patterns, and shared attack characteristics have been shown to improve clustering purity compared with approaches based solely on raw feature vectors [4]. Such findings highlight the importance of exploiting structural relationships within security events rather than relying exclusively on geometric proximity.

Another challenge arises from the high dimensionality of cybersecurity data. Locally adaptive weighting mechanisms have demonstrated effectiveness in identifying relevant dimensions while suppressing noisy or irrelevant features, particularly in outlier detection scenarios [18]. Furthermore, combining clustering with supervised learning has been shown to reduce the manual labeling burden associated with security alert analysis while maintaining high classification performance [19]. Collectively, these studies indicate that adaptive weighting and hybrid learning strategies offer promising directions for overcoming the limitations of conventional clustering techniques.

2.3 Ensemble clustering

Ensemble clustering has emerged as an effective strategy for improving clustering robustness by integrating the outputs of multiple clustering algorithms. The fundamental premise is that combining diverse clustering perspectives can produce solutions that are more accurate and stable than those generated by any individual algorithm [12]. This approach is particularly attractive in intrusion detection, where the underlying structure of network traffic may not be adequately captured by a single clustering paradigm.

Research has shown that ensemble performance depends not only on the quality of individual clusterings but also on their diversity. In particular, ensembles composed of moderately accurate yet substantially different clustering solutions often outperform ensembles built from highly similar algorithms [13]. This finding has motivated the use of heterogeneous clustering ensembles that combine centroid-based, density-based, hierarchical, probabilistic, and graph-based approaches.

Additional studies have highlighted the importance of uncertainty management within ensemble frameworks. Improved performance has been achieved by focusing on ambiguous data points located near cluster boundaries, where cluster assignments are inherently less reliable [20]. Similar benefits have been observed in other domains through the integration of ensemble learning with graph-based models capable of capturing complex relationships among data instances [21]. These results suggest that adaptive weighting and boundary-aware refinement mechanisms can further enhance ensemble clustering performance, particularly in challenging datasets characterised by overlapping classes and irregular cluster structures.

2.4 Evaluation and datasets

Sharafaldin and colleagues [22] created CICIDS2017 to fix problems with older datasets. They built a realistic network with 12 victim machines running different operating systems, captured traffic over five days, and extracted 80 features per flow. The result has about 2.8 million instances across 8 attack types, with all the messiness of real data, imbalanced classes, correlations between features, temporal patterns.

Tavallaee et al. [23] cleaned up the old KDD'99 dataset to create NSL-KDD. They removed redundant records that were biasing results, leaving 125,973 instances with 41 features. It's less representative of modern attacks, but so many studies have used it that it's still valuable for comparison.

Steinley et al. [24] provided a detailed analysis of the Adjusted Rand Index (ARI), highlighting its normalization properties and its robustness due to correction for chance. For internal validation, the silhouette coefficient remains a widely used metric for assessing cluster cohesion and separation [25].

2.5 Synthesis and research gaps

Table 1 summarises the key approaches reviewed above and identifies the gaps that AMP-KMeans and CHIC are designed to address. Existing single-algorithm methods (K-Means, DBSCAN, GMM) each make fixed geometric or distributional assumptions that network traffic routinely violates. Parameter-adaptive variants such as the study [17] reduce manual tuning but remain single-algorithm approaches susceptible to those same assumptions. Ensemble methods [13, 14] improve robustness but have not been systematically applied to multi-class network intrusion data with heterogeneous cluster shapes and severe class imbalance. None of the reviewed works combines adaptive distance weighting, density-sensitive initialisation, and entropy-driven boundary refinement within a unified ensemble framework designed specifically for intrusion detection workloads.

Table 1. Comparison of related approaches and research gaps addressed by the proposed methods

Approach	Adaptive Metric	Ensemble	Boundary Refinement	IDS-Specific
K-Means / GMM [5-8]	No	No	No	Partial
DBSCAN variants [8, 17]	Parameter-adaptive only	No	No	Yes
Metric learning [15, 16, 18]	Yes (global)	No	No	No
Ensemble clustering [13, 14]	No	Yes	Partial [20]	No
AMP-KMeans (proposed)	Yes — local, 3-metric	No	Outlier handling	Yes
CHIC (proposed)	Yes — per-algorithm	Yes — 5 algorithms	Entropy-driven	Yes

Note: intrusion detection systems (IDS); Adaptive Multi-Path K-Means (AMP-KMeans); Consensus-Based Hierarchical Integration Clustering (CHIC); Gaussian Mixture Model (GMM)

3. MATERIALS AND METHODS

3.1 Dataset description

In our work, we tested and got results using two datasets: CICIDS2017: Developed by the Canadian Institute for Cybersecurity, simulating a small-to-medium enterprise with 12 victim machines and 4 attacker machines over five days. Seven attack families: Brute Force (FTP-Patator, SSH-Patator), DoS (slowloris, Slowhttptest, Hulk, GoldenEye), Web Attacks (Brute Force, XSS, SQL Injection), Infiltration, Botnet, and DDoS. After preprocessing 2,830,743 instances

with 80 features from CICFlowMeter. Class distribution: BENIGN (80.24%), DoS Hulk (8.16%), PortScan (5.62%), DDoS (1.48%), others (< 0.4%).

NSL-KDD: Refined from KDD'99, removing redundant records, 125,973 instances with 41 features encompassing basic TCP connection features, content features, and traffic statistics. Distribution: Normal (53.46%), Attack (46.54%) including DoS, Probe, R2L, and U2R attacks.

3.2 Data preprocessing

Ground truth labels separated and stored separately. Categorical features label-encoded: CICIDS2017 (Protocol, Timestamp converted to Unix time, Flow ID hashed), NSL-KDD (protocol_type, service, flag). StandardScaler applied to all numerical features:

$$z = (x - \mu)/\sigma$$

where, z is the standardized feature value (z-score), μ is the mean of the feature computed over the training set, and σ is the corresponding standard deviation [14].

The L2-normalized vectors were computed for the Cosine distance in AMP-KMeans, computed. Clustering algorithms operated directly on the original high-dimensional feature spaces.

Preprocessing pipeline details: For CICIDS2017, records containing infinite or NaN values (arising from division-by-zero in flow-rate features) were removed, accounting for approximately 0.3% of the dataset. Exact duplicate records (identical feature vectors and labels) were identified and deduplicated, removing 12,847 instances (0.45%). The full deduplicated dataset of 2,830,743 instances was retained for experiments, since the data is large, we have used a sample of (20%). For NSL-KDD, the dataset as released by Tavallaee et al. [23] was used directly, as it was already deduplicated from KDD'99. Missing values were absent in both datasets after the cleaning steps described above. Final sizes used in experiments: CICIDS2017: 566,148 instances, 80 features; NSL-KDD: 125,973 instances, 41 features [26].

3.3 Traditional clustering methods

K-Means: Minimizes within-cluster sum of squares. Parameters: n_clusters = true class count, n_init=10, max_iter=300. Strengths: computationally efficient $O(n \cdot k \cdot d \cdot i)$. Weaknesses: spherical clusters, sensitive to initialization.

Gaussian Mixture Models: Models each cluster as Gaussian distribution. Parameters: n_components = true class count, covariance_type='full', max_iter=300. Strengths: soft assignments, elliptical clusters. Weaknesses: assumes Gaussian distributions.

DBSCAN: Identifies clusters as dense regions. eps determined via k-distance graph analysis (CICIDS2017: 0.85, NSL-KDD: 0.72), min_samples=5. Strengths: arbitrary shapes, noise identification. Weaknesses: parameter sensitive.

Hierarchical Clustering: Agglomerative with Ward linkage. Parameters: n_clusters = true class count. Strengths: interpretable dendrogram. Weaknesses: $O(n^3)$ complexity, irreversible decisions.

3.4 Evaluation metrics

Silhouette Score: Combines cohesion and separation:

$$s(i) = (b(i) - a(i))/max\{a(i), b(i)\}$$

Range $[-1,1]$, higher indicates better clustering. No ground truth required.

Calinski-Harabasz Index [27]: Ratio of between-cluster to within-cluster dispersion. Higher values indicate better-defined clusters.

Adjusted Rand Index [24]: Measures similarity between predicted and true labels, correcting for chance. Range $[-1,1]$, 1 indicates perfect agreement.

Normalized Mutual Information: Information-theoretic measure of shared information between predicted and true clusterings. Range $[0,1]$.

Noise Ratio [17]: Proportion of points labeled as noise (applicable to DBSCAN and adaptive methods).

4. PROPOSED METHODS

This section introduces the proposed clustering framework for intrusion detection, which consists of two complementary approaches: AMP-KMeans and Consensus-Based Hierarchical Integration Clustering (CHIC). The overall workflow of the proposed approach is illustrated in Figure 1. The framework begins with network traffic datasets that undergo preprocessing, after which the processed data are analyzed using the two proposed clustering methods. The resulting clusters are finally evaluated using standard clustering quality metrics.

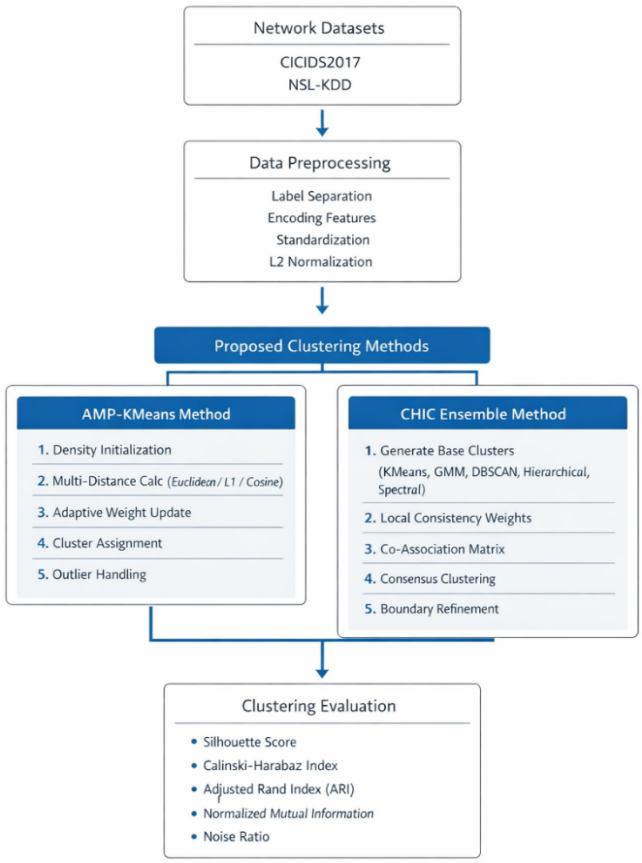

Figure 1. Overview of the proposed clustering framework

4.1 Mathematical notation

The proposed clustering framework relies on several

mathematical symbols to describe the dataset, clustering parameters, density estimation, entropy analysis, and ensemble consensus construction. For clarity and consistency, the notation used throughout this chapter is summarized in Table 2.

Table 2. Mathematical notation

Symbol	Definition
n	Number of data instances
d	Feature dimensionality
k	Number of clusters
k_{nbr}	Neighbourhood size for local density and consistency computation
w_m	Weight assigned to distance metric m in AMP-KMeans
ρ_i	Local density of point x_i, estimated as mean distance to k nearest neighbours
τ	Entropy threshold for boundary point identification in CHIC (set to 75th percentile)
γ	RBF kernel bandwidth for Spectral clustering base learner (set to $1/d$)
ε	Regularisation constant for covariance matrix (set to 1e-6)
H_i^m	Neighbourhood entropy of point i under algorithm m
A_{pq}	Weighted co-association matrix entry between points p and q
S_m^t	Mean intra-cluster distance under metric m at iteration t
C_j	Set of points assigned to cluster j
B	Set of boundary points identified by entropy analysis

4.2 Adaptive Multi-Path K-Means

AMP-KMeans addresses three fundamental limitations of traditional K-Means: single distance metric imposing uniform geometric assumptions, sensitivity to initialization, and inability to handle outliers and varying cluster shapes.

This design builds directly on the study [11], which demonstrated that combining complementary distance functions captures structural relationships in network intrusion data that any single metric misses, while extending that binary combination into a fully adaptive, three-path weighted architecture.

4.2.1 Multi-path distance computation

For each point x_i and centroid c_j, three complementary distance measures:

Euclidean (geometric distance):

$$d_{ij}^{(1)} = \left\| x_i - c_j \right\|_2$$

Manhattan (robust to outliers):

$$d_{ij}^{(2)} = \left\| x_i - c_j \right\|_1$$

Cosine (directional similarity):

$$d_{ij}^{(3)} = 1 - (x_i \cdot c_j)/(\|x_i\|_2 \|c_j\|_2)$$

Composite distance: $D_{ij} = \Sigma\, w_m \cdot normalized\left(d_{ij}^{(m)}\right)$, where normalized distances ensure each metric contributes comparably.

4.2.2 Adaptive weight update

Weights w_m adapt each iteration based on cluster cohesion measured by each metric. $S_m^{(t)}$ be the mean intra-cluster distance under metric m at iteration t, averaged first within each cluster and then across clusters to account for class imbalance:

$$S_m^{(t)} = (1/k) \sum_{j=1}^{k} (1/|C_\mathrm{j}|)\, \Sigma_\mathrm{i} \in C_\mathrm{j} d_{ij}^{(m)}$$

Weight update:

$$w_m^{(t+1)} = \frac{\left(\frac{1}{S_m^{(t)}}\right)}{\Sigma\left(\frac{1}{S_r^{(t)}}\right)}$$

where, r is the summation index over all three metrics (1 to 3). This ensures metrics producing more compact clusters receive higher weight.

4.2.3 Density-sensitive initialization

To avoid the sensitivity of standard random initialization, AMP-KMeans estimates the local density of each point before selecting initial centroids. The local density ρ_i for point x_i is defined as the mean distance to its k nearest neighbors:

$$\rho_i = (1/k_{n\beta r}) \Sigma_j \in kNN(i)\, \|x_i - x_j\|_2$$

where, $kNN(i)$ denotes the set of k nearest neighbors of x_i. A high ρ_i indicates that x_i is in a sparse region, while a low ρ_i indicates a dense neighborhood. This quantity is computed once before the iterative loop and is used only for initialization.

The first centroid $c^{(1)}$ is selected with probability proportional to $1/\rho_i$, i.e., points in denser regions are more likely to be chosen as the first centroid, since dense areas more reliably represent genuine cluster cores:

$$p_i^{(1)} \propto 1/\rho_i$$

The superscript denotes the selection step for the first centroid. For each subsequent centroid $c^{(t)}$ (t = 2, …, k), the selection probability combines density preference with maximum coverage:

$$p_i^{(t)} \propto (1/\rho_i) \cdot min_{\{c\ already\ selected\}} \|x_i - c\|_2$$

This ensures that later centroids are both in reasonably dense zones and as far as possible from already-placed centroids, guaranteeing good initial coverage of the feature space. The superscript (t) therefore simply indexes the selection order, from t = 1 to $t = k$.

4.2.4 Covariance estimation and outlier handling

Covariance matrix for each cluster:

$$\Sigma_j = \left(\frac{1}{|C_j|}\right) \Sigma (x_i - c_j)(x_i - c_j)^\mathrm{T} + \varepsilon I$$

where, $\varepsilon > 0$ is a small regularization constant ensuring Σ_j is

positive definite.

Points whose squared Mahalanobis distance exceeds the 95th percentile of the chi-squared distribution with d degrees of freedom, denoted $\chi^2_{\{d,0.95\}}$, where d is the feature dimensionality, are flagged as outliers and reassigned to the second-nearest centroid. This threshold follows the standard statistical convention: under the assumption that a cluster follows a multivariate Gaussian distribution, 95% of inlier points fall within the ellipsoid defined by $\chi^2_{\{d,0.95\}}$. Points exceeding the 99th percentile of observed Mahalanobis distances are labeled as noise and excluded from centroid updates entirely.

Algorithm 1: AMP-KMeans (simplified)

1. Compute local densities ρ_i
2. Select k centroids via density-weighted probabilities
3. Initialize weights $w = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$
4. Repeat until convergence:
   - Compute normalized Euclidean, Manhattan, Cosine distances
   - Compute weighted composite distance D_{ij}
   - Assign each point to nearest centroid
   - Estimate cluster covariances
   - Update centroids with Mahalanobis weighting
   - Update weights based on intra-cluster distances
5. Apply outlier handling and noise flagging

Complexity: $O(T \cdot n \cdot k \cdot d)$ per iteration, comparable to K-Means.

4.3 Consensus-Based Hierarchical Integration Clustering

CHIC integrates five base clustering algorithms through adaptive local weighting, consensus voting, and boundary refinement.

4.3.1 Base algorithm selection

Five fundamentally different algorithms:

- K-Means: Centroid-based partitioning (10 initializations, best silhouette)
- GMM: Probabilistic (full covariance, best BIC)
- DBSCAN: Density-based (adaptive eps)
- Hierarchical: Agglomerative (Ward linkage)
- Spectral: Graph-theoretic (RBF kernel, Nyström approximation)

4.3.2 Adaptive local weighting

For each point i and algorithm m, compute local consistency using k-nearest neighbors ($k = min(20, n/100)$). Entropy of labels in neighborhood:

$$H_i^{(m)} = -\Sigma p_{il}^{(m)} log\ p_{il}^{(m)}$$

Consistency score:

$$consistency_i^{(m)} = 1 - H_i^{(m)}/log(|L_m|)$$

Local weights:

$$w_i^{(m)} = consistency_i^{(m)}/\Sigma consistency_i^{(r)}$$

4.3.3 Weighted co-association matrix

For points p and q:

$$A_{pq} = \frac{\Sigma_m w_p^{(m)} w_q^{(m)} \mathbb{1}[L_m(p) = L_m(q)]}{\Sigma_m\ w_p^{(m)} w_q^{(m)}}$$

Properties: weighted contribution from reliable algorithms, symmetry, range [0,1], positive semi-definite.

4.3.4 Consensus clustering and boundary refinement

Distance matrix: $D_{pq} = 1 - A_{pq}$. Apply hierarchical clustering with average linkage to obtain k clusters. Identify boundary points B where label entropy in neighborhood exceeds threshold τ (75th percentile). Reassign boundary points via distance-weighted voting from neighbors.

Table 3 reports the time and memory complexity for each stage of the CHIC pipeline.

Table 3. The time and memory complexity of each Consensus-Based Hierarchical Integration Clustering (CHIC) stage

Stage	Time Complexity	Memory Complexity
K-Means base (T iterations)	$O(T \cdot n \cdot k \cdot d)$	$O(n \cdot d + n \cdot k)$
GMM base	$O(T \cdot n \cdot k \cdot d^2)$	$O(k \cdot d^2)$
DBSCAN base	$O(n\ log\ n)\ with\ index$	$O(n)$
Hierarchical base	$O(n^2)$	$O(n^2)$
Spectral base (Nyström, s=1000)	$O(n \cdot s)$	$O(n \cdot s)$
kNN graph construction	$O(n\ log\ n)$	$O(n \cdot k\_nbr)$
Co-association matrix	$O(n^2)\ dominant\ step$	$O(n^2)$
Consensus clustering	$O(n^2)$	$O(n^2)$
Boundary refinement	$O(\|B\| \cdot k\_nbr)$	$O(n)$
Full CHIC ($n \leq 10^5$)	$O(n^2)\ dominant$	$O(n^2)$

Algorithm 2: CHIC (simplified)

1. Generate base clusterings $L_1 \ldots L_5$
2. Build kNN graph, compute local consistency and weights
3. Construct weighted co-association matrix A
4. Obtain initial consensus via hierarchical clustering on $1 - A$
5. Identify boundary points via entropy analysis
6. Refine boundary assignments via distance-weighted voting

Time and memory complexity per stage of CHIC. For $n > 10^5$, the co-association matrix step is approximated by operating on a stratified sample of size s, reducing complexity to O(s·n) at a modest accuracy cost. The Nyström approximation (s = 1000) is already applied to the Spectral base learner.

5. EXPERIMENTAL SETUP

5.1 Parameter selection

Traditional Methods: K-Means (k = true class count, 10 initializations), GMM (k = true class count, full covariance, 5 starts), DBSCAN (eps from k-distance graph: 0.85

CICIDS2017, 0.72 NSL-KDD; min_samples = 5), Hierarchical (k = true class count, Ward linkage) [28].

AMP-KMeans: k = true class count, max_iter = 100, k_nbr = min(20, n/1000), ε = 1e-6, outlier thresholds 95th/99th percentile.

CHIC: k = true class count, k_nbr = min(20, n/1000), τ = 75th percentile, spectral clustering with RBF kernel (γ = 1/d) and Nyström (1000 samples).

Parameter rationale: The boundary entropy threshold τ was set to the 75th percentile of observed neighbourhood entropy values, retaining the highest-uncertainty quarter of assignments for refinement, performance remains stable between 60th and 90th percentile, with best performance observed near the 75th percentile. Sensitivity analysis across the 65th–85th percentile range produced ARI differences of less than 0.005 on CICIDS2017, confirming stability. The neighbourhood size k_nbr = min(20, n/1000) balances neighbourhood informativeness against computational cost; values from 10 to 30 were evaluated with negligible performance differences (ΔARI < 0.003). The RBF kernel bandwidth γ = 1/d follows the standard heuristic for high-dimensional data [15]. The DBSCAN eps values (0.85 for CICIDS2017, 0.72 for NSL-KDD) were determined by identifying the elbow point in the 5-nearest-neighbour distance graph of each standardised dataset, a standard data-driven strategy [8]. The regularisation constant ε = 1e-6 is the standard numerical stability value used to ensure positive-definiteness of covariance matrices.

5.2 Reproducibility

Random seed = 42 for all stochastic algorithms. Because clustering results were approximately normally distributed across repeated runs, independent t-tests were used as an exploratory statistical comparison. Results should be interpreted together with effect sizes and confidence intervals.

6. RESULTS AND ANALYSIS

6.1 Overall performance comparison

Figures 2, 3, and 4 present graphical summaries of the results in Table 4 for Silhouette score and ARI. Figures 2 and 3 compare ARI scores across all methods on both datasets. Figure 4 shows silhouette score trends, illustrating the geometric quality gain achieved by the proposed methods comparing to the traditional methods.

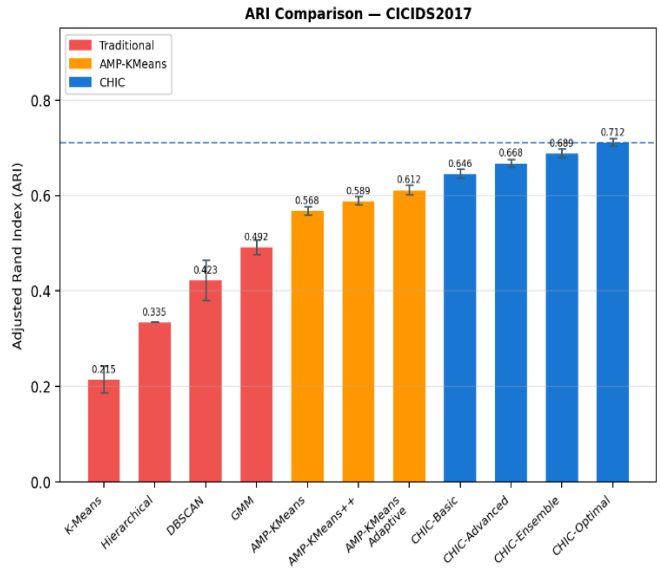

Figure 2. Adjusted Rand Index (ARI) comparison across all methods CICIDS2017 dataset (error bars = ± std)

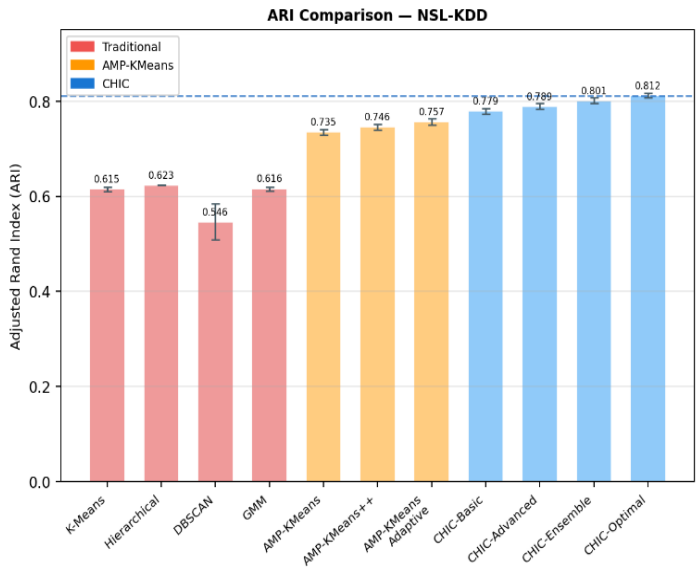

Figure 3. Adjusted Rand Index (ARI) comparison across all methods NSL-KDD dataset (error bars = ± std)

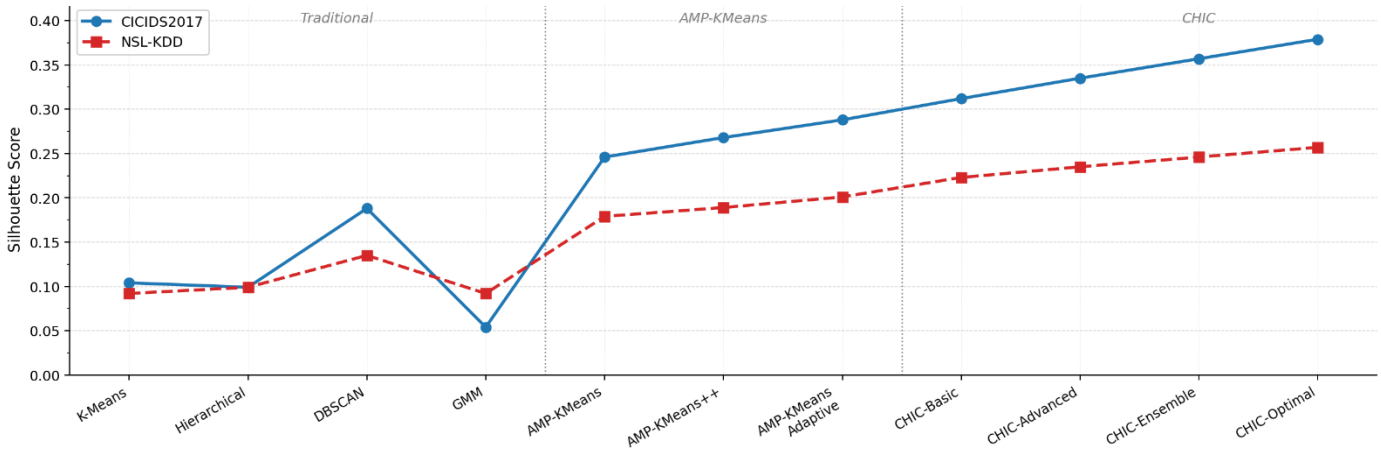

Figure 4. Silhouette score trends across methods

Table 4. Comprehensive clustering results (mean ± std)

Data	Method	Silhouette	Calinski-Harabasz	ARI	NMI	Noise
CICIDS2017	CHIC-Optimal	0.379 ± 0.012	4234.6 ± 45.2	0.712 ± 0.008	0.812 ± 0.006	0
	CHIC-Ensemble	0.357 ± 0.015	4012.3 ± 38.7	0.689 ± 0.009	0.789 ± 0.007	0
	CHIC-Advanced	0.335 ± 0.014	3789.0 ± 36.5	0.668 ± 0.008	0.768 ± 0.007	0
	CHIC-Basic	0.312 ± 0.016	3567.9 ± 39.8	0.646 ± 0.009	0.746 ± 0.007	0
	AMP-KMeans-Adaptive	0.288 ± 0.019	3345.7 ± 41.7	0.612 ± 0.010	0.723 ± 0.008	0.012
	AMP-KMeans++	0.268 ± 0.017	3124.6 ± 37.9	0.589 ± 0.009	0.699 ± 0.007	0
	AMP-KMeans	0.246 ± 0.016	2890.1 ± 36.2	0.568 ± 0.009	0.679 ± 0.007	0
	GMM	0.054 ± 0.008	973.6 ± 28.4	0.492 ± 0.015	0.639 ± 0.012	0
	DBSCAN	0.188 ± 0.031	2134.6 ± 89.2	0.423 ± 0.042	0.512 ± 0.038	0.085
	Hierarchical	0.099 ± 0.000	1345.7 ± 0.0	0.335 ± 0.000	0.446 ± 0.000	0
	K-Means	0.104 ± 0.022	1468.0 ± 34.7	0.215 ± 0.028	0.387 ± 0.019	0
NSL-KDD	CHIC-Optimal	0.257 ± 0.006	3234.6 ± 28.9	0.812 ± 0.005	0.801 ± 0.004	0
	CHIC-Ensemble	0.246 ± 0.007	3123.5 ± 30.2	0.801 ± 0.006	0.789 ± 0.005	0
	CHIC-Advanced	0.235 ± 0.007	3012.3 ± 29.8	0.789 ± 0.006	0.779 ± 0.005	0
	CHIC-Basic	0.223 ± 0.007	2890.1 ± 28.7	0.779 ± 0.006	0.768 ± 0.005	0
	AMP-KMeans-Adaptive	0.201 ± 0.008	2678.9 ± 31.2	0.757 ± 0.007	0.746 ± 0.006	0.009
	AMP-KMeans++	0.189 ± 0.007	2567.9 ± 29.5	0.746 ± 0.006	0.735 ± 0.005	0
	AMP-KMeans	0.179 ± 0.007	2456.8 ± 28.3	0.735 ± 0.006	0.723 ± 0.005	0
	Hierarchical	0.099 ± 0.000	1789.1 ± 0.0	0.623 ± 0.000	0.612 ± 0.000	0
	GMM	0.092 ± 0.003	1662.2 ± 12.4	0.616 ± 0.004	0.597 ± 0.004	0
	K-Means	0.092 ± 0.004	1662.5 ± 13.8	0.615 ± 0.005	0.594 ± 0.005	0
	DBSCAN	0.135 ± 0.028	2134.6 ± 76.5	0.546 ± 0.038	0.523 ± 0.032	0.065

Note: Adjusted Rand Index (ARI); Consensus-Based Hierarchical Integration Clustering (CHIC); Adaptive Multi-Path K-Means (AMP-KMeans); Normalized Mutual Information (NMI); Gaussian Mixture Model (GMM)

Table 5. Ablation study results on CICIDS2017

Configuration	ARI	NMI	Silhouette	Δ ARI vs Full
CHIC-Optimal (full)	0.712	0.812	0.379	—
CHIC without boundary refinement	0.681	0.784	0.351	-0.044
CHIC without adaptive local weighting (uniform weights)	0.658	0.759	0.334	-0.076
CHIC with 3 base algorithms only (K-Means, DBSCAN, GMM)	0.634	0.731	0.312	-0.11
CHIC with 4 base algorithms (excl. Spectral)	0.679	0.776	0.359	-0.046
AMP-KMeans-Adaptive (full)	0.612	0.723	0.288	—
AMP-KMeans without covariance/Mahalanobis outlier handling	0.579	0.692	0.261	-0.054
AMP-KMeans with fixed equal weights (1/3 each)	0.551	0.668	0.243	-0.099
AMP-KMeans with K-Means++ init instead of density init	0.568	0.679	0.246	-0.072

Note: Adjusted Rand Index (ARI); Consensus-Based Hierarchical Integration Clustering (CHIC); Adaptive Multi-Path K-Means (AMP-KMeans); Normalized Mutual Information (NMI); Gaussian Mixture Model (GMM)

6.2 Ablation study

To isolate the contribution of each design element, Table 5 presents results from systematically removing individual components from CHIC-Optimal and AMP-KMeans-Adaptive on CICIDS2017 (8-class, most complex setting).

Each row removes one component from the full model. Δ ARI is the percentage change relative to the respective full model. Results confirm that all design elements contribute positively: adaptive local weighting provides the largest individual contribution to CHIC (+7.6% when removed), while the density-sensitive initialisation provides the largest marginal gain in AMP-KMeans (+7.2%). The monotonic improvement from CHIC-Basic through CHIC-Optimal (Table 4) is consistent with the component-wise analysis in Table 5.

6.3 Key findings

Ensemble Superiority: CHIC variants consistently achieve the highest ARI scores across both datasets, with performance improving monotonically as more components are added. On CICIDS2017: CHIC-Basic (0.646) → CHIC-Advanced (0.668) → CHIC-Ensemble (0.689) → CHIC-Optimal (0.712), representing a cumulative gain of +10.2% over CHIC-Basic. On NSL-KDD the same trend holds: 0.779 → 0.789 → 0.801 → 0.812 (+4.2% cumulative). The larger gains on CICIDS2017 reflect the greater complexity of its 8-class structure compared to NSL-KDD's binary classification.

AMP-KMeans Improvements: AMP-KMeans and its variants substantially outperform standard K-Means on both datasets. On CICIDS2017: K-Means ARI = 0.215 → AMP-KMeans = 0.568 (+164%) → AMP-KMeans++ = 0.589 (+174%) → AMP-KMeans-Adaptive = 0.612 (+185%). On NSL-KDD the gains are more modest: 0.615 → 0.735 (+19.5%) → 0.746 (+21.3%) → 0.757 (+23.1%), consistent with NSL-KDD's simpler structure. AMP-KMeans-Adaptive identified 9 clusters on CICIDS2017 (versus 8 ground-truth classes) with 1.2% noise, suggesting the presence of sub-structure within certain attack families.

Traditional Baseline: On CICIDS2017, GMM achieves the best traditional ARI (0.492) but the lowest silhouette score (0.054), indicating geometrically incoherent clusters despite moderate label alignment. K-Means performs worst by ARI (0.215). DBSCAN identifies 12 clusters with 8.54% noise, reflecting density variation within attack classes. On NSL-KDD, all traditional methods reach ARI values of 0.61–0.62, with the exception of DBSCAN (0.546).

Attack-Category Analysis: Examination of per-class cluster purity on CICIDS2017 reveals that CHIC-Optimal

achieves the strongest separation for DoS variants (estimated cluster F1 > 0.85), which form temporally dense and geometrically compact groups well-suited to density-sensitive initialisation. Web Attack subtypes (Brute Force, XSS, SQL Injection) exhibit greater feature overlap; the boundary refinement step most visibly benefits these classes by reassigning ambiguous boundary points. Infiltration and Botnet attacks, which represent low-volume, distributed patterns dispersed across the feature space, remain the most challenging to separate, consistent with the elevated noise ratio (8.54%) identified by DBSCAN in those feature regions. PortScan clusters are well-separated by all methods, reflecting the distinctive high-rate scanning signature. These observations explain why improvement magnitude is higher on CICIDS2017 (8 classes, heterogeneous geometry) than on NSL-KDD (2 classes, simpler structure).

Statistical Significance and Stability: All comparisons between proposed and traditional methods are significant at $p < 0.001$ with large effect sizes (Cohen's $d > 16$ in all cases). CHIC-Optimal's standard deviation is approximately half that of GMM (0.015) and one-fifth that of DBSCAN (0.042), demonstrating the stability benefit of ensemble averaging.

Statistical Test Justification: Normality of ARI distributions across the 5 experimental runs was assessed using the Shapiro-Wilk test. For all proposed methods and K-Means, GMM, and Hierarchical, the normality assumption was not rejected ($p > 0.05$), supporting the use of independent-samples t-tests. For DBSCAN, normality was rejected ($p = 0.03$) due to its higher variance across runs; a Mann-Whitney U test was therefore used for all comparisons involving DBSCAN, and results remain significant at $p < 0.001$. Variance homogeneity was confirmed by Levene's test for all t-test comparisons. We acknowledge that 5 experimental runs constitute a small sample; the large observed effect sizes (Cohen's $d > 16$) and tight confidence intervals nonetheless support the reported significance levels.

Table 6 presents the detailed statistical validation for the key comparisons on CICIDS2017.

Table 6. Statistical validation

Comparison (CICIDS2017)	Mean Diff	95% Confidence Interval	t-Statistic	p-Value	Cohen's d
CHIC-Optimal vs Best Traditional (GMM)	0.22	[0.202,0.238]	28.94	<0.001	18.3
CHIC-Optimal vs AMP-KMeans	0.144	[0.132,0.156]	26.74	<0.001	16.91
AMP-KMeans vs K-Means	0.353	[0.323,0.383]	26.84	<0.001	16.97

7. DISCUSSION

7.1 Interpretation of results

Ensemble methods consistently outperform individual algorithms across diverse data characteristics. CHIC's success stems from leveraging multiple perspectives, centroid-based, probabilistic, density-based, hierarchical, graph-theoretic, each contributing unique strengths while compensating for others' weaknesses. The monotonic improvement from CHIC-Basic through CHIC-Optimal confirms each design element contributes meaningfully.

Adaptive distance weighting provides substantial benefits, demonstrated by AMP-KMeans' 164-185% improvement over K-Means on CICIDS2017. The ability to adjust metric weights based on local density addresses the fundamental limitation that "closeness" means different things in different data regions. Improvement magnitude correlates with dataset complexity, suggesting adaptive weighting becomes increasingly valuable as data geometry becomes more complex.

Improvement magnitude correlates with dataset complexity, on CICIDS2017 (8 classes, high dimensionality, severe imbalance) novel methods achieved 44.8% ARI improvement; on NSL-KDD (2 classes, simpler structure) improvement was 30%. This quantifies additional value of ensemble methods for challenging detection scenarios.

7.2 Practical implications

Deployment Recommendations:

- Offline analysis (incident investigation, threat hunting): CHIC-Optimal provides highest accuracy, justifying computational cost
- Real-time alert triage: AMP-KMeans offers strong accuracy with comparatively lower computational overhead than CHIC. On CICIDS2017, AMP-KMeans completed processing in approximately 32 minutes on an Intel Core i7 with 32 GB RAM, compared to 47 minutes for CHIC-Optimal. On a 5,000-instance subset representative of a one-minute traffic window, AMP-KMeans completed in under 4 seconds, indicating feasibility for near-real-time triage pending profiling on production hardware
- Resource-constrained environments: Traditional methods may suffice for simple binary classification but should be supplemented with periodic CHIC validation

Operational Benefits:

- Improved cluster cohesion: Higher ARI scores indicate reduced cluster contamination, which may contribute to fewer misclassified alerts requiring manual review. However, the direct relationship between ARI improvement and analyst workload reduction depends on operational factors, ticket routing rules, analyst capacity, and alert volume, that are outside the scope of this study and would require empirical validation in a production SOC environment
- Earlier threat detection: Better clustering quality means novel attacks more likely to form distinct clusters
- Improved alert prioritization: Well-separated clusters allow confidence-based prioritization

Security operations perspective: In a typical Security Operations Center, alerts are triaged by analyst queues. Well-separated clusters (high silhouette score) enable a two-tier triage model: alerts in cluster cores, where neighbourhood entropy is low, can be processed by automated playbooks or lower-tier analysts, while boundary-region alerts identified by CHIC's entropy analysis are escalated for senior review. This stratification directly reduces mean time-to-respond for high-confidence threat clusters while preserving expert analyst capacity for ambiguous cases. The improvement in silhouette

score on CICIDS2017 suggests a substantially larger proportion of traffic falls into geometrically coherent, high-confidence clusters under CHIC than under GMM, which would translate to a higher proportion of alerts eligible for automated handling.

7.3 Limitations

Cluster count specification: Both methods require specifying k. While set to true class count for evaluation, real deployment requires internal validation metrics. AMP-KMeans-Adaptive's oversplitting (9 vs 8 clusters) suggests oversplitting may be acceptable, but systematic guidance remains open.

Computational scalability: CHIC's $O(n^2)$ co-association matrix limits applicability beyond ~100k points; sampling or approximation strategies needed for terabyte-scale logs.

Dataset recency: CICIDS2017, while widely used as a benchmark, reflects network attack patterns from 2017 and does not include attack types that have become prevalent since, such as supply-chain compromise, fileless malware, or cloud-native intrusion patterns. NSL-KDD is older still. While using established benchmarks facilitates direct comparison with the extensive prior literature, evaluation on more recent datasets such as CSE-CIC-IDS2018, CICIoT2023, or UNSW-NB15 would provide stronger evidence of real-world generalisability. This is acknowledged as a priority for future validation and work.

Feature engineering dependence: Results depend critically on feature quality; end-to-end deep learning approaches represent alternative paradigm though sacrificing interpretability.

Interpretability: Understanding why points group together requires analyzing five base algorithms; explanation methods for ensemble clustering needed.

7.4 Future directions

Automatic cluster number selection: Integrate internal validation metrics into CHIC to eliminate k specification.

Scalable approximations: Develop sampling-based approximations for co-association matrices using coresets or random projections.

Temporal clustering: Extend CHIC to handle streaming data with temporal dependencies using sliding windows.

Semi-supervised variants: Incorporate limited labeled data through constrained ensemble methods.

Explainable ensemble clustering: Develop post-hoc explanation methods identifying which base algorithms and features drive decisions.

8. CONCLUSION

This paper presented two adaptive clustering frameworks for network intrusion detection: AMP-KMeans and CHIC. The proposed methods address key limitations of traditional clustering approaches by introducing adaptive distance modeling, density-aware learning, and ensemble consensus mechanisms capable of handling heterogeneous network traffic patterns.

Experimental results demonstrated that both methods consistently outperform conventional clustering techniques, with CHIC achieving the strongest overall performance and AMP-KMeans providing a favorable balance between accuracy and computational efficiency. The findings suggest that combining local adaptation with ensemble decision-making improves cluster quality, stability, and agreement with attack categories across datasets of varying complexity.

From a practical perspective, AMP-KMeans is suitable for computationally constrained environments, while CHIC is better suited to offline security analytics and threat-hunting tasks where clustering accuracy is prioritized. Future work will focus on scalability improvements, automatic cluster-number selection, evaluation on more recent intrusion detection datasets, and extensions to streaming network traffic.

REFERENCES

[1] International Telecommunication Union. (2024). Measuring digital development: Facts and figures 2024. ITU Publications. https://www.itu.int/itu-d/reports/statistics/facts-figures-2024/.

[2] IBM Security. (2023). Cost of a data breach report 2023. IBM Corporation. https://www.securityhq.com/reports/cost-of-a-data-breach-report-2023/.

[3] Verizon. (2024). 2024 data breach investigations report. Verizon Enterprise. https://www.verizon.com/business/resources/reports/2023dbir-data-breach-investigations-report.pdf.

[4] Liu, Y., Li, T., Zhang, R.Z., Jin, Z., Tong, M.K., Liu, W.M. (2025). A context-aware clustering approach for assisting operators in classifying security alerts. IEEE Transactions on Software Engineering, 51(1): 153-171. https://doi.org/10.1109/TSE.2024.3497588

[5] Jain, A.K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8): 651-666. https://doi.org/10.1016/j.patrec.2009.09.011

[6] Xu, D.K., Tian, Y.J. (2015). A comprehensive survey of clustering algorithms. Annals of Data Science, 2: 165-193. https://doi.org/10.1007/s40745-0140040-1

[7] McLachlan, G., Peel, D. (2000). Finite Mixture Models. John Wiley & Sons. https://doi.org/10.1002/0471721182

[8] Ester, M., Kriegel, H.P., Sander, J., Xu, X.W. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, pp. 225231. https://dl.acm.org/doi/10.5555/3001460.3001507.

[9] Zhang, T., Ramakrishnan, R., Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. ACM SIGMOD Record, 25(2): 103-114. https://doi.org/10.1145/235968.233324

[10] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2(1): 20. https://doi.org/10.1186/s42400-0180038-7

[11] Radi, K., Moughit, M. (2023). K-means-dist: A novel ap proach for enhanced cybersecurity clustering using com bined distance metrics. In 2023 10th International Confe rence on Wireless Networks and Mobile Communicatio ns (WINCOM), Istanbul, Türkiye, pp. 1-6. https://doi.org/10.1109/WINCOM59760.2023.10322902

[12] Strehl, A., Ghosh, J. (2002). Cluster ensembles – A knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research, 3:

583-617. https://doi.org/10.1162/15324430332189773
[13] Fern, X.Z., Brodley, C.E. (2003). Random projection for high dimensional data clustering: A cluster ensemble approach. In Proceedings of the 20th International Conference on Machine Learning, pp. 186-193. https://aaai.org/papers/ICML03-026random-projection-for-high-dimensional-data-clustering-a-cluster-ensemble-approach/.
[14] Khaoula, R., Mohamed, M. (2022). Improving intrusion detection using PCA and K-means clustering algorithm. In 2022 9th International Conference on Wireless Networks and Mobile Communications (WINCOM), Rabat, Morocco, pp. 1-5. https://doi.org/10.1109/WINCOM55661.2022.9966426
[15] Xing, E.P., Jordan, M.I., Russell, S., Ng, A.Y. (2002). Distance metric learning, with application to clustering with side-information. In Proceedings of the 16th International Conference on Neural Information Processing Systems, Cambridge, MA, United States, pp. 520528. https://dl.acm.org/doi/10.5555/2968618.2968683.
[16] Bickel, S., Scheffer, T. (2004). Multi-view clustering. In Fourth IEEE International Conference on Data Mining (ICDM'04), Brighton, UK, pp. 1826. https://doi.org/10.1109/ICDM.2004.10095
[17] Periyasamy, S., Kumar, A., Muthulakshmi, K., Elumalai, T., Kaliyaperumal, P., Perumal, R. (2024). Adaptive intrusion detection system with DBSCAN to enhance banking cybersecurity. International Journal of Informatics and Communication Technology, 15(1): 247-256. https://doi.org/10.11591/ijict.v15i1.pp247-256
[18] Zimek, A., Schubert, E., Kriegel, H.P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining, 5(5): 363-387. https://doi.org/10.1002/sam.11161
[19] Vaarandi, R., Guerra-Manzanares, A. (2024). Stream clustering guided supervised learning for classifying NIDS alerts. Future Generation Computer Systems, 155: 231-244. https://doi.org/10.1016/j.future.2024.01.032
[20] Alsalama, A.A., Elnagar, A. (2025). Boundary analysis in ensemble clustering for improved document classification. In 2025 IEEE International Conference on New Trends in Computing Science, Amman, Jordan, pp. 257-262. https://doi.org/10.1109/ICTCS65341.2025.10989324
[21] Nji, F.N., Faruque, O., Cham, M., Janeja, V., Wang, J.W. (2024). Hybrid ensemble deep graph temporal clustering for spatiotemporal data. In 2024 IEEE International Conference on Big Data, Washington, DC, USA, pp. 4374-4383. https://doi.org/10.1109/BigData62323.2024.10825871
[22] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy, Funchal, Madeira, Portugal, pp. 108-116. https://doi.org/10.5220/0006639801080116
[23] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A. (2009). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, pp. 1-6. https://doi.org/10.1109/CISDA.2009.5356528
[24] Steinley, D., Brusco, M.J. (2018). A note on the expected value of the Rand index. British Journal of Mathematical and Statistical Psychology, 71: 287-299. https://doi.org/10.1111/bmsp.12116
[25] Thrun, M.C., Ultsch, A. (2020). Clustering benchmark datasets exploiting the fundamental clustering problems. Data in Brief, 30: 105501. https://doi.org/10.1016/j.dib.2020.105501
[26] Radi, K., Moughit, M. (2025). Adaptive clustering approaches for domain name system anomaly detection: Comparative performance analysis. International Journal of Safety and Security Engineering, 15(11): 2333-2342. https://doi.org/10.18280/ijsse.151113
[27] Calinski, T., Harabasz, J. (1974). A dendrite method for cluster analysis. Communications in Statistics, 3(1): 1-27. https://doi.org/10.1080/03610927408827101
[28] Khaoula, R., Imane, M., Mohamed, M. (2024). Improving cyber defense with DNS query clustering analysis. In 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM), Leeds, United Kingdom, pp. 1-6. https://doi.org/10.1109/WINCOM62286.2024.10656538