FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 10, October, 2025, pp 348 – 354
DOI: https://doi.org/10.33003/fjs-2025-0910-3752

MACHINE LEARNING DRIVEN NETWORK TRAFFIC ANALYSIS FOR CYBERSECURITY: A COMPARATIVE STUDY OF SUPERVISED AND UNSUPERVISED LEARNING APPROACHES

***[1]Abdulrahman Tunde Alabelewe, [2]Nasir Shinkafi, [1]Adeyinka Samson, [1]Sulaiman Abu Usman, [1]Maryam Masari, [1]Muhammad Auwal Bello and [1]Joshua Yakubu Anche**

[1]Department of Cyber Security, Airforce Institute of Technology, Kaduna, Nigeria.
[2]Galaxy Backbone Limited, FCT-Abuja, Nigeria.

Corresponding author's email: abdulrahman.alabelewe@gmail.com Phone: +2347037363957

ABSTRACT

This study looks at how well machine learning (ML) methods work in cybersecurity, focusing on their ability to tell apart malicious and normal network traffic. Using the CICIDS2017 dataset, we compare supervised learning models like Random Forest and Support Vector Machines with unsupervised techniques such as K-means clustering and Isolation Forest. We evaluate their performance using multiple metrics, including accuracy, precision, recall, F1-score, and cluster validity indices, to find the most effective approach for spotting anomalies in network data. The results show that Random Forest delivers the best overall performance, achieving over 99.4% accuracy with very few false negatives. Meanwhile, unsupervised methods excel at detecting new, previously unseen patterns without needing labeled data. In particular, the Isolation Forest model achieves a recall of 93%, making it highly effective at identifying anomalies. K-means clustering also performs well, clearly separating traffic patterns with strong Silhouette scores (0.8622) and favorable Davies-Bouldin indices (0.6063).

Keywords: Machine Learning, Network Traffic Analysis, Anomaly Detection, Supervised Learning, Unsupervised Learning

INTRODUCTION

The rapid growth of digital technologies has brought unprecedented connectivity, but it has also made modern networks increasingly vulnerable to cyber-attacks. From financial institutions and healthcare systems to critical infrastructure and government agencies, organizations across all sectors face a wide range of evolving cyber threats. Traditional rule-based security systems, while useful for known attack signatures, often struggle to keep up with the dynamic nature of modern cyber threats, which frequently involve novel techniques that bypass static detection rules (Alloghani et al., 2020)

In response to these challenges, machine learning (ML), a branch of artificial intelligence (AI), has gained significant attention in the field of cybersecurity such as in intrusion detection and cyberbullying (Maikano, 2024; Sakhai & Wielgosz, 2021). Unlike rule-based systems, ML algorithms can learn from data, identify complex patterns, and adapt to new types of attacks as they emerge. This flexibility makes ML particularly well-suited for detecting anomalies and previously unseen attack behaviors that conventional systems may miss.

While many studies have investigated the application of ML to intrusion detection, most focus on either supervised or unsupervised methods in isolation, or lack direct comparisons across different algorithm types within the same experimental setup (Alloghani et al., 2020). As cybersecurity threats become more sophisticated, understanding how different ML models perform under comparable conditions is increasingly important for building effective and adaptable defense systems.

The primary aim of this study is to provide a comprehensive comparative evaluation of both supervised and unsupervised ML algorithms for network intrusion detection. Specifically, we assess the performance of Random Forest and Support Vector Machines as supervised models, alongside K-Means clustering and Isolation Forest as unsupervised techniques. The CICIDS2017 benchmark dataset, which closely simulates real-world network environments and includes a diverse set of contemporary attack scenarios, serves as the basis for our analysis.

Literature Review

Machine Learning in Cybersecurity

The increasing complexity of cyber threats has driven significant interest in artificial intelligence (AI)-based solutions, particularly machine learning (ML), as an alternative to traditional rule-based systems. AI encompasses a broad set of computational technologies designed to replicate human cognitive functions such as learning, reasoning, and decision-making (Korteling et al., 2021). Within this broader field, ML has gained prominence due to its ability to autonomously extract meaningful patterns from large datasets and continuously improve performance without explicit programming (Kim & Park, 2021)

This adaptability makes ML particularly well-suited for cybersecurity, where attack methods evolve rapidly and may exhibit subtle patterns that are difficult to capture using static detection rules. Applications of ML in this domain include intrusion detection, malware classification, anomaly detection, and behavioral analysis.

Supervised Learning in Cybersecurity

Supervised learning relies on labeled datasets, where each data point is paired with a known outcome. This allows models to learn specific mappings between input features and target labels, making these algorithms highly effective for well-defined classification problems (Vu et al., 2020). In cybersecurity, supervised models are often used to classify network traffic as either benign or malicious, based on historical attack data (Alloghani et al., 2020).

Among supervised algorithms, decision tree-based models such as Random Forest have proven especially effective. Random Forest operates by combining multiple decision trees

to form an ensemble model that improves prediction accuracy while minimizing overfitting—an advantage when dealing with high-dimensional network traffic data (Musleh et al., 2023). Similarly, Support Vector Machines (SVMs) create optimal decision boundaries to separate data classes and are particularly effective when the data is well-separated in the feature space (Bin Sarhan & Altwaijry, 2023)

Other supervised algorithms have also found application in cybersecurity. Logistic Regression offers a probabilistic framework well-suited to binary classification tasks, often used when computational resources are limited (Jony & Arnob, 2024). Naïve Bayes classifiers, valued for their scalability and robustness to missing data, have been widely adopted in applications such as spam filtering, malware detection, and anomaly detection (Chen et al., 2020).

While these supervised approaches generally deliver strong classification accuracy when ample labeled data is available, their effectiveness is often limited when encountering novel or previously unseen attack patterns.

Unsupervised Learning in Cybersecurity

Unsupervised learning represents a crucial machine learning method that analyzes unlabeled data to identify hidden patterns and structures without predefined classifications (Alloghani et al., 2020). Unlike supervised methods, these algorithms autonomously discover hidden relationships within datasets, making them particularly valuable in cybersecurity where labeled data remains scarce and new threats continuously emerge (Bohara et al., 2017).

Several key algorithms demonstrate significant potential in network security applications. Hierarchical clustering builds nested structures that capture complex behavioral patterns, including privilege escalation and geographically inconsistent access attempts (Murtagh & Contreras, 2017). Meanwhile, Isolation Forest excels at detecting anomalies in high-dimensional data by recursively partitioning points to isolate outliers, proving superior to traditional signature-based methods for identifying novel threats like covert malware communications (Tao et al., 2018). Recent progress in deep unsupervised learning have further enhanced this field's capabilities. Alom & Taha, (2017) demonstrated that autoencoders and restricted Boltzmann machines can achieve detection accuracies exceeding 91% on benchmark datasets like KDD-99, highlighting unsupervised learning's unique ability to process large-scale network data and adapt to evolving threat landscapes without extensive labeled training requirements.

MATERIALS AND METHODS

Research Framework

This study adopted a structured, four-phase methodological framework designed to systematically evaluate supervised and unsupervised machine learning algorithms for network intrusion detection. The overall research design is illustrated in Figure 1.

The framework consists of (1) data preparation, (2) data preprocessing, (3) model implementation, and (4) performance evaluation.

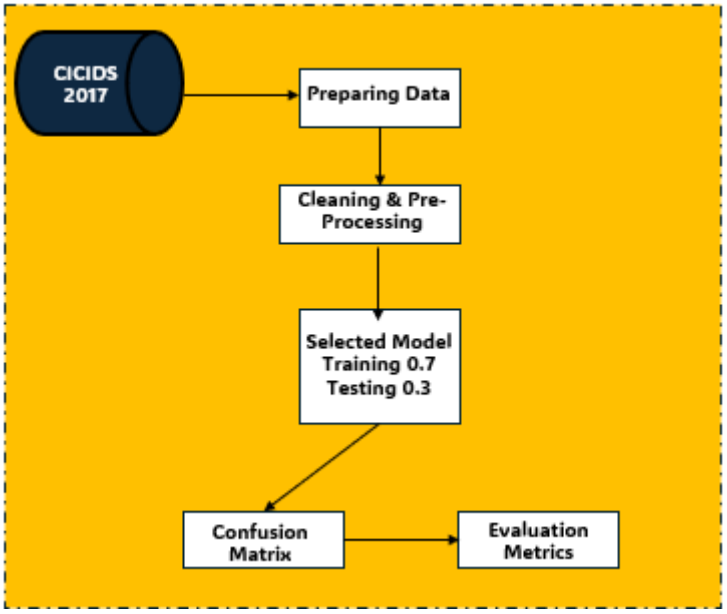

Figure 1: Research Methodology Framework

Tools and Development Environment

All implementations were performed using Python, a widely adopted programming language in data science and machine learning. Key libraries included:

i. ***scikit-learn*** for algorithm implementation and model evaluation (Sarkar et al., 2018).
ii. ***Pandas*** for data cleaning and manipulation.
iii. ***NumPy*** for efficient numerical computation.
iv. ***Matplotlib*** for data visualization.
v. ***StandardScaler*** (from scikit-learn) for feature normalization.

Development was conducted in Jupyter Notebook, chosen for its interactive coding environment, reproducibility, and integration of code, results, and documentation (Toomey, 2016).

Dataset Selection

The CICIDS2017 dataset was selected due to its realistic simulation of enterprise network traffic and inclusion of diverse attack types, such as DDoS, DoS, brute-force, infiltration, and botnet activities (Okey et al., 2023). Its detailed labeling of attack types versus normal traffic made it highly suitable for both supervised and unsupervised learning

models (Maseer et al., 2021). The dataset provides a balanced representation of malicious and benign network behavior, allowing for robust evaluation under realistic conditions.

Data Preparation and Preprocessing

In the Data Preparation phase, raw network traffic from multiple days and attack scenarios was consolidated into a unified dataset.

During Preprocessing, several data quality issues were addressed, including:

i. Handling missing values and structural inconsistencies.
ii. Removing outliers to improve model stability.
iii. Normalizing numerical features using StandardScaler to ensure comparable scales across input features, especially important for algorithms sensitive to feature magnitudes.

The cleaned dataset was partitioned into 70% for model training and 30% for testing, ensuring unbiased evaluation.

Model Development

Both supervised and unsupervised learning models were implemented. For the Supervised Models, Random Forest (RF) was selected for its robustness to high-dimensional data and ability to capture complex feature interactions through ensemble learning (Musleh et al., 2023). Support Vector Machines (SVM) was also used because of its effectiveness in creating optimal class separation boundaries, particularly in binary classification contexts (Bin Sarhan & Altwaijry, 2023). To ensure reliable performance estimates, both supervised models were evaluated using 5-fold cross-validation.

For the unsupervised learning phase, K-means clustering and Isolation Forest were implemented. K-means was employed to group the network traffic into two primary clusters representing benign and malicious behaviors, allowing the detection of distinct behavioral patterns without prior labels. The quality of these clusters was assessed using the Silhouette Coefficient and Davies-Bouldin Index, which measure intra-cluster cohesion and inter-cluster separation. Isolation Forest was used specifically for anomaly detection, leveraging its strength in isolating outliers within high-dimensional feature spaces without requiring labeled training data.

Performance Evaluation

Performance was evaluated using appropriate metrics for each learning algorithm. For supervised models, standard classification metrics were used, including accuracy, precision, recall, and F1-score, providing a comprehensive assessment of the models' ability to correctly classify network traffic. For unsupervised models, clustering performance was measured using validity indices that reflect how well the algorithms grouped similar patterns and distinguished anomalies.

To ensure computational efficiency, Principal Component Analysis (PCA) was applied when necessary for dimensionality reduction, allowing the models to process complex datasets more effectively without compromising predictive accuracy. All experiments were conducted on a system equipped with an Intel Core i7 processor, 8GB RAM, and running Windows 11 Pro, which provided sufficient computational resources for the study's machine learning workloads.

In summary, the methodological framework applied in this study allowed for a systematic, consistent, and practical evaluation of both supervised and unsupervised machine learning techniques. The design ensured that each model was fairly tested under controlled yet realistic conditions, providing reliable insights into their capabilities for detecting malicious network activity.

RESULTS AND ANALYSIS

Random Forest Performance Evaluation

Evaluation Metrics

The Random Forest model demonstrated consistently strong performance across all evaluation metrics and cross-validation folds. Both training and validation accuracy consistently exceeded 0.98, indicating excellent agreement between predictions and actual traffic classifications as shown in figure 2 below

	fit_time	score_time	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1
0	47.164009	0.099621	0.9946	1.0	0.985670	1.0	0.986680	1.0	0.986175	1.0
1	36.726436	0.109349	0.9950	1.0	0.990712	1.0	0.983607	1.0	0.987147	1.0
2	41.840580	0.084273	0.9954	1.0	0.991744	1.0	0.984631	1.0	0.988175	1.0
3	38.715935	0.115775	0.9966	1.0	0.996891	1.0	0.985656	1.0	0.991242	1.0
4	43.928144	0.159086	0.9952	1.0	0.994808	1.0	0.980553	1.0	0.987629	1.0

Figure 2: Random Forest Evaluation

Accuracy measurements ranging from 0.9946 to 0.9968 indicate exceptional agreement between predicted and actual classifications. Precision values consistently above 0.98 confirm the model's reliability in positive case identification, while recall scores demonstrate comprehensive detection of true positive instances. The harmonic mean represented by F1-scores, also maintaining values above 0.98, further validates the model's balanced performance in both precision and recall domains. Table 1 below summarizes the evaluation for the random forest algorithm.

Table 1: Random Forest Performance Analysis

Metric	Performance	Key Observations
Accuracy	0.9946 - 0.9968 across all folds	Consistently high accuracy indicating excellent classification capability
Precision	> 0.98 across all folds	High reliability of positive predictions, indicating minimal false positives
Recall	> 0.98 across all folds	Excellent capture of attack instances, with very few missed attacks
F1-Score	> 0.98 across all folds	Well-balanced precision and recall, confirming overall effectiveness

Random Forest Confusion Matrix Analysis

The confusion matrix analysis as shown in Figure 3 provided nuanced insights into the Random Forest classifier's performance that extended beyond aggregate accuracy metrics. The model demonstrated exceptional discriminative capability, correctly identifying 6,035 normal traffic instances (True Negatives) and 1,412 attack cases (True Positives), indicating strong pattern recognition for both classification categories.

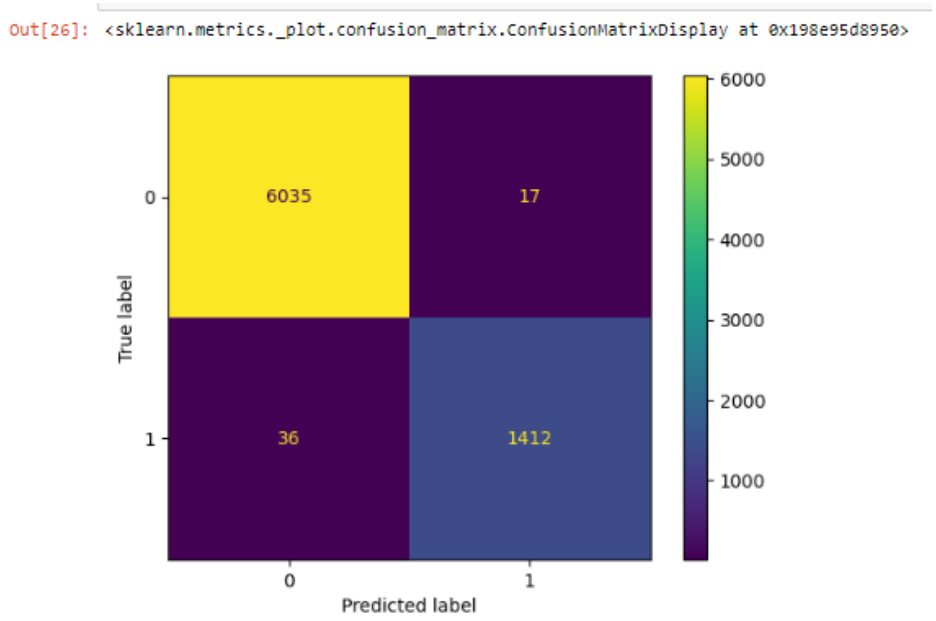

Figure 3: Random Forest Confusion Matrix

While the observed False Positive rate of 36 instances represents a relatively small proportion (0.5% of normal traffic), these misclassifications may warrant consideration in operational environments where false alarms could impact system usability. More critically, the minimal False Negative count (17 instances) suggests robust security coverage, with fewer than 1.2% of attacks going undetected - a crucial factor for mission-critical systems. The overall performance metrics, with correct classifications exceeding 99% of evaluated cases, position this model as highly effective for production deployment.

Support Vector Machine Evaluation

The Support Vector Machine (SVM) model demonstrates strong performance in classifying network traffic, as evidenced by comprehensive cross-validation results. Across all five validation folds, the model maintains consistently high accuracy scores above 0.95 as shown in figure 4, indicating excellent agreement between predicted classifications and actual labels. This robust performance suggests the SVM effectively learns the distinguishing patterns between attack and normal traffic within the CICIDS2017 dataset.

Out[30]:

	fit_time	score_time	test_accuracy	train_accuracy	test_precision	train_precision	test_recall	train_recall	test_f1	train_f1
0	7.253026	2.056682	0.9560	0.96075	0.897059	0.920939	0.875000	0.874008	0.885892	0.896860
1	5.000108	2.079912	0.9598	0.95995	0.909187	0.919233	0.882172	0.871447	0.895476	0.894702
2	6.584984	2.897108	0.9588	0.96055	0.914871	0.918143	0.869877	0.876056	0.891807	0.896606
3	6.791842	1.897639	0.9638	0.95925	0.935378	0.914876	0.875000	0.872471	0.904182	0.893171
4	4.256503	1.955848	0.9570	0.96010	0.924276	0.914797	0.849539	0.877305	0.885333	0.895659

Figure 4: SVM Result Evaluation

A deeper analysis of classification metrics reveals important insights into the model's behavior. Precision values ranging from 0.87 to 0.93 across folds indicate the model correctly identifies most attack instances, though the variation suggests some susceptibility to false positives in certain data segments. The recall metrics, consistently exceeding 0.86, demonstrate the model's ability to capture the majority of actual attack cases. These findings are further supported by F1-scores between 0.87 and 0.90, which reflect a balanced performance between precision and recall. The evaluation metric for SVM is summarized in table 2 below.

Table 2: Support Vector Machine Performance Analysis

Metric	Performance	Key Observations
Accuracy	> 0.95 across all folds	Good overall classification performance
Precision	0.87 - 0.93 across folds	Generally reliable positive predictions with some variation
Recall	> 0.86 across all folds	Good identification of attack instances with room for improvement
F1-Score	0.87 - 0.90 across folds	Solid balance between precision and recall

SVM Confusion Matrix Interpretation

The confusion matrix analysis as shown in Figure 5 reveals critical insights into the SVM model's classification behavior on the CICIDS 2017 dataset. The model demonstrates particularly strong performance in identifying legitimate network activity, with 5,929 true negatives correctly classified as normal traffic. This represents a 98.8% success rate for benign traffic identification, suggesting the model effectively learns the characteristics of typical network behavior.

From a security perspective, the 190 false negatives (2.5% of attack instances) warrant careful consideration, as these represent undetected threats that could compromise system integrity. While the model correctly identifies 1,258 true positives (86.9% of attack cases), the false negative rate suggests potential limitations in detecting certain attack patterns. The 123 false positives (2.0% of normal traffic) indicate occasional misclassification of benign activity as malicious, which could impact operational efficiency through unnecessary alerts.

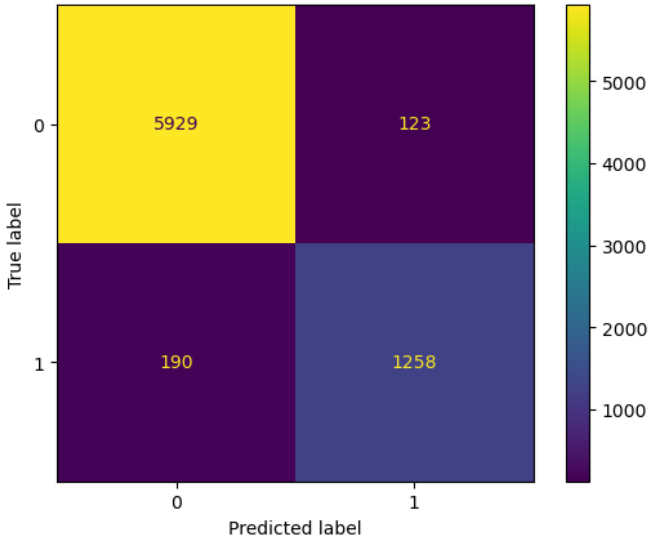

Figure 5: Confusion Matrix for SVM

When compared to the previously evaluated Random Forest model, the SVM shows a higher false negative rate (190 vs 17 instances), suggesting reduced sensitivity to certain attack signatures. This performance difference highlights potential areas for model refinement, including Kernel function optimization to better capture complex attack patterns, Feature space enhancement to improve discriminative capability, Class weight adjustment to prioritize attack detection sensitivity and Decision boundary calibration for operational requirements

The overall classification accuracy of 95.8% (7,187 correct classifications out of 7,500) demonstrates the model's fundamental effectiveness, while the observed error patterns provide clear direction for targeted improvements in security-sensitive applications.

Isolation Forest Evaluation

Isolation Forest excelled in anomaly detection with remarkable recall performance (93.0%), successfully capturing most actual anomalies in the dataset. The precision of 82.8% indicates reasonable accuracy with some false alarms, resulting in an F1-score of 0.876. This recall-prioritized performance profile aligns well with security applications where comprehensive threat identification outweighs false alarm concerns. Table 3 below summarizes the evaluation metric of the Isolation Forest performance analysis.

Table 3: Isolation Forest Performance Analysis

Metric	Performance	Key Observations
Precision	0.828	Good but not excellent; 82.8% of anomaly predictions were correct
Recall	0.930	Excellent capture of actual anomalies; 93% of all anomalies detected
F1-Score	0.876	Good balance between precision and recall

This evaluation demonstrates that while the Isolation Forest may benefit from refinement to reduce false positives, its exceptional anomaly detection capability makes it a compelling choice for security monitoring systems where comprehensive threat identification is critical. The model's architecture appears fundamentally sound for network security applications, with tuning opportunities available to adapt it to specific operational contexts.

K-Means Clustering Evaluation

K-Means demonstrated exceptional unsupervised performance through rigorous validation metrics: Silhouette score of 0.8622 (near-optimal cluster separation), Calinski-Harabasz index of 48,526 (excellent between-cluster variance), and Davies-Bouldin index of 0.6063 (strong cluster distinctiveness). These convergent metrics confirm effective separation of normal and malicious traffic patterns without labeled data.

Discussion

The study reveals individual algorithm strengths aligned with specific operational requirements. Random Forest emerged as the superior supervised approach, achieving the highest accuracy (>99.4%) with remarkable attack detection capability (98.8% recall) and minimal false alarms. Its consistent performance across validation folds demonstrates

robust generalization, making it ideal for environments with well-labeled datasets and known threat signatures.

Among unsupervised methods, both algorithms showed complementary capabilities. K-Means excelled at traffic pattern separation with computational efficiency, making it suitable for baseline monitoring in resource-constrained environments. Isolation Forest's high recall (93%) positions it as valuable for detecting novel or evolving threats that supervised methods might miss, despite generating more false positives.

The comparative analysis emphasizes critical trade-offs between detection accuracy, computational efficiency, and adaptability. Supervised methods excel with labeled data and known attack patterns, while unsupervised approaches provide flexibility for emerging threats without extensive training requirements. Future research should explore hybrid architectures that combine Random Forest for primary threat detection with Isolation Forest for novel anomaly identification, capitalizing on their complementary strengths to achieve comprehensive network security coverage.

CONCLUSION

This comprehensive evaluation of machine learning approaches for cybersecurity using the CICIDS2017 dataset reveals distinct performance characteristics across supervised and unsupervised methods, with Random Forest achieving exceptional classification accuracy (>99.4%) and minimal false negatives (17 out of 1,429 attacks), while Support Vector Machines maintained respectable performance (>95%) but with higher false negative rates (190 instances). Among unsupervised methods, Isolation Forest demonstrated strong anomaly detection capabilities with 93% recall, and K-means clustering showed effective traffic pattern separation with favorable computational efficiency. The analysis highlights critical trade-offs between detection accuracy, computational resources, and adaptability, suggesting that hybrid architectures combining these complementary strengths offer optimal solutions—with Random Forest serving as primary detection for well-labeled environments, supported by Isolation Forest for novel threat identification, while resource-constrained or label-limited contexts benefit from K-means baseline monitoring enhanced by Isolation Forest anomaly detection. Future research should focus on advanced ensemble frameworks integrating multiple learning paradigms and investigating temporal model stability, as increasingly sophisticated cyber threats necessitate robust ML-based detection systems that leverage diverse algorithmic approaches for comprehensive network security.

REFERENCES

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A systematic review on supervised and unsupervised machine learning algorithms for data science. *Supervised and Unsupervised Learning for Data Science*, 3–21.

Alom, M. Z., & Taha, T. M. (2017). Network intrusion detection for cyber security using unsupervised deep learning approaches. *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, 63–69. https://doi.org/10.1109/NAECON.2017.8268746

Bin Sarhan, B., & Altwaijry, N. (2023). Insider Threat Detection Using Machine Learning Approach. *Applied Sciences*, *13*(1). https://doi.org/10.3390/app13010259

Bohara, A., Noureddine, M. A., Fawaz, A., & Sanders, W. H. (2017). An Unsupervised Multi-Detector Approach for Identifying Malicious Lateral Movement. *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, 224–233. https://doi.org/10.1109/SRDS.2017.31

Chen, S., Webb, G. I., Liu, L., & Ma, X. (2020). A novel selective naïve Bayes algorithm. *Knowledge-Based Systems*, *192*, 105361. https://doi.org/https://doi.org/10.1016/j.knosys.2019.105361

Jony, A. I., & Arnob, A. K. B. (2024). Securing the Internet of Things: Evaluating Machine Learning Algorithms for Detecting IoT Cyberattacks Using CIC-IoT2023 Dataset. *International Journal of Information Technology and Computer Science*, *16*(4), 56–65. https://doi.org/10.5815/ijitcs.2024.04.04

Kim, S., & Park, K. J. (2021). A survey on machine-learning based security design for cyber-physical systems. *Applied Sciences (Switzerland)*, *11*(12). https://doi.org/10.3390/app11125458

Korteling, J. E. (Hans), van de Boer-Visschedijk, G. C., Blankendaal, R. A. M., Boonekamp, R. C., & Eikelboom, A. R. (2021). Human-versus artificial intelligence. *Frontiers in Artificial Intelligence*, *4*, 622364.

Maikano, F. A. (2024). 8 Machine Learning Approaches for Cyber Bullying Detection in Hausa Language Social Media: a Comprehensive Review and Analysis. *MACHINE LEARNING APPROACHES… Maikano FJS FUDMA Journal of Sciences (FJS*, *8*(3), 344–348. https://doi.org/10.33003/fjs-2024-0803-2517

Murtagh, F., & Contreras, P. (2017). Algorithms for hierarchical clustering: an overview, II. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(6), e1219.

Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *Journal of Sensor and Actuator Networks*, *12*(2). https://doi.org/10.3390/jsan12020029

Sakhai, M., & Wielgosz, M. (2021). *Modern Cybersecurity Solution using Supervised Machine Learning*. http://arxiv.org/abs/2109.07593

Tao, X., Peng, Y., Zhao, F., Zhao, P., & Wang, Y. (2018). A parallel algorithm for network traffic anomaly detection based on Isolation Forest. *International Journal of Distributed Sensor Networks*, *14*, 155014771881447. https://doi.org/10.1177/1550147718814471

APPENDIX

Evaluation

```
In [23]: # Calculate evaluation metrics
precision = precision_score(testing_labels, anomaly_scores)
recall = recall_score(testing_labels, anomaly_scores)
f1 = f1_score(testing_labels, anomaly_scores)


print("Precision:", precision)
print("Recall:", recall)
print("F1 score:", f1)
```

```
Precision: 0.8345284059569774
Recall: 0.9365521510368306
F1 score: 0.8826017208691848
```

```
In [30]: results = {"Precision": precision, "Recall": recall, "F1 score": f1}
index = ["Metrics"]
results = pd.DataFrame(results, index=index)
```

```
In [31]: results
```

Out[31]:

	Precision	Recall	F1 score
Metrics	0.834528	0.936552	0.882602

Figure 6: Isolation Forest Result Evaluation

```
In [10]: # Calculate evaluation metrics
silhouette_avg = silhouette_score(data, labels)
ch_index = calinski_harabasz_score(data, labels)
db_index = davies_bouldin_score(data, labels)

print("Silhouette score:", silhouette_avg)
print("Calinski-Harabasz index:", ch_index)
print("Davies-Bouldin index:", db_index)
```

```
Silhouette score: 0.8622363933400274
Calinski-Harabasz index: 48526.004230398365
Davies-Bouldin index: 0.6062838469451424
```

```
In [35]: results = {"Silhouette score": silhouette_avg, "Calinski-Harabasz index": ch_index, "Davies-Bouldin index": db_index}
index = ["Metrics"]
results = pd.DataFrame(results, index = index)
results
```

Out[35]:

	Silhouette score	Calinski-Harabasz index	Davies-Bouldin index
Metrics	0.862236	48526.00423	0.606284


```
In [ ]:
```

Figure 7: K-Means Evaluation