International Journal of Safety and Security Engineering
Vol. 15, No. 1, January, 2025, pp. 171-179
Journal homepage: http://iieta.org/journals/ijsse

A Comparative Study of Incremental and Batch Machine Learning Methodologies for Network Intrusion Detection

Rawabi Alanzi[1], Murad A. Rassam[1,2*]

[1] Department of Information Technology, College of Computer, Qassim University, Buraydah 51452, Saudi Arabia
[2] Faculty of Engineering and Information Technology, Taiz University, Taiz 6801, Yemen

Corresponding Author Email: m.qasem@qu.edu.sa

https://doi.org/10.18280/ijsse.150118

Received: 30 November 2024
Revised: 27 December 2024
Accepted: 15 January 2025
Available online: 31 January 2025

Keywords:
machine learning, intrusion detection systems, incremental learning, batch learning, detection framework

ABSTRACT

The exponential growth of digital networks necessitates robust intrusion detection systems (IDS) to counter evolving cyber threats effectively. Machine learning offers adaptive solutions for these challenges. This study evaluates the comparative performance of Incremental Learning and Batch Learning methodologies for IDS using two datasets, UNSW-NB15 and CI-CIDS 2017. Three machine learning algorithms—Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forest (RF)—were assessed. Results indicate that Incremental Learning outperforms Batch Learning in dynamic environments. For instance, on the CI-CIDS 2017 dataset, SVM with Incremental Learning achieved a precision of 98%, recall of 97%, and an F1-score of 97.5%, compared to Batch Learning, which obtained 95%, 93%, and 94%, respectively. These findings highlight Incremental Learning's adaptability to real-time threats despite higher computational demands. This research offers valuable insights for optimizing IDS using machine learning and proposes a framework for enhancing network security.

1. INTRODUCTION

Cybersecurity and network security concerns have grown significantly with the rapid expansion of the Internet and communication technologies and the increasing variety of network applications. Network security has become a critical issue as attack methods have also evolved. Key challenges include the sheer volume of data to be analyzed and the rapid emergence of novel attack techniques.

Intrusion detection systems (IDS) play a pivotal role in identifying and mitigating cyber threats by monitoring network traffic to detect anomalies and cyberattacks [1]. However, intrusion detection faces new challenges, such as scalability, adaptability, and efficiency in the age of big data and rapidly evolving attack strategies. Previous studies have proposed several approaches to improve IDS performance using machine learning techniques. For instance, some works have focused on enhancing system performance through multi-stage frameworks, feature selection, and hyperparameter optimization [2]. Others, like the previous study [3], integrated chi-square feature selection with support vector machines (SVM) to improve accuracy and reduce false positives. Feature reduction techniques, such as CFS, IG, and GR, have also been utilized to enhance classifier efficiency [4]. Similarly, anomaly detection models for cloud computing environments have employed SVM with particle swarm optimization (PSO) to refine feature selection and improve classification results [5]. Incremental learning has emerged as a promising methodology in IDS, addressing challenges such as memory and computational requirements when handling large datasets. For example, researchers [6] demonstrated how incremental techniques could reduce training time and classifier size while maintaining performance comparable to batch learning. Other work [7], developed multi-classifiers based on incremental learning to improve training efficiency on massive datasets. Furthermore, distance-weighted outlier detection and domain-specific anomaly detection methods have been explored to detect hidden anomalies in cellular networks [8, 9]. Deep learning approaches have also been employed, such as the system proposed in previous study [10] using deep neural networks (DNN) and optimization algorithms to detect network intrusions in cloud environments. Two-layer anomaly detection models combining Naive Bayes and customized KNN classifiers have achieved reasonable detection rates with fewer false alarms, particularly for rare attack types like U2R [11].

Despite these advancements, there is a limited comparative analysis of incremental learning and batch learning methodologies in the context of IDS. While incremental learning offers adaptability to new threats, it may incur higher computational costs, whereas batch learning is computationally efficient but slower to adapt to dynamic environments [12]. This study addresses this gap by investigating the comparative effectiveness of incremental and batch-learning methodologies in network intrusion detection. The main contributions of this work can be summarized as follows:

·Comparing batch and incremental learning methodologies in IDS systems based on machine learning.

·Evaluating the performance of several machine learning

algorithms using two recent datasets: CI-CIDS 2017 and UNSW-NB15.

The remainder of this paper is structured as follows: Section 2 introduces key concepts of intrusion detection and machine learning alongside a detailed review of relevant literature. Section 3 describes the methodology and model design for the comparative framework. Section 4 presents the experimental results, while Section 5 discusses the findings, highlighting the strengths and limitations of each methodology. Finally, Section 6 concludes the paper.

2. BACKGROUND

2.1 Intrusion detection system

IDS is software that monitors and tracks malicious activity (attacks) on a single or networked computer to steal, censor, or tamper with network protocols. Due to their reliance on known malware signatures or patterns, traditional intrusion detection systems are only sometimes successful against novel and emerging threats. Nevertheless, machine learning may be applied to create IDS systems that are more effective and adaptable. IDS systems using machine learning can react to threats and learn from fresh data. They can now identify new and evolving malware with more effectiveness. Machine learning is a promising technology for creating IDS systems that are more effective and adaptable [13]. IDS can be categorized based on how it is deployed or what detection techniques it uses. In Figure 1, a classification taxonomy is provided [14].

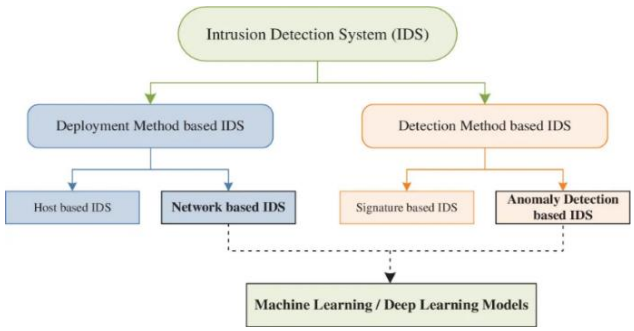


Figure 1. Intrusion detection taxonomy

Host-Based-IDS (HIDS) monitors system activity, such as file modifications and memory usage. Internal monitoring establishes whether a system has been mainly compromised depending on audit trails and system logs. This method relies on individual hosts keeping an eye on the gadget and identifying inappropriate use of the sources. Well-known HIDS examples include Tripwire and OSSEC [15].

Network-based IDS (NIDS) aids in detecting network security vulnerabilities in businesses by system administrators. Specification-based NIDSs specify the permitted activities and mark any other activity as prohibited. Behavior recognition NIDSs examine activity patterns and the environment for additional proof of attacks [16]. Regarding the detection mechanism, IDS are categorized into signature-based IDS and anomaly-based IDS. Signature-Based Intrusion Detection (SIDS) aims to identify known attacks with established patterns or signatures. Although signature-based IDSs frequently perform well on tasks involving detecting known attacks, they cannot identify new or zero-day attacks since their patterns are unknown [17]. On the other hand, anomaly detection-based intrusion detection (AIDS) is used to identify unknown attacks from pre-defined regular behaviors and detect zero-day attacks. However, they frequently perform worse than signature-based IDSs regarding known attack detection [17].

2.2 Machine learning (ML)

Artificial intelligence (AI) studies how machines think and act like humans [18]. Machine learning (ML) is a collection of methods using mathematical formulas to find, investigate, and extract patterns from data automatically. Meaningful data can be extracted and acquired to aid machine learning algorithms in making defensible decisions and predictions.

Several classification approaches use supervised machine learning techniques [19]. Each approach uses a different dataset and data classification method. These are some key machine-learning algorithms employed in classification problems.

(1) Decision Tree (DT)

One of the fundamental supervised machine learning algorithms, DT, applies a set of decisions (rules) to classify and predict the dataset. The model is organized like a typical tree, with nodes, branches, and leaves. Each node is an attribute or a property [14].

(2) K-Nearest Neighbor (KNN)

One of the most straightforward supervised machine learning (ML) algorithms, KNN, uses the concept of "feature similarity" to determine the class of a given data sample. It determines a sample's identity based on its neighbors by measuring its distance from them [14].

(3) Support Vector Machine (SVM)

The max-margin separation hyperplane in n-dimensional feature space is the foundation for the supervised machine learning method SVM. It is utilized to resolve both linear and nonlinear issues [14].

(4) Random Forest (RF)

It is a supervised learning technique that classifies data using a group of decision trees. RF constructs numerous decision trees and blends them to produce a more precise and reliable prediction of class membership [20].

(5) Naïve Bayes

The Nave Bayes (NB) and Gaussian Nave Bayes (GNB) algorithms utilize the Bayes theorem, assuming that the dataset's features are independent [21].

2.2.1 Incremental machine learning

A machine learning technique in which an AI model gradually takes in all new information while maintaining what it already knows. It mimics how humans learn by continuously gaining new knowledge while preserving and expanding upon what they already know. When data arrives sequentially or storing all the data for processing is not practical, incremental learning becomes essential. Also, using the incremental flow of data to retrain a model and incrementally build up its knowledge over time is known as online learning. Like transfer learning, the ML model and its learning parameters are often kept in mind between incremental training iterations. To improve the previously inferred information, the stored model is incrementally trained over new data [22]. The incremental learning process is shown below in Figure 2 as adopted from previous study [23].

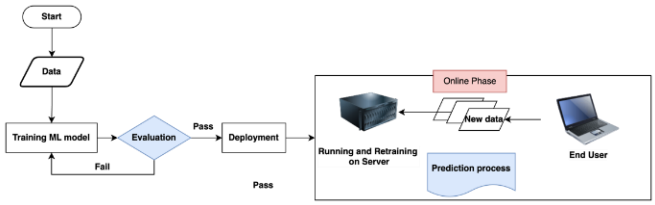


Figure 2. Process of incremental learning in machine learning

2.2.2 Batch machine learning

Batch learning is sometimes referred to as offline learning. Only past learning allows a running model on a server or in the cloud to forecast the results of fresh data. However, if the model needs to be retrained, we should replace the previously trained model and train it again using both old and new data. The process of batch learning is shown below in Figure 3 [23].

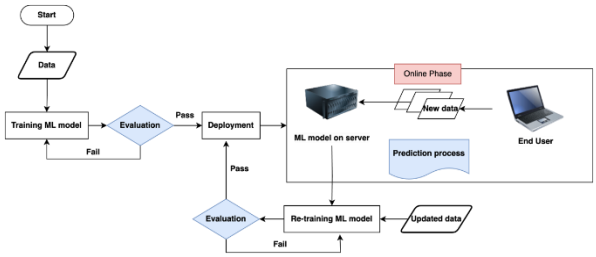


Figure 3. Process of batch learning in machine learning

2.3 Literature review

Several studies have investigated various methodologies to enhance the performance of intrusion detection systems (IDS) performance, focusing on frameworks, feature selection techniques, and anomaly detection models. For instance, a multi-stage machine learning framework was proposed [2], integrating data augmentation, feature selection, and hyperparameter optimization to enhance system performance, with an emphasis on precision, true positive rate (TPR), and false positive rate (FPR). Similarly, previous research [3] combined chi-square feature selection with support vector machines (SVM) to improve accuracy and reduce false positives. In another study [4], a feature reduction methodology compared correlation-based feature selection (CFS), information gain (IG), and gain ratio (GR) using Naive Bayes classifiers on the NSL-KDD dataset, demonstrating improvements in efficiency and effectiveness.

Anomaly detection systems tailored for specific environments have also been explored. For example, previous research [5] developed an anomaly-based IDS for cloud computing using SVM classifiers enhanced by Particle Swarm Optimization (PSO) for feature selection, evaluated on the NSL-KDD dataset using criteria such as detection accuracy and TPR. Another study [10] introduced the Anomalous Network Intrusions Detection System (ANIDS), which utilized deep neural networks (DNN) and optimization algorithms like genetic algorithms (GA) and simulated annealing (SAA) to optimize parameters for improved detection.

Incremental learning approaches have emerged as a promising solution to address challenges like high memory and computational requirements. In previous research [6], incremental techniques reduced training time and classifier size while maintaining performance comparable to batch algorithms. Similarly, previous research [7] proposed a multi-classifier system using Proximal Support Vector Machines (PSVM) combined with incremental learning, improving training efficiency on massive datasets.

Other works have focused on supervised and hybrid anomaly detection models. For instance, previous research [8] proposed a K-Nearest Neighbor (KNN) algorithm that combines profile-based and domain-based techniques to detect hidden errors, introducing performance improvements through derived formulas and metrics. Additionally, previous research [9] explored a distance-weighted outlier detection algorithm combined with enhanced classification methods using Victorian support machines and smart agents. Furthermore, previous research [11] presented a two-layer anomaly detection model incorporating Naive Bayes and customized KNN classifiers with linear skew analysis for dimensionality reduction, achieving high detection rates with fewer false alarms, particularly for U2R attack scenarios.

Despite these advancements, the comparative analysis of incremental learning and batch learning methodologies in the context of IDS remains limited. While incremental learning offers adaptability to new threats by incorporating new data dynamically, it often incurs higher computational costs. On the other hand, batch learning is computationally efficient but slower in adapting to rapidly evolving threats. This study addresses this gap by systematically comparing the effectiveness of incremental and batch-learning methodologies in intrusion detection to provide insights into their strengths and limitations.

3. PROPOSED METHODOLOGY

This section outlines the methodology used in this study, including data collection, preprocessing, model design, and evaluation. The proposed framework evaluates supervised machine learning algorithms for classifying activities as suspicious or non-suspicious. Four machine learning algorithms, Random Forest, Support Vector Machine (SVM), Naïve Bayes, and K-Nearest Neighbor (KNN), are applied independently to compare their performance. Incremental and batch learning approaches are evaluated separately to understand their strengths and weaknesses. Figure 4 provides an overview of the framework.

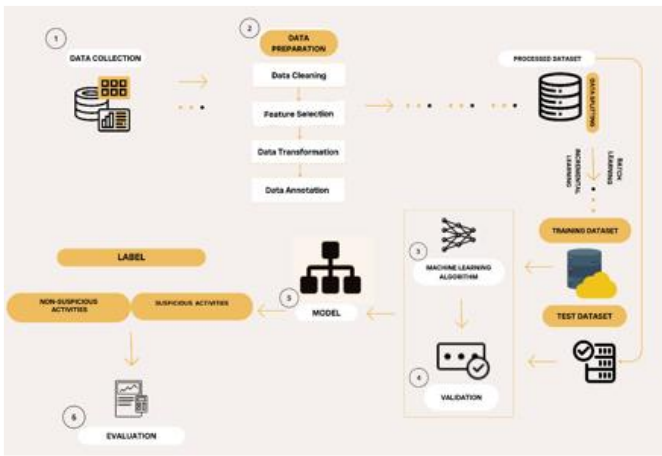


Figure 4. The general framework of the proposed methodology

3.1 Dataset

This study uses two widely recognized datasets:

·UNSW-NB15 Dataset: Collected by the Cyber Range Lab in 2015, this dataset integrates contemporary network traffic and various attack behaviors, including Backdoors, DoS, Exploits, Reconnaissance, and Worms. It consists of 2.5 million records with 49 features, and it is balanced, containing equal proportions of normal and attack logs. The dataset's richness and balance make it suitable for training and evaluating intrusion detection models [23].

·CI-CIDS 2017 Dataset: Developed by the Canadian Institute for Cybersecurity (CIC), this large-scale dataset comprises over 80 million records, each representing a network flow. It features both normal and malicious traffic and includes annotations for 15 types of attacks. With 81 features, the dataset provides extensive coverage of network behaviors, offering a robust basis for testing intrusion detection systems [24].

The datasets were selected for their balanced representation of attack and normal traffic, their scale, and the diversity of attack types. Thus, they are ideal for evaluating the generalizability and effectiveness of machine learning models.

3.2 Data preparation

To ensure that the data is suitable for analysis, preprocessing was performed in several stages:

1. Data Cleaning: Noise, incomplete, and irrelevant data were removed or modified. Errors were documented, and the effectiveness of the cleaning process was assessed against predefined requirements to ensure high-quality input data.

2. Feature Selection: Selecting relevant features is crucial for reducing computation time and improving model accuracy. The Binary Particle Swarm Optimization (BPSO) algorithm was used to rank features, with the SVM model as the fitness function. Features were retained if their value was set to 1 and removed if set to 0. This step minimized the impact of redundant features, improving detection time and classification accuracy.

3. Data Annotation: Data was labeled as "suspicious" or "non-suspicious" to facilitate supervised learning. This labeling ensures that models learn to differentiate between normal and malicious activities effectively.

4. Data Splitting: Datasets were divided into training and testing sets, with at least 20% of the data reserved for testing. This ensures that models are evaluated on unseen data, providing a reliable measure of their generalizability.

5. Data Transformation: It converts data into a more acceptable format for analysis. Cleaning, filtering, and standardizing the data may be required. The process of transforming data ensures that the data is clean and consistent. It refers to defining a restricted range of attribute values that might aid in improving detection results and avoiding numerical issues during calculations. This step is completed to guarantee that the detecting system can effectively process the data. Many traits have values that vary widely among species and ranges. As a result, to properly process and analyze the data, these numbers must be restricted to a particular range. Normalization methods based on min-max. Normalization of the average range [0, 1] (from least to most): This function converts attribute values to a value between 0 and 1. This is accomplished by subtracting the lowest value from each and dividing the result by the range (highest value minus lowest value) as in Eq. (1).

$$X' = \frac{x - minA}{maxA - minA} \tag{1}$$

where, x and primes boldrimes are the normalized attribute value and the value to be normalized, respectively, before normalization, the minimum and maximum allowed values for attribute A are $MinA$ and $MaxA$ [10].

3.3 Model design

The machine learning models were designed with two primary phases:

·Training Phase: Models were trained on labeled data to learn the relationship between input features and their corresponding outputs. Parameters were iteratively adjusted to minimize prediction errors.

·Testing Phase: Models were evaluated on unseen data to measure their generalization ability to new situations. This step highlights the models' robustness, precision, and potential limitations, such as overfitting or underfitting.

3.4 Evaluation

The models were assessed using standard evaluation metrics derived from the confusion matrix:

(1) Confusion Matrix Metrics is to evaluate the models' ability to classify activities correctly, metrics such as the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), and false negative rate (FNR) were calculated.

a. TP represents normal conduct that is accurately expected and given in Eq. (2).

$$\text{True Positive Rate} = \frac{TP}{TP+FN} \tag{2}$$

b. FP represents normal behavior that is incorrectly supposed to be abnormal and given in Eq. (3).

$$\text{False Positive Rate} = \frac{FP}{FP+TN} \tag{3}$$

c. TN represents normal performance that is identified as correct and given in Eq. (4).

$$\text{True Negative Rate} = \frac{TN}{TN+FP} \tag{4}$$

d. FN represents aberrant performance that is misdiagnosed as normal and given in Eq. (5).

$$\text{False Negative Rate} = \frac{FN}{FN+TP} \tag{5}$$

(2) Accuracy is a trained model's proportion of true predictions. It is derived by dividing the number of forecasts by the number of correct guesses and given in Eq. (6).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

(3) Precision is another metric showing the percentage of positive anticipated values. Precision aids in visualizing the machine learning model's dependability and classifying the model as positive and given in Eq. (7).

$$\text{Precision}=\frac{TP}{TP+FP} \quad (7)$$

(4) Recall measures the proportion of positive cases correctly identified by the model and given in Eq. (8).

$$\text{Recall}=\frac{TP}{TP+FN} \quad (8)$$

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

This section describes the model's implementation and experimental results. First, each learning strategy is evaluated separately. Then, batch and Incremental Learning strategies are compared. Confusion matrices assess the performance of each method and show how they differ. Table 1 shows the results of the tests carried out to evaluate the performance of the three models in batch learning and incremental learning on the UNSW-NB15 Dataset, whereas Table 2 reports the results on the CI-CIDS 2017 Dataset. Furthermore, Figures 5 and 6 present the confusion matrices for the proposed models on both datasets.

4.1 Results on UNSW-NB15 dataset

As Table 1 and Figure 5 show, the evaluation of the UNSW-NB15 dataset demonstrates high performance across all metrics (accuracy, precision, recall, and F1 score) for both batch and incremental learning approaches, with minor differences in their results. For instance, incremental learning achieved perfect recall (1.0) with Random Forest (RF), indicating its capability to detect all positive cases. Similarly, the performance of K-Nearest Neighbor (KNN) was comparable between both approaches, with minor deviations. Batch learning slightly outperformed incremental learning for Support Vector Machine (SVM) in precision and accuracy, but the difference was minimal.

These findings align with prior studies, which highlight the strengths of incremental learning in dynamic environments where data evolves. Incremental learning's ability to adapt to changes in network traffic patterns is particularly valuable for real-time intrusion detection systems, addressing challenges discussed in previous research [10]. On the other hand, batch learning demonstrated its computational efficiency and reliability when processing static datasets, consistent with findings in previous research [25].

From a practical perspective, the results suggest that both learning approaches are suitable for intrusion detection depending on the application scenario. For systems requiring real-time adaptability, incremental learning offers a clear advantage. However, batch learning may be more appropriate for offline analysis or environments where computational efficiency is prioritized.

4.2 Results on CI-CIDS 2017 dataset

As reported in Table 2 and Figure 6, the results on CI-CIDS 2017 dataset further highlights the effectiveness of incremental learning, especially for algorithms like KNN and RF. Incremental learning with KNN achieved 99.51% accuracy, outperforming batch learning, which suffered from reduced accuracy and recall. This emphasizes incremental learning's ability to handle large-scale datasets with diverse attack types effectively.

For SVM, incremental learning achieved high accuracy (99.67%) and precision (99.17%), outperforming batch learning in recall (99.77% vs. 72.36%) and F1 score (99.67% vs. 63.56%). These results align with previous research [8, 9], emphasizing the importance of feature selection and dynamic updates for maintaining high classification performance in changing environments.

The RF algorithm demonstrated robust performance under both learning strategies, achieving near-perfect metrics (accuracy: 99.96% for incremental and 99.72% for batch). This consistency underscores RF's ability to handle imbalanced and high-dimensional datasets [2].

Table 1. Incremental and batch learning and batch learning on UNSW-NB15 dataset

		Performance Measures			
Model Name		Accuracy	Precision	Recall	F1 Score
KNN	Incremental learning	99.51	99.08	99.77	99.43
	Batch learning	93.06	93.19	93.06	93.08
SVM	Incremental learning	99.67	99.17	99.47	99.77
	Batch learning	99.98	80.11	72.36	63.56
RF	Incremental learning	99.96	100.00	99.91	99.96
	Batch learning	99.72	99.72	99.72	99.72

Table 2. Batch and incremental learning metrics on CI-CIDS 2017 dataset

		Performance Measures			
Model Name		Accuracy	Precision	Recall	F1 Score
KNN	Incremental learning	0.98	0.99	0.99	0.99
	Batch learning	0.98	0.98	0.98	0.98
SVM	Incremental learning	0.98	0.99	0.99	0.99
	Batch learning	0.99	0.99	0.99	0.99
RF	Incremental learning	0.99	0.99	1.0	0.99
	Batch learning	0.99	0.99	0.99	0.99

UNSW-NB15 Dataset

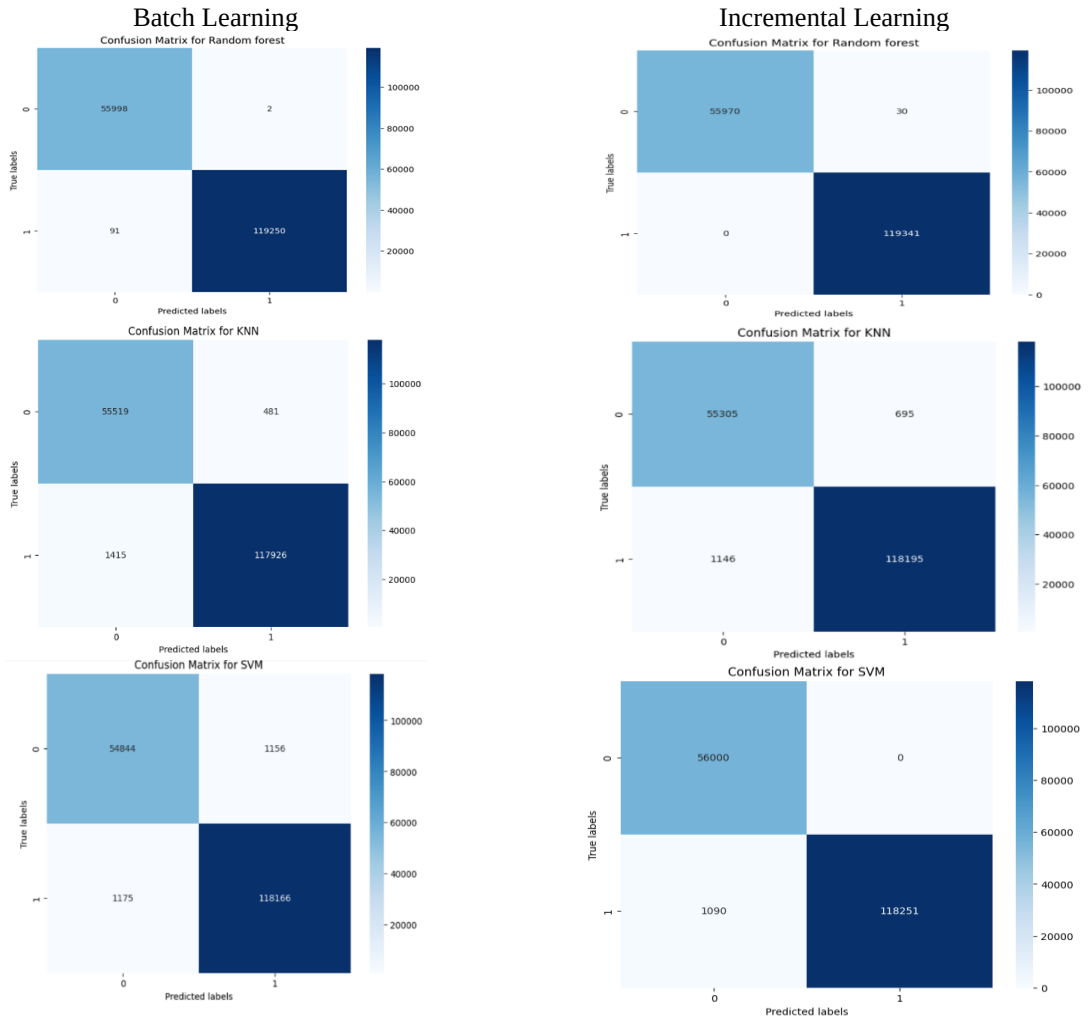
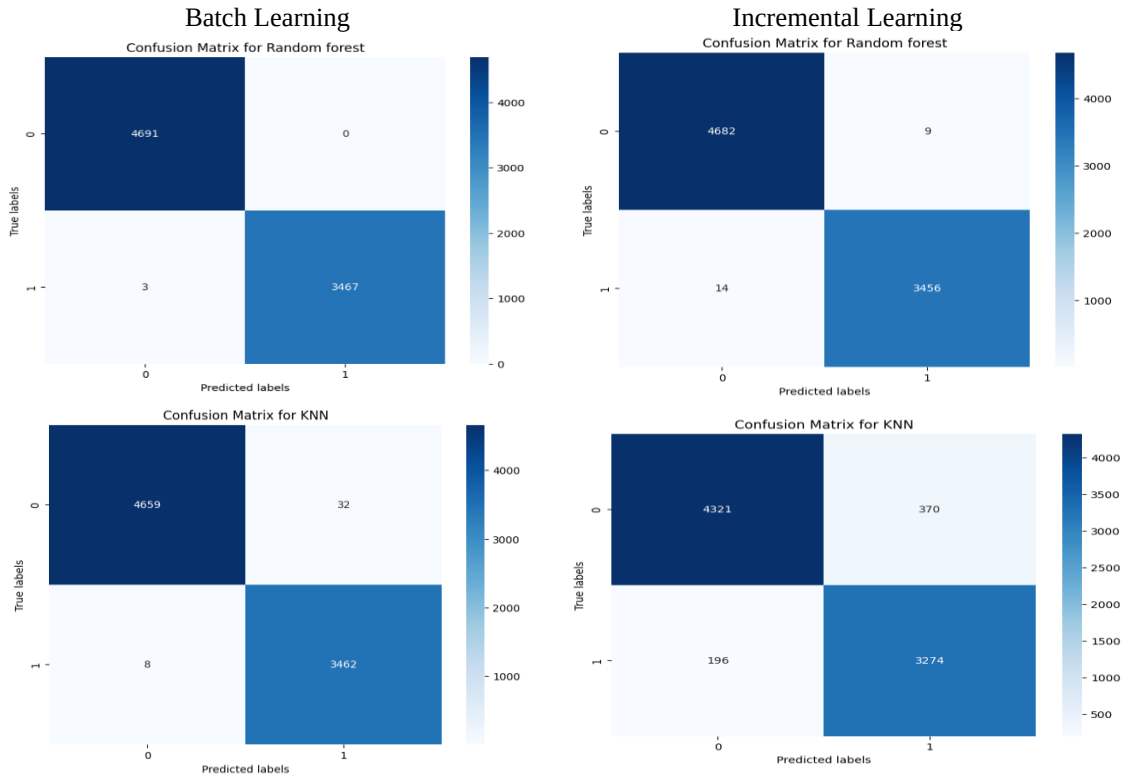


Figure 5. Confusion matrices for the proposed batch and incremental models on the UNSW-NB15 dataset

CI-CIDS 2017 Dataset



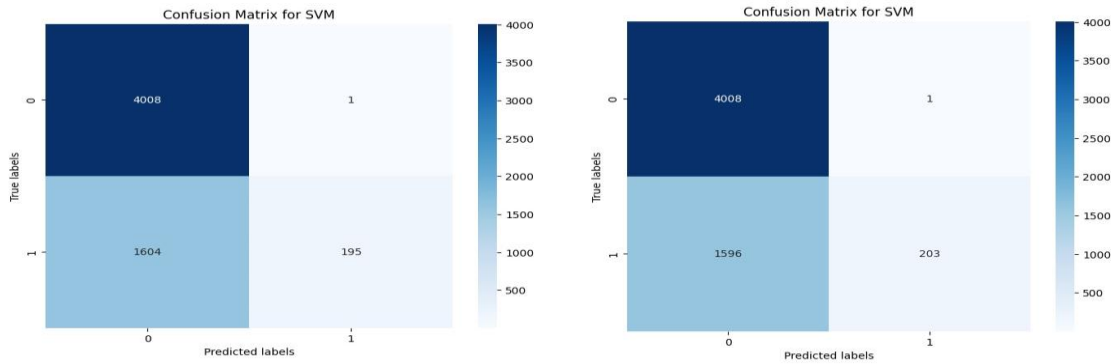


Figure 6. Confusion matrices for the proposed batch and incremental models on CI-CIDS 2017 dataset

4.3 Implications and contributions

These findings contribute to the ongoing debate about the comparative effectiveness of incremental and batch learning in intrusion detection systems. The results provide several practical implications:

Incremental Learning's Adaptability: Incremental learning's ability to maintain high performance while adapting to new data makes it particularly suitable for real-time intrusion detection systems, where attack patterns can change rapidly.

Batch Learning's Stability: Despite its limitations in handling evolving data, batch learning remains a reliable option for scenarios involving static datasets or offline processing, where computational efficiency is critical.

Algorithm-Specific Insights: While KNN and RF performed well under both learning strategies, SVM demonstrated a stronger dependence on the learning approach chosen. This emphasizes the need for careful algorithm selection based on the application's requirements.

4.4 Comparison with existing literature

This study builds upon prior works by directly comparing batch and incremental learning strategies using two large-scale datasets. While previous studies [3, 5] primarily focused on the standalone performance of individual algorithms, this work highlights how different learning methodologies impact algorithmic effectiveness in various contexts.

Moreover, the findings extend the literature by showcasing the adaptability of incremental learning, which has often been underexplored in large-scale, real-world datasets like CI-CIDS 2017. The insights from this study offer practical guidelines for implementing machine learning-based intrusion detection systems tailored to specific operational needs.

4.5 Practical applications

The results have several practical applications in the field of network security:

Real-Time Monitoring: Incremental learning can be leveraged for systems requiring continuous updates, such as intrusion detection in dynamic and large-scale networks.

Resource Optimization: Batch learning can be employed in environments with limited computational resources where real-time adaptability is not critical.

Hybrid Systems: A hybrid approach combining incremental and batch learning could balance real-time adaptability with computational efficiency, particularly in scenarios with fluctuating workloads.

5. DISCUSSION

In the UNSW-NB15 dataset, KNN showed better incremental and batch learning accuracy, with a precision rate 0.99. Batch learning achieved the highest percentage of 55,519 TP cases, while incremental learning classified a larger number of 118,195 TN cases without abnormal or malicious activity. Regarding false positive (FP) cases, batch learning classified more cases than incremental learning.

Regarding SVM, batch learning was better in accuracy at 0.99, while incremental learning showed 0.98. Both batch and incremental learning performed well in classifying true positive (TP) cases, with batch learning achieving 54,844 TP cases and incremental learning achieving 56,000 TN cases. Regarding RF, batch learning was 1.00 better in recall, while incremental learning showed a 0.99 performance. Batch learning identified many actual network attacks (TP) and classified many normal activities (TN) as attacks, resulting in unnecessary alarms. Incremental learning significantly reduced the number of false alarms (FP) by classifying only 30 normal activities as attacks, while in batch learning, 91 normal activities were classified as attacks.

Regarding the CI-CIDS 2017 dataset, when learning incrementally, KNN performed better than batch learning (recall 0.989187 and F1-measure 0.989210), with recall (0.990397) and F1-measure (0.992272). KNN classified some negative cases as false positives by producing more false positives (3,701) with incremental learning compared to batch learning (828). Compared with batch learning, which produced 196 false negatives, KNN incremental learning produced 4370 false negatives, indicating that some positive cases were incorrectly classified. For SVM, incremental learning outperformed batch learning across parameters (F1 measure, precision, and recall), with incremental learning being 0.99 accurate compared to batch learning, which was 0.80 less accurate. On the other hand, incremental learning achieved a high recall rate of 0.99, but batch learning achieved a weak rate of 72%. Finally, the performance of batch learning in F-Score was 0.63, which is considered a very weak performance compared to the achievement of incremental learning with a higher performance rate of 0.99. RF achieves remarkable accuracy (0.9995 and 0.9998, respectively) and high F1 metrics (0.999470 and 0.999874) while maintaining good incremental and batch learning performance.

Batch learning was less effective than RF incremental learning regarding recall (1.0 vs. 0.99) or F1 scale (0.99987 vs. 0.999470). Both incremental RF and batch learning show false positive counts of (1604 and 1596, respectively), indicating their exceptional accuracy in identifying positive situations.

False negatives (0) are also produced by batch learning or incremental learning of the RF, indicating that all positive cases are accurately identified. If the dataset is static and does not show significant changes over time, incremental learning may not be the most efficient or cost-effective approach. This is especially true for the UNSW-NB15 -NB15 dataset, where both incremental and batch learning showed similar performance results. The main benefit of incremental learning is its ability to adapt to evolving data patterns. This advantage is not fully utilized if the data set remains relatively constant. Incremental learning algorithms also require more computational resources than batch learning, especially when dealing with large data sets.

On the other hand, the CICID17 dataset showed a difference in performance between incremental learning and batch learning. The performance of incremental learning was better and more effective than batch learning, which means there is a great benefit and a big difference between using them. The CICID dataset shows the advantages of incremental learning in dynamic network intrusion detection scenarios. The ability of incremental learning to adapt to new data and reduce false positives makes it a promising approach for real-world cybersecurity applications.

6. CONCLUSIONS

This study evaluated batch and incremental learning approaches for classifying and detecting suspicious activities in intrusion detection systems (IDS) using machine learning (ML). The experiments utilized the UNSW-NB15 and CI-CIDS 2017 datasets with SVM, KNN, and RF algorithms. While both learning methods demonstrated comparable performance across several metrics, incremental learning offered notable advantages in specific scenarios. For example, KNN in incremental mode showed a slight accuracy improvement, and RF achieved fewer false positives. On the CI-CIDS dataset, incremental learning significantly outperformed batch methods for KNN and SVM, achieving higher precision, recall, and F1 scores, highlighting its suitability for dynamic, evolving environments.

KNN excelled over batch learning for specific tasks, while SVM performed better than ensemble methods. RF consistently achieved high accuracy and F1 scores, proving its versatility across both approaches. These findings underline that while batch learning remains efficient for static datasets, incremental learning is better suited to adapt to real-time or dynamically changing environments.

Employing more advanced ML algorithms or hybrid techniques may provide new insights in future research. In addition, addressing the computational cost of incremental learning in scenarios with frequent data changes is another critical avenue. Furthermore, a metric can be employed to detect change points in the data. This is important because while effective in dynamic environments, incremental learning can be computationally expensive. Batch learning offers a more cost-effective baseline but might not adapt well to data changes. Therefore, changing point detection allows us to switch to incremental learning when necessary, ensuring continuous learning without missing data points.

REFERENCES

[1] Pasikhan, A.M., Clark, J.A., Gope, P. (2023). Incremental hybrid intrusion detection for 6LoWPAN. Computers & Security, 135: 103447. https://doi.org/10.1016/j.cose.2023.103447

[2] Injadat, M., Moubayed, A., Nassif, A.B., Shami, A. (2020). Multi-stage optimized machine learning framework for network intrusion detection. IEEE Transactions on Network and Service Management, 18(2): 1803-1816. https://doi.org/10.1109/TNSM.2020.3014929

[3] Thaseen, I.S., Kumar, C.A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University-Computer and Information Sciences, 29(4): 462-472. https://doi.org/10.1016/j.jksuci.2015.12.004

[4] Mukherjee, S., Sharma, N. (2012). Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology, 4: 119-128. https://doi.org/10.1016/j.protcy.2012.05.017

[5] Sakr, M.M., Tawfeeq, M.A., El-Sisi, A.B. (2019). Network intrusion detection system based PSO-SVM for cloud computing. International Journal of Computer Network and Information Security, 14(3): 22. https://doi.org/10.5815/ijcnis.2019.03.04

[6] Domeniconi, C., Gunopulos, D. Incremental support vector machine construction. [Online]. Available: http://www.cs.uci.edu/.

[7] Li, K., Huang, H.K. (2002). Incremental learning proximal support vector machine classifiers. In Proceedings. International Conference on Machine Learning and Cybernetics, pp. 1635-1637. https://doi.org/10.1109/ICMLC.2002.1167488

[8] Al Mamun, S.A., Valimaki, J. (2018). Anomaly detection and classification in cellular networks using automatic labeling technique for applying supervised learning. Procedia Computer Science, 140: 186-195. https://doi.org/10.1016/j.procs.2018.10.328

[9] Ganapathy, S., Yogesh, P., Kannan, A. (2012). Intelligent agent-based intrusion detection system using enhanced multiclass SVM. Computational Intelligence and Neuroscience, 2012(1): 850259. https://doi.org/10.1155/2012/850259

[10] Chiba, Z., Abghour, N., Moussaid, K., Rida, M. (2019). Intelligent approach to build a Deep Neural Network based IDS for cloud environment using combination of machine learning algorithms. Computers & Security, 86: 291-317. https://doi.org/10.1016/j.cose.2019.06.01

[11] Sillito, R.R. (2010). Incremental semi-supervised learning for anomalous trajectory detection.

[12] Pajouh, H.H., Dastghaibyfard, G., Hashemi, S. (2017). Two-tier network anomaly detection model: A machine learning approach. Journal of Intelligent Information Systems, 48: 61-74. https://doi.org/10.1007/s10844-015-0388-x

[13] Zamani, M., Movahedi, M. (2013). Machine learning techniques for intrusion detection. arXiv preprint arXiv:1312.2177. http://arxiv.org/abs/1312.2177

[14] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1): e4150. https://doi.org/10.1002/ett.4150

[15] Martins, I., Resende, J.S., Sousa, P.R., Silva, S., Antunes, L., Gama, J. (2022). Host-based IDS: A review and open

issues of an anomaly detection system in IoT. Future Generation Computer Systems, 133: 95-113. https://doi.org/10.1016/j.future.2022.03.001

[16] Kasongo, S.M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199: 113-125. https://doi.org/10.1016/j.comcom.2022.12.010

[17] Yang, L., Shami, A. (2022). IDS-ML: An open source code for intrusion detection system development using machine learning. Software Impacts, 14: 100446. https://doi.org/10.1016/j.simpa.2022.100446

[18] "Machine Learning | An Introduction | by Gavin Edwards | Towards Data Science." Accessed: Nov. 02, 2023. [Online]. Available: https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0.

[19] Janiesch, C., Zschech, P., Heinrich, K. (2021). Machine learning and deep learning. Electronic Markets, 31(3): 685-695. https://doi.org/10.1007/s12525-021-00475-2

[20] Herbert, S., Sengupta, A. (2018). Using reinforcement learning to find efficient qubit routing policies for deployment in near-term quantum computers. arXiv preprint arXiv:1812.11619. http://arxiv.org/abs/1812.11619

[21] Kalouptsoglou, I., Siavvas, M., Ampatzoglou, A., Kehagias, D., Chatzigeorgiou, A. (2023). Software vulnerability prediction: A systematic mapping study. Information and Software Technology, 107303. https://doi.org/10.13140/RG.2.2.25265.97127

[22] Ardali, D.R. (2019). Incremental learning framework for mass personalisation of skin-care products. Doctoral dissertation, University of Portsmouth.

[23] Denkena, B., Wichmann, M., Noske, H., Stoppel, D. (2023). Boundary conditions for the application of machine learning based monitoring systems for supervised anomaly detection in machining. Procedia CIRP, 118: 519-524. https://doi.org/10.1016/j.procir.2023.06.089

[24] Stiawan, D., Idris, M.Y.B., Bamhdi, A.M., Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access, 8: 132911-132921. https://doi.org/10.1109/ACCESS.2020.3009843

[25] "Google Colab." Accessed: Nov. 08, 2023. [Online]. Available: https://research.google.com/colabora80t9ory/.