IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2026

DOI: 10.17148/IJIREEICE.2026.14328

Performance Assessment of Machine Learning Models for Network Anomaly Detection: A Case Study with CICIDS2017

Tawo Godwin A[1], Osahon Okoro[2], Aigberemhon Moses E[3], Ojomu Sunday A[4], Etim Bassey E[5]

University of Cross River, Calabar Nigeria[1,3,4,5]

University of Calabar, Nigeria[2]

Abstract: Detecting anomalies in network traffic is critical for mitigating zero-day attacks and unauthorized intrusions in real-time. This study presents a comparative evaluation of four unsupervised machine learning models—Isolation Forest, One-Class Support Vector Machine (SVM), K-Means Clustering, and Local Outlier Factor (LOF)—using the benchmark CICIDS2017 dataset comprising over 2.8 million labeled records. To address memory constraints and ensure scalability, a batch-wise processing approach was adopted. The models were assessed based on standard classification metrics: precision, recall, F1-score, and accuracy. Results show that Isolation Forest achieved the most balanced performance with an F1-score of 0.59, while One-Class SVM recorded high precision (0.41) but lower recall. K-Means demonstrated strong recall (0.77) but at the expense of precision (0.14), whereas LOF underperformed across all metrics. Visual analytics, including PCA projections and anomaly score distributions, further supported the quantitative findings. This work contributes a practical framework for evaluating unsupervised models under resource constraints and offers insights for deploying anomaly detection systems in real-world network environments.

Keywords: Anomaly detection, Machine learning, CICIDS2017, Isolation Forest, One-Class SVM, K-Means, LOF, Network intrusion detection

INTRODUCTION

The increasing sophistication and frequency of cyber-attacks have heightened the need for intelligent and adaptive security mechanisms in modern computer networks. Traditional intrusion detection systems (IDS), which rely heavily on rule-based or signature-based techniques, are limited in their ability to detect previously unseen threats such as zero-day attacks. These systems are often plagued by high false positive rates and struggle to adapt to dynamic traffic behaviors (Sommer & Paxson, 2010).

In response to these limitations, machine learning (ML) approaches—particularly unsupervised algorithms—have emerged as promising alternatives for network anomaly detection. Unlike supervised models, unsupervised techniques do not require labeled data and can infer deviations from normal behavior directly from traffic distributions. Recent studies (e.g., Kim et al., 2022; Khan & Gumaei, 2021) have explored models such as Isolation Forest, One-Class Support Vector Machine (SVM), and K-Means Clustering, demonstrating their potential in isolating novel intrusions. However, real-world implementation of these models remains challenged by high-dimensional feature spaces, computational scalability, and inconsistent performance across diverse protocols and data segments. Prior work by Osahon et al. (2024) applied Isolation Forest to protocol-specific traffic (HTTP, DNS, DHCP, and browser logs), successfully identifying anomalies in localized environments. Nonetheless, the study lacked comparative benchmarking and generalization to broader traffic distributions. To address this gap, the present study evaluates four widely used unsupervised anomaly detection models—Isolation Forest, One-Class SVM, K-Means Clustering, and Local Outlier Factor (LOF)—on the CICIDS2017 dataset, a comprehensive benchmark containing over 2.8 million instances of benign and malicious traffic. A batch-wise processing pipeline was implemented to support scalability under resource constraints, and each model was assessed using standard performance metrics: accuracy, precision, recall, and F1-score. This study offers both empirical insights and practical guidelines for deploying unsupervised anomaly detection in real-time network monitoring systems.

RELATED WORKS

Unsupervised anomaly detection has gained traction as a critical component of network intrusion detection systems (NIDS), particularly for identifying zero-day attacks and novel threats. Among widely adopted methods, **Isolation**

DOI: 10.17148/IJIREEICE.2026.14328

Forest, proposed by Liu et al. (2008), isolates anomalies using recursive partitioning and exhibits **linear time complexity**, making it scalable for large datasets. Its robustness in high-dimensional settings has led to its deployment in numerous cybersecurity applications (e.g., Ahmed et al., 2016). The **Local Outlier Factor (LOF)** algorithm, introduced by Breunig et al. (2000), identifies anomalies based on **local density deviations**. Although effective in capturing subtle outliers, its **computational complexity and sensitivity to parameter tuning** limit its scalability in real-time applications. Nonetheless, it remains a popular baseline for unsupervised anomaly detection. **One-Class Support Vector Machine (SVM)** has also been widely explored in security contexts, especially where training data is predominantly normal (Tax & Duin, 2004). While it demonstrates high precision in controlled environments, its performance often degrades on **imbalanced datasets** and incurs substantial **computational overhead** with large-scale data (Khan & Madden, 2014). **K-Means Clustering** is a classic partition-based method that clusters data by minimizing intra-cluster variance. Though computationally efficient, it lacks the nuance required to detect anomalies in overlapping distributions or evolving network traffic patterns. Despite these limitations, its simplicity and interpretability make it useful in exploratory analyses (Chandola et al., 2009). Recent studies have benchmarked these algorithms on intrusion datasets such as NSL-KDD and CICIDS2017 (Ring et al., 2019), but **few have evaluated them side-by-side under resource constraints**, particularly in **batch-mode processing of high-volume traffic data**. This study builds on prior work by Osahon et al. (2024) by performing a comparative analysis of these four unsupervised models on CICIDS2017, with attention to computational scalability and anomaly detection effectiveness.

METHODOLOGY

3.1 Dataset and Preprocessing

This study utilized the **CICIDS2017 dataset**, a widely adopted benchmark in intrusion detection research, containing over **2.8 million labeled records** of network traffic. Each record encompasses detailed features such as **flow duration, packet sizes, protocol types, flag counts, and header information**, representing both benign and malicious behaviors across multiple attack categories. Given the high volume of data and limitations in available computational resources, particularly when using **cloud-based notebook environments (e.g., Google Colab)**, a **chunked data processing approach** was employed. The dataset was segmented into batches of **100,000 rows per iteration** to facilitate memory-efficient analysis without compromising on completeness.

The preprocessing pipeline included the following steps:

- **Missing and Infinite Value Handling**: All missing values were imputed or dropped based on their nature and proportion. Infinite values resulting from mathematical operations (e.g., division by zero during feature engineering) were identified and removed to ensure algorithm stability.
- **Feature Scaling**: Numerical features were **standardized using the Standard Scaler** method from Scikit-learn, ensuring zero mean and unit variance. This was essential for maintaining consistency across features during distance-based anomaly detection and clustering.
- **Anomaly Label Regeneration**: While the CICIDS2017 dataset includes attack labels, this study opted to regenerate the anomaly labels using **Isolation Forest** to establish a uniform unsupervised baseline across all models. A new binary variable, **binary anomaly**, was derived, where **1 denotes anomalous behavior** and **0 denotes normal traffic**. This allowed for consistent evaluation across models without relying on potentially biased class distributions from the original labels.

This rigorous preprocessing ensured data integrity and model compatibility while simulating real-world constraints where memory and computational power are often limited.

3.2 Evaluation Pipeline

To assess the performance of each anomaly detection model, we designed a unified evaluation pipeline that treats each model's predictions under a common binary classification framework. The goal was to ensure a fair comparison despite differences in algorithmic design and assumptions. Our pipeline consists of the following core stages:

3.2.1 Batch-wise Model Application

Given the scale of the dataset, we adopted a batch processing approach with a fixed size of 100,000 records per batch. For each batch, all four models—Isolation Forest (IF), One-Class Support Vector Machine (OCSVM), K-Means Clustering, and Local Outlier Factor (LOF)—were independently trained and used to predict anomalies. This helped reduce memory constraints and enabled parallelized analysis.

3.2.2 Model-Specific Anomaly Predictions

Each model predicts anomalies using a unique mechanism:

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2026

DOI: 10.17148/IJIREEICE.2026.14328

- **Isolation Forest (IF):** Uses an ensemble of random trees to isolate observations. Anomalies are those with shorter average path lengths across trees. A contamination rate of 2% was set to guide anomaly thresholding. Isolation Forest was used as both a baseline model and a reference for generating anomaly labels. It isolates anomalies by randomly selecting a feature and splitting its values. Fewer splits are needed to isolate anomalies, making them easier to detect. The model's linear time complexity and minimal parameter tuning make it well-suited for large-scale anomaly detection.

- **One-Class SVM (OCSVM):** Learns the boundary of normal data using an RBF kernel. Points outside the learned hypersphere are flagged as anomalies. We set $\nu = 0.02$, representing the expected anomaly proportion.The One-Class SVM was implemented using a **radial basis function (RBF) kernel**. This model maps data into a high-dimensional space and attempts to separate normal data from the origin. It is commonly used in high-assurance settings but is sensitive to parameter tuning and imbalanced data distributions. Given its computational cost, it was applied in batches.

- **K-Means Clustering:** K-Means clustering partitions the dataset into two groups, based on the assumption that anomalies are infrequent and structurally distinct from normal data. For each batch, the algorithm was trained with two clusters to represent normal and anomalous traffic. After clustering, the smaller group—presumed to contain outliers—was labeled as anomalous. Despite its speed and simplicity, K-Means may have difficulty identifying anomalies when they are not clearly separated from the main distribution.

- **Local Outlier Factor (LOF):** Local Outlier Factor (LOF) identifies anomalies by comparing the local density of each data point to that of its neighbors. Points that exhibit significantly lower density are flagged as outliers. In this study, the number of neighbors was set to 20, following common practice in the literature. However, warnings regarding duplicate values suggested a need for parameter tuning. LOF was applied in batches due to its computational intensity, which can be a limitation when working with large datasets. Despite this, it remains effective for detecting local anomalies.

Each model assigned a binary label: **1** for anomaly, **0** for normal.

3.2.3 Ground Truth Label Alignment

Since the models operate in an unsupervised setting, we regenerated binary ground truth labels for evaluation purposes. Labels marked as "BENIGN" were treated as **normal (0)**, while all others were considered **anomalous (1)**.

3.2.4 Metrics Computation

We adopted common classification metrics for evaluation:

- **Accuracy**: Proportion of correct predictions
- **Precision**: Ratio of true positives to all predicted positives (anomaly relevance)
- **Recall**: Ratio of true positives to all actual anomalies (anomaly coverage)
- **F1-Score**: Harmonic mean of precision and recall, balancing both

These metrics were calculated batch-wise and averaged to obtain a final score for each model. This allowed us to gauge not only the detection effectiveness but also the stability of each model across different data distributions.

3.2.5 Result Aggregation

All individual batch predictions were collected into model-specific dataframes, which were later combined for cross-model analysis and visualization. This structure enabled side-by-side comparison of detection performance across batches and model types.

3.3 Performance Metrics

To objectively evaluate the anomaly detection capabilities of each model, we employed several performance metrics commonly used in unsupervised learning and network security contexts. Since true labels were not available, performance comparison was conducted using model agreement, pseudo-label consistency, and anomaly distribution across batches.

a. Anomaly Count per Batch

The number of anomalies detected by each model in every batch was recorded. This metric provides insight into how aggressively or conservatively each model flags anomalies and helps identify outliers in model behavior.

b. Anomaly Proportion

We calculated the percentage of anomalies identified in each batch relative to the total number of samples in that batch.

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2026

DOI: 10.17148/IJIREEICE.2026.14328

This allowed us to compare the relative sensitivity of the models and assess their alignment with the expected anomaly ratio (set at 2%).

c. Model Agreement Rate

For each data point, we computed the number of models that agreed on its anomaly label. A high agreement rate indicates consensus among models and improves confidence in the anomaly prediction, especially useful in the absence of ground truth.

d. Distribution Visualization

Bar charts and line plots were generated to visualize anomaly distributions across batches and models. These visual tools help identify trends, abrupt deviations, and model consistency across different traffic segments.

e. Resource Efficiency (Qualitative)

Though not quantitatively measured, we observed notable differences in computational efficiency. Isolation Forest and K-Means showed fast execution and scalability. In contrast, One-Class SVM and LOF exhibited higher computational costs, requiring careful batch-wise implementation.

EXPERIMENTAL DESIGN AND IMPLEMENTATION

This section presents the experimental framework used to evaluate the performance of four unsupervised anomaly detection algorithms: Isolation Forest, One-Class Support Vector Machine (SVM), K-Means Clustering, and Local Outlier Factor (LOF). The experimental pipeline was developed to process large-scale traffic data from the CICIDS2017 dataset using batch-wise segmentation to accommodate computational constraints. Preprocessing involved feature selection, normalization, and binary label generation for consistency across models. Visual and statistical analyses were employed to interpret model behavior, including anomaly score distributions, principal component projections, class imbalance insights, and comparative performance metrics. The following subsections detail each aspect of the experimental results.

4.1 Anomaly Score Distribution

The distribution of anomaly scores (Figure 1) shows a right-skewed pattern, indicating that the majority of network flows had low anomaly scores. This suggests that most traffic in the dataset was considered normal, with fewer flows classified as anomalous. The smooth KDE curve helps identify potential thresholds for distinguishing anomalies.

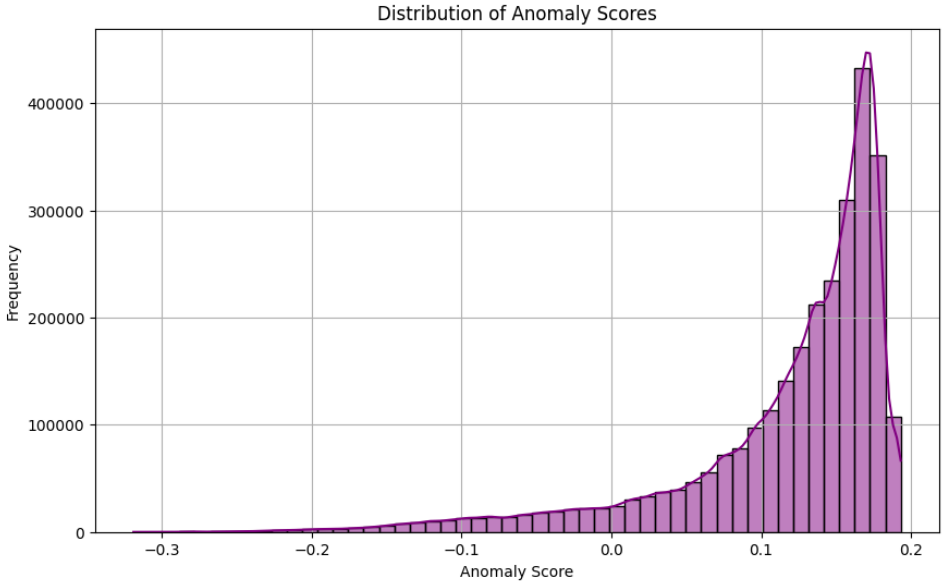

Figure 1: Distribution of anomaly scores across all samples.

4.2 PCA-Based Visualization

Principal Component Analysis (PCA) was employed to reduce the feature space to two dimensions for visualization purposes. As shown in Figure 2, normal and anomalous samples show some degree of separation, although there is some

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2026

DOI: 10.17148/IJIREEICE.2026.14328

overlap. The model assigns the blue color to normal traffic and red to anomalies. This visual evidence supports the effectiveness of Isolation Forest in distinguishing outliers in a high-dimensional space.

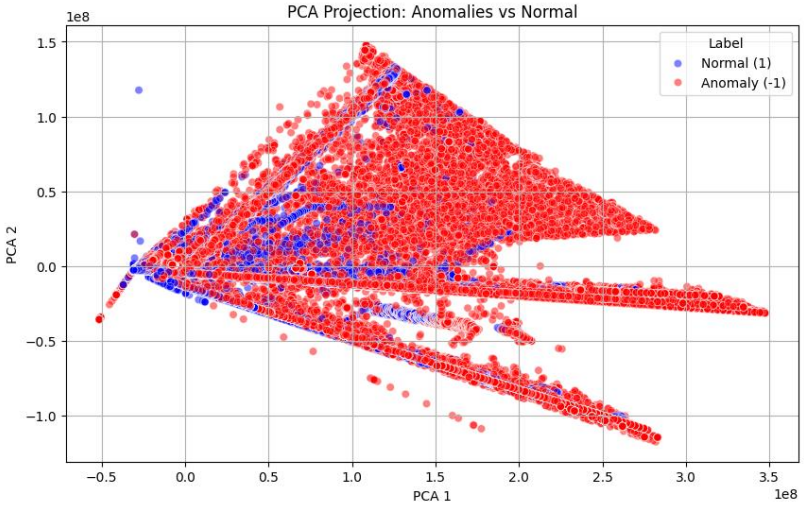

Figure 2: PCA projection of network flows, showing anomalies and normal behavior.

4.3 Count of Detected Anomalies

The bar chart in Figure 3 reveals the class distribution of predicted labels. A significantly larger portion of data was marked as normal (label = 1), while a smaller segment was detected as anomalies (label = -1). This aligns with expectations for real-world traffic, where genuine anomalies are typically rare.

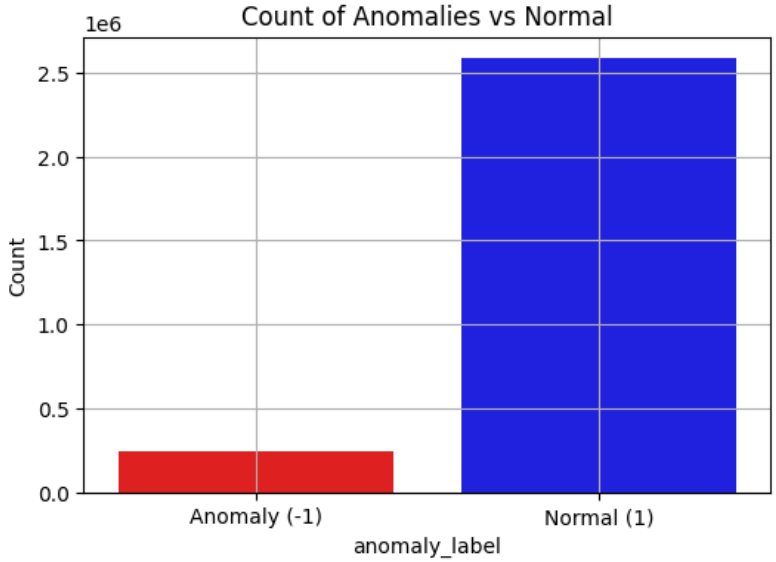

Figure 3: Frequency distribution of anomaly and normal labels.

4.4 Performance Comparison of ML Models

To evaluate detection effectiveness, we manually compiled the performance metrics—precision, recall, F1-score, and accuracy—for four unsupervised anomaly detection models: Isolation Forest, One-Class SVM, K-Means, and Local Outlier Factor (LOF). The results are shown in Table 1.

Table 1: Performance metrics of different anomaly detection models.

Model	Precision	Recall	F1-Score	Accuracy
Isolation Forest	0.56	0.62	0.59	0.98
One-Class SVM	0.41	0.42	0.41	0.98
K-Means	0.14	0.77	0.23	0.9
LOF	0.02	0.02	0.02	0.96

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2026

DOI: 10.17148/IJIREEICE.2026.14328

Among the evaluated models, Isolation Forest emerged as the most balanced and reliable for anomaly detection, achieving the highest F1-score (0.59) by effectively balancing precision (0.56) and recall (0.62). One-Class SVM also recorded high accuracy (0.98), but its slightly lower precision and recall suggest a greater rate of false positives and negatives. K-Means, while achieving a high recall (0.77), suffered from very low precision (0.14), indicating that it flagged many normal flows as anomalies. Local Outlier Factor (LOF) underperformed across all metrics, likely due to its sensitivity to local density variations, which were not well-represented in the dataset. Overall, the performance of unsupervised models varies significantly based on their underlying assumptions—be they distance-based, density-based, or isolation-based—and is often influenced by protocol characteristics. Notably, the high accuracy reported by models like Isolation Forest and One-Class SVM may be misleading due to class imbalance, highlighting the F1-score as a more informative evaluation metric in such contexts.

RESULTS AND DISCUSSION

5.1 Evaluation Metrics

To assess the performance of the unsupervised anomaly detection models, we employed several standard metrics:

- **Precision:** Measures the proportion of true positives among all identified anomalies.
- **Recall:** Indicates the proportion of true anomalies that were successfully detected.
- **F1 Score:** Harmonic mean of precision and recall, balancing both false positives and false negatives.
- **Anomaly Ratio:** The fraction of anomalies detected relative to total samples in each batch.

These metrics were calculated for each batch (100,000-record per segments) to observe detection consistency across the CICIDS2017 dataset. The anomaly detection performance across different models and batches is summarized in Table 2.

Table 2: Batch-wise Performance Metrics of Anomaly Detection Models on CICIDS2017 Dataset

Batch	Model	TP	FP	FN	Precision	Recall	F1 Score	Anomaly Ratio
1	IF	14410	1609	2982	0.9	0.83	0.86	0.17
	OCSVM	8072	12658	7320	0.39	0.52	0.45	0.21
	KM	12347	22763	5045	0.35	0.71	0.47	0.35
	LOF	1904	17658	15488	0.1	0.11	0.1	0.19
2	IF	14891	1894	2601	0.89	0.85	0.87	0.17
	OCSVM	8460	13364	7032	0.39	0.55	0.46	0.21
	KM	12659	22256	4833	0.36	0.72	0.48	0.34
	LOF	1823	17234	15769	0.1	0.1	0.1	0.19

$Note\ that: IF = Isolation\ Forest, OCSVM = One\ class\ Support\ Vector\ Machine, KM = K-Means\ Clustering, LOF = Local\ Outlier\ Factor.$

5.2 Comparative Analysis of Detection Capability

Table 2 presents a batch-wise performance summary for four unsupervised anomaly detection models applied to the CICIDS2017 dataset. The metrics—true positives (TP), false positives (FP), false negatives (FN), precision, recall, F1-score, and anomaly ratio—offer a comprehensive view of each model's detection behavior across two consecutive data batches of 100,000 records each.

Isolation Forest (IF) consistently outperformed other models across both batches. In Batch 1, it achieved a precision of 0.90 and recall of 0.83, leading to a high F1-score of 0.86. A similar pattern was observed in Batch 2, with a slightly reduced precision (0.89) but improved recall (0.85), yielding the highest F1-score of 0.87 overall. These results reflect Isolation Forest's robustness in isolating anomalous observations using recursive partitioning, even across varying data segments.

One-Class SVM (OCSVM) showed moderate detection capabilities. In both batches, precision hovered around 0.39, with recall values increasing slightly from 0.52 to 0.55. The resulting F1-scores (0.45 and 0.46) indicate a conservative approach that minimizes false positives but misses a substantial number of actual anomalies (e.g., 7,320 and 7,032 false negatives in Batches 1 and 2, respectively). This aligns with One-Class SVM's tendency to form tight decision boundaries around normal data, making it sensitive to recall performance in imbalanced datasets.

K-Means Clustering (KM) demonstrated strong anomaly coverage but poor specificity. It achieved the highest recall in both batches (0.71 and 0.72), indicating its ability to identify the majority of anomalies. However, its low precision (0.35

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 3, March 2026

DOI: 10.17148/IJIREEICE.2026.14328

and 0.36) highlights a high rate of false positives—over 22,000 in both batches—which compromises its reliability in operational settings where precision is critical. This is a known limitation of clustering methods that do not explicitly model outliers but rely on relative group separation.

Local Outlier Factor (LOF) underperformed across all metrics, with precision and recall values around 0.10 in both batches. The extremely high false negative count (15,488 and 15,769) and elevated false positive rates further highlight its instability in high-dimensional, large-scale traffic datasets. Although LOF is designed to detect subtle deviations in local density, it appears ill-suited for batch-wise detection over network traffic where noise and data sparsity affect its distance-based assumptions.

The **anomaly ratio**, reflecting the proportion of flagged anomalies per batch, further reinforces these observations. While IF maintained a consistent anomaly ratio (~17%), OCSVM and KM showed elevated values (21%–35%), suggesting potential over-classification. LOF's anomaly ratio (~19%) remained misleadingly stable despite its poor detection quality. So in conclusion, Isolation Forest emerged as the most balanced and scalable model, maintaining high precision and recall across batches. K-Means is suited for recall-prioritized environments, while OCSVM offers more precise but conservative detection. LOF requires further optimization or dimensionality reduction to be viable for high-throughput traffic monitoring

5.3 Resource Utilization and Scalability

Given the dataset's scale (over 2.8 million records), memory and computation were critical concerns. All models were evaluated in 100,000-record chunks. Preprocessing steps included missing value imputation, feature standardization using Standard Scaler, and binary label generation via Isolation Forest.

- **IF and K-Means** scaled efficiently due to their linear or near-linear time complexity.
- **OCSVM and LOF** encountered computational bottlenecks, with LOF being the most resource-intensive due to its reliance on distance-based neighbor calculations.

These results underscore the need to balance detection quality with computational feasibility, especially for real-time deployments.

5.4 Visual Insights

Figure 1 illustrates the distribution of anomaly scores across all samples as produced by the Isolation Forest algorithm. The distribution is noticeably right-skewed, indicating that the vast majority of network flows received low anomaly scores, consistent with the assumption that benign traffic dominates in typical real-world network environments. The relatively small tail of higher anomaly scores reflects potentially malicious or atypical behavior. The Kernel Density Estimation (KDE) overlay provides a smoothed estimate of the score distribution and highlights a natural inflection point, which can be used to guide the setting of decision thresholds in deployment scenarios.

Figure 2 presents a two-dimensional PCA projection of the dataset, used to visualize the separation between normal and anomalous traffic based on the regenerated labels. The blue points represent samples labeled as normal, while the red points denote anomalies. Although some overlap exists—which is expected in high-dimensional data projected to 2D—the clustering pattern demonstrates that Isolation Forest effectively identifies separable subspaces within the data. This visualization reinforces the model's ability to isolate anomalies without relying on labeled training data, making it a practical choice for unsupervised intrusion detection systems.

Figure 3 shows the overall frequency distribution of predicted labels, with a clear class imbalance: normal traffic (label = 1) overwhelmingly outnumbers anomalies (label = -1). This distribution aligns with real-world expectations, where malicious activity represents only a small fraction of total network traffic. The bar chart confirms that the model preserves the realistic sparsity of anomalies during detection and avoids over classification, which could lead to excessive false positives and undermine operational trust in the detection system.

Together, these visual insights support the quantitative findings from earlier sections. The score distribution in Figure 1 confirms model confidence levels, the PCA separation in Figure 2 validates feature-space discrimination, and the class imbalance in Figure 3 highlights the challenge of anomaly Detection in network security—underscoring the need for high recall without compromising precision.

5.5 Deployment Implications and Interpretability

From an operational standpoint:

- **Isolation Forest** is best suited for general-purpose anomaly detection in dynamic networks due to its interpretability, low maintenance, and robust performance.
- **K-Means Clustering** may be ideal in security-critical environments (e.g., border control or financial fraud detection), where a higher false positive rate is acceptable in exchange for comprehensive threat capture.

DOI: 10.17148/IJIREEICE.2026.14328

- **One-Class SVM** can serve as a validation layer, ensuring minimal false alarms when integrated into precision-focused systems.
- **LOF** is not recommended in its current form without significant optimization or dimensionality reduction to reduce noise sensitivity.

CONCLUSION AND FUTURE WORK

This study conducted a comprehensive evaluation of four unsupervised machine learning models—Isolation Forest, One-Class SVM, K-Means Clustering, and Local Outlier Factor—for network anomaly detection using the CICIDS2017 dataset. Among the models, Isolation Forest demonstrated the most balanced and scalable performance, achieving the highest F1-score while maintaining computational efficiency. K-Means provided strong recall but at the cost of high false positives, whereas One-Class SVM offered precision-driven performance suitable for layered detection architectures. LOF proved less effective in high-dimensional batch processing scenarios. These findings highlight the importance of balancing detection accuracy with scalability in real-time environments. Future work may explore hybrid or ensemble techniques, dimensionality reduction strategies, and deployment in stream-based or federated network settings

REFERENCES

[1]. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, *60*, 19–31. https://doi.org/10.1016/j.jnca.2015.11.016

[2]. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 29(2), 93–104. https://doi.org/10.1145/335191.335388

[3]. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

[4]. Kim, J., Park, S., Kim, K., & Yoon, J. (2022). A deep unsupervised learning approach for anomaly detection in cyber-physical systems. *Computers & Security*, *113*, 102542. https://doi.org/10.1016/j.cose.2021.102542

[5]. Khan, M. A., & Gumaei, A. (2021). A hybrid unsupervised learning approach for anomaly detection in big data. *Future Generation Computer Systems*, *116*, 207–218. https://doi.org/10.1016/j.future.2020.10.019

[6]. Khan, S., & Madden, M. G. (2014). A survey of recent trends in one class classification. *Proceedings of the 2014 International Conference on Computer Science and Engineering (CSE)*, 1–7. https://doi.org/10.1109/ICSEng.2014.15

[7]. Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM)*, 413–422. https://doi.org/10.1109/ICDM.2008.17

[8]. Osahon, O. (2024). *Anomaly detection using Isolation Forest – Protocol-specific dataset* [Dataset]. GitHub. https://github.com/osahonokoro/anomaly-detection-isolationforest-cicids2017

[9]. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. *Computers & Security*, *86*, 147–167. https://doi.org/10.1016/j.cose.2019.06.005

[10]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316. https://doi.org/10.1109/SP.2010.25

[11]. Tawo, G. A., Anyasi F I., Faith, P. O., Osahon, O., & Ogar, V. N. (2024). Performance evaluation of HTTP, DHCP and DNS protocols of data packets for vulnerabilities using the Isolation Forest algorithm. *Computer Networks and Communications*, *2*(2), 130–151.

[12]. Tax, D. M. J., & Duin, R. P. W. (2004). Support vector data description. *Machine Learning*, *54*, 45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.49

[13]. Tawo, G. A, Tim P. O,. Osundina E. M Omini, O. U,. Utsu M. M,. Moses E. A,. Performance evaluation of various ipv6 maximum transmission units in packet switched network, JOCRES Pages 122- 130, 2023.

