Cross-layer Federated Heterogeneous Ensemble Learning for Lightweight IoT Intrusion Detection System

Suzan Hajj*, Joseph Azar†, Jacques Bou Abdo‡, Jacques Demerjian§, Abdallah Makhoul† and Dominique Ginhac*
* ImViA, Université de Bourgogne Franche-Comté, 21078 Dijon, France
† Femto-St Institute, UMR 6174 CNRS, Université de Franche-Comté, France
‡ School of Information Technology, University of Cincinnati, Cincinnati, OH 45221 USA
§ LaRRIS, Faculty of Sciences, Lebanese University, Fanar, Lebanon

***Abstract*—This paper presents a heterogenous federated ensemble model for intrusion detection system, employing a semi-supervised novelty detection technique - the baseline Kmeans. The technique learns normal traffic from baseline data and utilizes the Mahalanobis distance to detect anomalous packets. To mitigate the false-positive rate inherent in anomaly-based intrusion detection system, we propose an ensemble approach that integrates local novelty detection models dedicated to each worker in both weighed and voting-based strategies. The federated design augments each worker's detection capability without increasing the false positive rate. Our extensive experiments showcase the system's robustness and adaptability over traditional standalone IDS, with marked improvements in precision, recall, and F1-score under varying sampling rates. We made this project's code publicly available on Github for reproducibility.**

***Index Terms*—Federated learning, Lightweight Internet of Things, Intrusion Detection, Lightweight Sampling, Anomaly Detection, Ensemble Learning**

I. Introduction

Distributed and federated Intrusion Detection Systems (IDS) have emerged as subjects of significant interest within the realm of Internet of Things (IoT) security, highlighting the need for more intricate and robust protection mechanisms [1]. However, extending distributed and federated IDS to lightweight IoT remains a gap.

In our previous research [2], we proposed a novel cross-layer Intrusion Detection System (IDS) that employs a cluster-based sampling technique. This approach ensures comprehensive data representation, including the accommodation of rare subgroups, effectively minimizing the sampling error due to data variance. Implementing IDS on such sampled data significantly reduces memory usage and energy consumption due to computation and data transmission. Building on this foundation, we further extended the proposed sampling algorithm to incorporate a federated intrusion detection system using a semi-supervised machine learning model [3]. Despite demonstrating a promising ability to detect malicious packets, this distributed approach encountered a notable increase in the false positive rate following each aggregation and parameters redistribution step by the centralized node.

The high false positive rate represents a persistent challenge within the domain of anomaly detection, particularly in anomaly-based intrusion detection [4]. As the model learns to represent normal traffic patterns,it faces an inherent challenge to cover all possible benign behaviors in the learning dataset. Consequently, benign data that deviates from the learnt distribution, or were not present during the training phase, may be misclassified as an attack.

Recent research works suggest ensemble learning as a promising strategy to mitigate false positives in anomaly detection systems [5], [6]. A notable example is the two-stage architecture proposed in [4], where the IDS initially applies an unsupervised methodology for attack detection, followed by supervised learning for classifying the attacks and reducing false positives. However, it is critical to consider, as noted in [7], that not all problem types are conducive to unsupervised learning, particularly when data lack distinct clusters. Furthermore, supervised techniques might be infeasible if appropriately labeled data is unavailable. Given these limitations, our research presupposes the unavailability of a fully labeled dataset, instead allowing for a minimal quantity of benign data at the outset. Consequently, we frame the problem as a semi-supervised novelty detection task. This paper's objective is to present a lightweight, semi-supervised, federated IDS for IoT, designed to operate following a sampling layer and employing a heterogeneous ensemble learning strategy. This approach not only efficiently detects malicious packets but also ensures that the false positive rate does not increase over time, thereby providing a more reliable and sustainable solution for IoT security.

This paper is structured as follows: Section II reviews relevant literature, focusing on cross-layer federated learning and ensemble learning in lightweight IoT IDS. Section III offers a comprehensive overview of the proposed work, starting with an introduction to the semi-supervised novelty detection algorithm, derived from the K-means algorithm, and referred to as baseline-Kmeans. This section also delves into the proposed ensemble approach. Section IV illustrates the

experimental setups and details the construction of the utilized ensemble. Section V delineates the experiments conducted, interprets the outcomes, and emphasizes how the proposed methodology fulfills the study's objectives. Finally, Section VI concludes the paper.

II. Related Work

A. Cross-layer Federated Learning in Lightweight IoT IDS

Internet of Things (IoT) nodes with their ad-hoc heterogeneous distributed connections are more vulnerable to network attacks than conventional nodes operating within the trust boundaries of secured perimeter networks. This makes Intrusion Detection Systems (IDS) essential for maintaining the security of IoT nodes. IDS in the context of IoT has received considerable literature attention [8], [9]. More specifically, federated learning in IoT IDS has been extensively studied and benchmarked in literature such as in [1], [10], [11], [12], [13], [14]. However, those IDSs are rarely customized to cater for lightweight IoT nodes, which are constrained on memory, computational resources and energy. One approach is adapting existing IoT IDS to lightweight IoT nodes to minimize the number of screened packets and thus reducing the needed computations. This is done through the introduction of a sampling layer (as figure 1 shows), which aims at creating, from the population of received packets, a smaller but equally representative sample. In previous studies, we benchmarked all sampling algorithms under the constraints of lightweight IoT [15] and proposed a new sampling algorithm, especially designed for lightweight IoT called cluster-based sampling algorithm [2]. Another approach is specifically designing lightweight IDS which is rarely investigated in literature.

We extended the first approach in cross-layer federated learning in the context of lightweight IoT IDS [3] as figure 1 shows. This extension is based on the federated Baseline KMeans algorithm, which is a distributed and privacy-preserving version of the KMeans algorithm designed for intrusion detection in IoT applications [3]. The federated version of the algorithm can train a clustering model without transmitting sensitive data, thereby preserving data privacy. In the federated version, multiple IoT devices participate in the training process and compute their own local statistics, including means and distances. The statistics are then transmitted to a central coordinator for aggregation. The coordinator updates the baseline and anomalous centroids based on the merged statistics from the IoT devices and computes a new threshold.

Cross-layer federated learning [3] involves four key steps:

- **Initialization**: the coordinator initializes the k-means clustering model with a fixed number of clusters and shares cluster statistics between workers. Each node uses the global representation of the benign cluster and the distance threshold to detect anomalies in its local data subset. Data points with distances above the threshold are classified as anomalies, while data points with distances below the threshold are classified as benign.
- **Local clustering**: each node uses its own local k-means clustering model on its own sampled data. Each node computes the Mahalanobis distance to the benign centroid for each of its local data points.
- **Share cluster statistics**: each node shares the cluster statistics, such as the cluster centroids and the distances to the benign centroid, with the other nodes through the coordinator, but not the data.
- **Merge statistics**: the coordinator merges the cluster statistics from each node to create a global representation of the benign and anomalous clusters. This can be done by averaging the cluster centroids and defining a classification threshold for the Mahalanobis distances based on the global representation of the benign cluster. The updated global model can then be shared with the worker nodes for further local training.

B. Ensemble Learning and Data Streams

Machine learning methods try to label a data point by finding one best hypothesis to explain it, but ensemble learning builds a set of hypotheses to be voted on [16]. Since each of the hypotheses included in the voting set is found using a machine learning method, ensemble learning is usually more accurate than any individual machine learning method [17]. Ensemble is found, by Dong et al. [18], to be extremely suitable for complex, imbalanced, high-dimensional and noisy data.

In the context of data streams, machine learning algorithms struggle to learn from a constrained sliding window leading to a sub-optimal model [19]. Ensemble learning is described, by Polikar [20], to be primarily used to reduce the likelihood of an unfortunate selection of a poor one. Sun et al. [21] proposed a class-based ensemble approach to detect gradually emerging or disappearing data classes. Van Rijn et al. [22] proposed a heterogeneous ensemble learning framework that performed competitively in comparison to state-of-the-art ensemble techniques but over a wide range of data streams. It is considered heterogeneous since, contrary to most dynamic data stream ensembles that rely on only one type of base-level classifier, the proposed ensemble relies on multiple different classifiers. Zhang and Jin [23] proposed a strategy to automatically configure the ensemble learning algorithm by adaptively distinguishing sensible classifiers and showed that this strategy outperform static configurations. Individual ensemble learning techninques have been proposed to deal with specific data stream problems such as concept drift [24], [25], [21], [26], imbalance [27], [28] and noise [29], [26].

Ensemble learning in the context of data streams has been extensively surveyed in the literature [30]. Zang et al. [31] compared incremental and ensemble learning with respect to accuracy and time efficiency. They found that ensemble learning is more stable than incremental learning and outperformed for smaller data chunks. Li et al. [32] surveyed ensemble learning in the context of data streams as "Extreme Learning Machine" variant. Others have introduced ensemble learning in their surveys on machine learning for data steams [33], [34].

C. Ensemble Learning in IoT IDS

Lama and Tim [35] systematically surveyed the use of ensemble learning in IDS. They showed that random forest

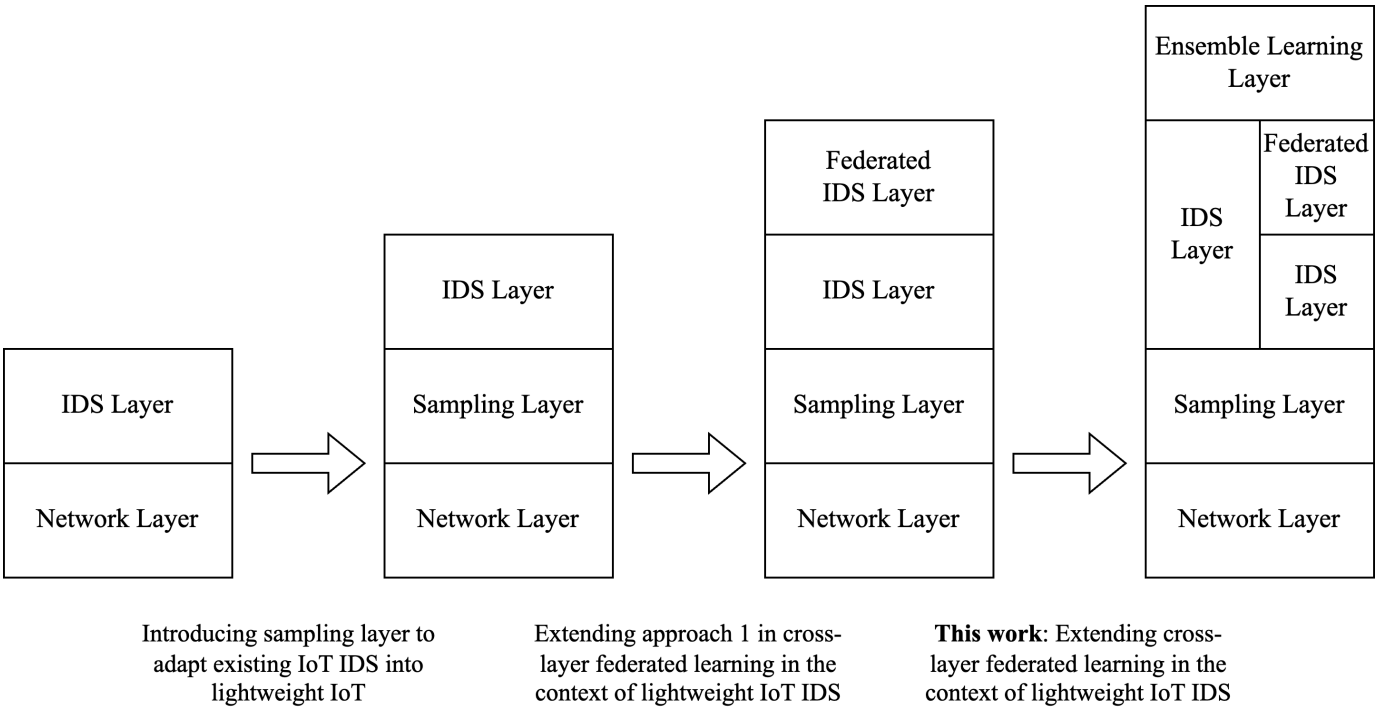


Fig. 1: Evolution in lightweigh IoT IDS research. The first step is adapting existing IoT IDS into lightweight IoT through the introduction of a sampling layer with small sampling ratio. The second step is introducing federated learning allowing for even lower sampling ratios. The last evolution, proposed in this work, is benefiting from ensemble learning and pushing sampling ratios to record lows.

is specially used under ensemble learning for IDS. A more detailed listing was provided where homogeneous ensemble used random forest, bagging, and boosting while majority voting and stacking architecture were used for heterogeneous ensembles. Illy et al. [36] proposed an ensemble learning mechanism for fog-to-things environment resulting in reduced classification latency and elevated accuracy. In the same direction, Verma and Ranga [37] proposed ELNIDS, an ensemble learning IDS for "Low-Power and Lossy" IoT networks. The authors used a four-model ensemble of bagged trees, boosted trees, subspace discriminant and RUSBoosted trees. This work is aligned with Abu Al-Haija and Al Badawi [38] and Mohy-Eddine et al. [39].

Alhowaide, Alsmadi and Tang [40] are aligned with [23] in using automatic Model Selection Method (MSM) for automatically configuring a heterogeneous set of classifiers. Abu Alghanam et al. [41] used ensemble learning for proposing an improved PIO feature selection algorithm, called LS-PIO, for IoT IDS. It used K-Means at the pre-processing stage to reduce the running time and making it fit for lightweight IoT. This work is aligned with Gopalakrishnan and Purusothaman[42]. Abu Al-Haija and Al-Dala'ien [43] proposed ELBA-IoT, an ensemble learning model for botnet attack detection in IoT networks using AdaBoosted, RUSBoosted, and bagged trees. Hazman et al. [44] proposed lIDS-SIoEL, an intrusion detection framework for IoT-based smart environments based on Ensemble Learning using AdaBoost, Boruta, mutual information and correlation. The methods surveyed in this section are summarized in table I.

Reference	Ensemble learning set
Illy et al. [36]	Random Forest, Bagging Classifier, AdaBoost and Voting
Verma and Ranga [37]	bagged trees, boosted trees, subspace discriminant and RUSBoosted trees
Abu Al-Haija and Al Badawi [38]	bagged trees, ensemble subspace kNN (ESK), RUSBoosted trees, shallow neural network (SNN), bilayered neural network (BNN and logistic regression kernel (LRK)
Mohy-Eddine et al. [39]	isolation forest (IF) and pearson's correlation coefficient (PCC)
Alhowaide, Alsmadi and Tang [40]	logistic regression, random forest, decision tree, gradient boosting, bagged tree, gaussian naive bayes, adaboosted, knn, bernoulli naive bayes, multi-layer perceptron, stochastic gradient descent and support vector machines
Abu Alghanam et al. [41]	support vector machines, isolation forest, local outlier factore and K-means
Gopalakrishnan and Purusothaman[42]	deep neural network (DNN), random forest, and AdaBoost
Abu Al-Haija and Al-Dala'ien [43]	AdaBoosted, RUSBoosted, and bagged trees
Hazman et al. [44]	AdaBoost, Boruta, mutual information and correlation

TABLE I: List of IoT IDS using ensemble learning and the set of used machine learning algorithms

More thorough detailing of ensemble learning in IoT IDS

can be seen in [45], [46], [47].

III. Proposed Federated Ensemble IDS

In this paper, we present a novel federated semi-supervised ensemble novelty detection technique designed for intrusion detection systems (IDS) in IoT networks. Our approach addresses the challenge commonly faced in real-world applications where only a limited amount of labeled regular or benign traffic data is available.

While entirely unsupervised IDS methods are beyond the scope of this work, we propose a modification to the unsupervised KMeans technique, transforming it into a semi-supervised method called Baseline KMeans [3], [48]. This technique learns a boundary approximating the baseline observations distribution. Let $X = x_1, x_2, \ldots, x_n$ be the initial baseline observations, and $x_i \in \mathbb{R}^d$ represent each data point in the d-dimensional feature space. We apply the Baseline KMeans algorithm to obtain a centroid $C \in \mathbb{R}^d$ for the baseline data.

The Mahalanobis distance $D_M(x_i, C)$ between a data point x_i and the centroid C is defined as:

$$D_M(x_i, C) = \sqrt{(x_i - C)^T S^{-1} (x_i - C)}$$

where S is the sample covariance matrix of the baseline data.

Let p be the chosen percentile (e.g., $p = 90\%$). The threshold τ is computed as:

$$\tau = \text{percentile}(D_M(x_1, C), \ldots, D_M(x_n, C), p)$$

For any new observation x', the Baseline KMeans classifies it as benign or anomalous based on the following decision rule:

$$\begin{cases} \text{benign}, & \text{if } D_M(x', C) \leq \tau \\ \text{anomalous}, & \text{if } D_M(x', C) > \tau \end{cases}$$

If further observations lie within this subspace, they are classified as benign traffic. The threshold is determined by the percentile of Mahalanobis distances between each baseline data point and the centroid. Observations outside the boundary are considered abnormal, possibly indicating an attack.

A. Motivation

Since our approach is built upon KMeans, we employ a distance metric to the benign/baseline centroid to determine whether an observation originates from the baseline population or is an outlier. This method effectively achieves a high true positive rate by minimizing the boundary to the baseline centroid, which is essential in security contexts where attacks may be intolerable. However, the approach is also characterized by a high false-positive rate. The main motivation for the proposed method is the observation that, during the merging operation in which the worker transmits its statistics to the coordinator and subsequently receives aggregated statistics, the worker's recall increases over time while its precision decreases. This trend can be attributed to the IDS rejecting a greater number of packets or data points than before. Consider the following equations summarizing the merge operation and threshold recalculation:

$$\begin{aligned} M_{\text{baseline}} &= \frac{1}{2}(M_{\text{coordinator,baseline}} + M_{\text{worker,baseline}}) \\ M_{\text{anomalous}} &= \frac{1}{2}(M_{\text{coordinator,anomalous}} + M_{\text{worker,anomalous}}) \\ S^{-1} &= \text{pinv}(S_{\text{coordinator,baseline}} + 0.001 \cdot I) \\ D(x_i, M_{\text{baseline}}) &= \sqrt{\sum_{i=1}^{n} (x_i - M_{\text{baseline}})^T S^{-1} (x_i - M_{\text{baseline}})} \\ D_{\text{min}} &= \min(D(x_i, M_{\text{worker,anomalous}})) \\ D' &= \{D(x_i, M_{\text{baseline}}) \mid \text{where } D(x_i) < D_{\text{min}}\} \\ \tau' &= \text{percentile}(D', p) \end{aligned}$$

Here, M_{baseline} and $M_{\text{anomalous}}$ represent the updated baseline and anomalous means, respectively, after the merge operation. S^{-1} is the pseudoinverse of the sum of the baseline covariance matrix and a small regularization term. $D(x_i, M_{\text{coordinator,baseline}})$ denotes the Mahalanobis distance of point x_i from the coordinator's baseline mean, while D_{min} represents the minimum distance of the worker's anomalous data points to the baseline mean. D' is the set of distances that are less than D_{min}, and τ' is the updated threshold calculated based on the specified percentile of distances in D'.

As the merging operation progresses, the coordinator's calculated threshold typically becomes more conservative, resulting in a greater number of data points being rejected. This occurs because the recalculated threshold considers both the specified percentile and the minimum distance between a worker's anomalous data points and the baseline mean. Therefore, all distances exceeding this minimum distance are considered abnormal.

To address this issue, we propose an ensemble-based approach that aids in reducing the false-positive rate. Figure 2 illustrates the proposed cross-layer federated ensemble learning in the context of lightweight IoT IDS.

B. Ensemble IDS

In this paper, we propose an ensemble learning approach to enhance the precision of the Baseline K-means model, especially when it predicts that a packet is anomalous. This approach leverages two other local models, both of which are semi-supervised novelty detection models, private to each worker, and trained on a small batch of benign data. The rationale behind this approach is to exploit the conservative nature of the Baseline K-means model, which is already good at detecting benign packets.

Let y_{bkmeans}, y_{pred1}, and y_{pred2} denote the predictions of the Baseline K-means model, local model 1, and local model 2, respectively. Then, the ensemble learning output, denoted as y_{el}, is computed as in algorithm1:

Here, the function $majority\_vote$ returns 0 (benign) if both local models predict that the packet is benign, and 1 (anomalous) otherwise. Thus, when the Baseline K-means

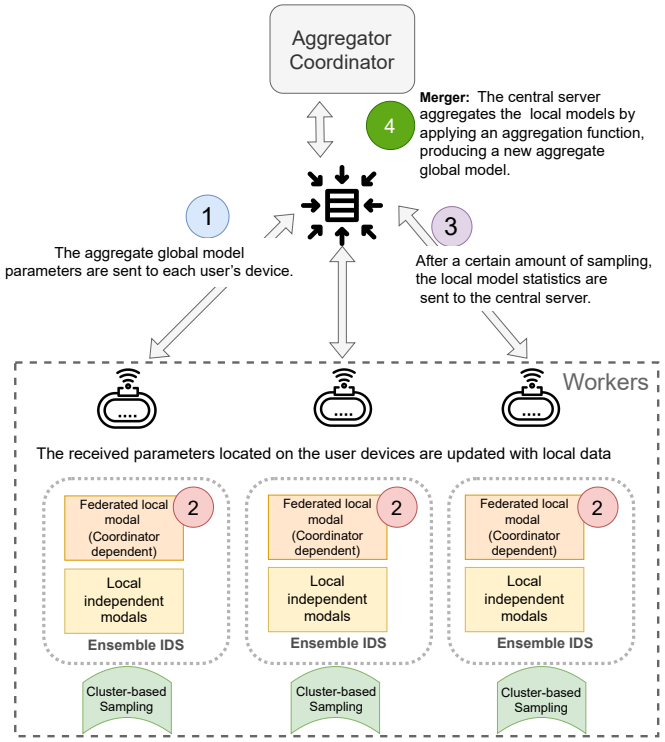


Fig. 2: The figure illustrates a federated intrusion detection approach for IoT networks. A BaselineKMeans coordinator initializes with baseline data and shares statistics with worker nodes. Each worker uses a cluster-based sampling algorithm to minimize data processing and labels data points with the coordinator's statistics. Local, independent models assist WorkerKmeans IDS in classification. Workers periodically send their statistics to the coordinator, which updates the global model and shares updated statistics.

model predicts that a packet is anomalous, we take the opinion of the two local models. If at least one of them predicts that a packet is anomalous, then the ensemble learning output also considers it anomalous. On the other hand, if both local models predict that a packet is benign, we overturn the Baseline K-means decision, considering the packet benign. In the proposed ensemble learning strategy for improving the Baseline KMeans method's accuracy, we incorporate voting-based and weight-based ensemble learning techniques. When classifying a data point as benign (0), the ensemble model prioritizes the Baseline KMeans prediction in its decision-making process. In this scenario, the ensemble learning technique exhibits weighted characteristics, as the Baseline KMeans model carries a greater weight in the decision-making procedure.

Nonetheless, the method retains voting-based characteristics when the Baseline KMeans prediction is anomalous (1). In this instance, the final label is determined by the majority vote of the Baseline KMeans and two other local models (Algotithm 1). As a result, ensemble learning effectively integrates the advantages of voting-based and weight-based techniques, resulting in a more precise and robust classification

Algorithm 1 Ensemble Learning IDS

function MAJORITY_VOTE($vote1, vote2, vote3$)
  if $(vote1 + vote2 + vote3) \geq 2$ **then**
    return 1
  else
    return 0
  end if
end function
for $i \leftarrow 1$ to n **do**
  if $y_{\text{bkmeans}}(i) = 0$ **then**
    $y_{\text{el}}(i) \leftarrow 0$
  else
    $y_{\text{el}}(i) \leftarrow \text{majority\_vote}(y_{\text{bkmeans}}(i), y_{\text{pred1}}(i), y_{\text{pred2}}(i))$
  end if
end for

of normal and abnormal data points.

This hybrid ensemble learning method is anticipated to improve the overall performance of the intrusion detection system, as it combines the benefits of both voting-based and weight-based ensemble learning techniques to improve the Baseline KMeans model's precision.

IV. Experimental Setup

The techniques discussed are implemented in Python, made publicly available on Github [48], and evaluated on the NSL-KDD dataset [49]. The NSL-KDD dataset is a widely-used and enhanced version of the original KDD Cup 1999 dataset [50] developed specifically for evaluating intrusion detection systems (IDS). It includes both regular (benign) and malicious (intrusion) instances of network traffic data. The NSL-KDD dataset tackles some of the issues common to the original KDD dataset, such as duplicated records and data imbalance, making it an improved choice for evaluating the effectiveness of intrusion detection techniques.

In our experiments, we compared the Baseline KMeans to several semi-supervised novelty detection techniques to determine their performance. Local Outlier Factor (LOF), Gaussian Mixture Model (GMM), One-Class Support Vector Machine (SVM), Isolation Forests, Minimum Covariance Determinant (MCD), KNN Detector, Kernel Density Estimation (KDE), and Shallow Autoencoder are among the methods considered. We aimed to develop an ensemble learning technique by assigning each worker two semi-supervised novelty detection models as local models. These models would aid the Baseline KMeans in determining whether a data point predicted as an attack is, in fact, an attack, thereby reducing the likelihood of false positives. In order to accomplish this, we tested numerous combinations of the methods above, as shown in Table II. The ensemble of Baseline KMeans, K Nearest Neighbor (KNN) detector, and Kernel Density Estimation (KDE) produced the highest F1-score (0.90 ± 0.01) and F2-score (0.93 ± 0.01) compared to the other ensembles. With an F1-score of 0.90, the ensemble composed of Baseline KMeans, a shallow autoencoder, and KDE also demonstrated efficacy. These results

TABLE II: Comparison of Baseline KMeans and various Ensemble combinations: Average performance metrics and standard deviations.

	Precision	Recall	F1-score	F2-score
Baseline Kmeans	0.76 ± 0.02	0.96 ± 0.01	0.85 ± 0.01	0.92 ± 0.01
Ensemble				
Bkmeans-AE-IF	0.85 ± 0.02	0.89 ± 0.01	0.87 ± 0.01	0.89 ± 0.01
Bkmeans-AE-KDE	**0.86 ± 0.02**	0.94 ± 0.01	**0.90 ± 0.01**	0.92 ± 0.01
Bkmeans-AE-KNN	0.85 ± 0.02	0.90 ± 0.01	0.87 ± 0.01	0.89 ± 0.01
Bkmeans-AE-SVM	**0.86 ± 0.02**	0.89 ± 0.01	0.88 ± 0.01	0.88 ± 0.01
Bkmeans-IF-KDE	0.83 ± 0.02	0.95 ± 0.01	0.89 ± 0.01	0.92 ± 0.01
Bkmeans-IF-KNN	0.83 ± 0.02	0.90 ± 0.01	0.87 ± 0.01	0.89 ± 0.01
Bkmeans-IF-LOF	0.82 ± 0.02	0.94 ± 0.01	0.88 ± 0.01	0.91 ± 0.01
Bkmeans-KNN-KDE	0.85 ± 0.02	**0.96 ± 0.01**	**0.90 ± 0.01**	**0.93 ± 0.01**
Bkmeans-LOF-SVM	0.85 ± 0.02	0.94 ± 0.01	0.89 ± 0.01	0.92 ± 0.01

indicate that combining Baseline KMeans with KNN and KDE or a shallow autoencoder can improve the model's overall performance in detecting anomalous data points. We can also observe that adding a one-class SVM to an ensemble improves performance.

It can also be noted from Table II that we excluded Gaussian Mixture Models and Minimum Covariance Determinant (MCD). This was done due to their computational complexity, which renders them unsuitable for deployment on resource-constrained devices such as microcontrollers or IoT devices. GMM requires calculating and storing covariance matrices for each Gaussian component, which necessitates a considerable amount of memory. Based on a subset of data points with the smallest determinant, MCD computes the covariance matrix, which requires a computationally intensive search for every possible subset of a given size.

The following section considers the ensemble consisting of the federated baseline KMeans and local KNN and KDE models.

V. Results and Analysis

A. Semi-supervised novelty detection for intrusion detection

This section compares our proposed ensemble IDS with a number of well-known novelty detection techniques. These novelty detection techniques and our proposed method have one thing in common: they all utilize a portion of benign data to discover the underlying patterns of normal traffic. We initially presented each method with approximately 4,000 benign data points to enable it to learn the normal traffic patterns. In order to categorize and classify the data, novelty detection techniques were applied to each sliding window of 1,000 data points.

Figure 3 demonstrates conclusively the superior performance of the ensemble learning approach, specifically the combination of Federated Baseline Kmeans, K-Nearest Neighbors (KNN), and Kernel Density Estimation (KDE) when compared to Federated Baseline Kmeans and other novelty detection techniques when applied independently.

Observations indicate that the ensemble model enhanced the precision score from 0.76 for the Baseline Kmeans model to 0.85, thereby effectively reducing the rate of false positives. This improvement in accuracy demonstrates the ensemble

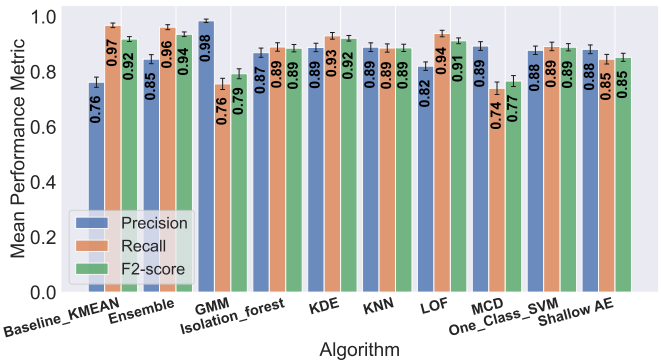


Fig. 3: Performance comparison of semi-supervised novelty detection algorithms for intrusion detection The bar plot depicts the precision, recall, and Fβ-score (beta = 2) for each algorithm, highlighting the ability of the proposed Ensemble to prioritize the rate of true positives (highest recall) while maintaining competitive precision in comparison to Baseline Kmeans alone.

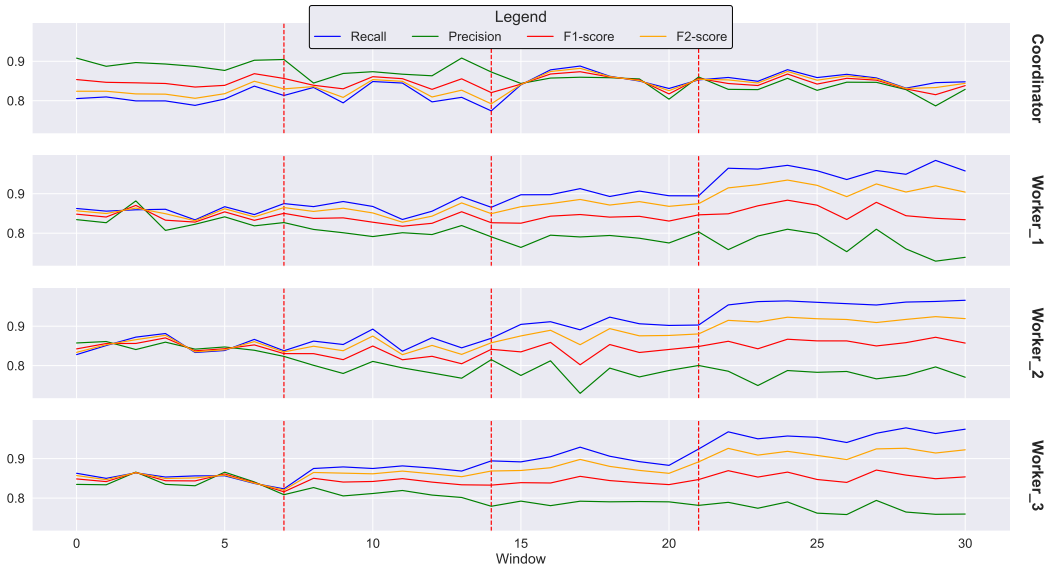
model's superior ability to correctly identify authentic instances of attacks, thereby reducing the risk of misclassifying benign data points as malicious. This enhancement is crucial in practical applications where a high rate of false positives can result in unnecessary costs and resource waste.

The ensemble learning method maintained a high recall value of 0.96, virtually on par with the Baseline Kmeans model, indicating its capacity to identify the most positive cases accurately. In the context of anomaly detection, a high recall score is essential, as it is essential to identify as many actual attacks as feasible to ensure system security.

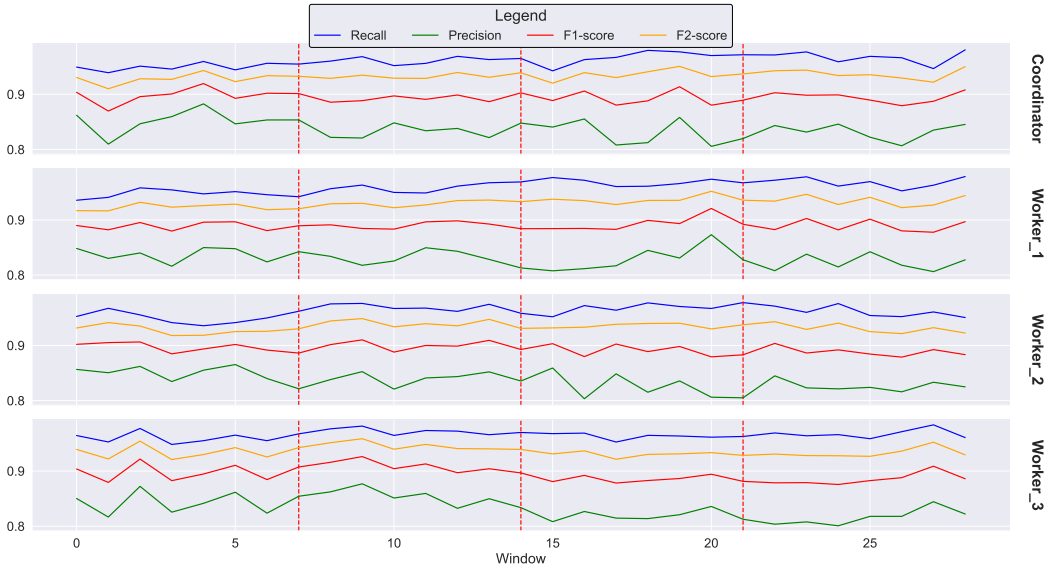
Compared to other novelty detection techniques, the ensemble approach demonstrated superior or comparable performance. While Gaussian Mixture Models (GMM) and Minimum Covariance Determinant (MCD) exhibited greater precision, their recall scores were lower, indicating a higher rate of false negatives. Other techniques, including Isolation Forest, KNN, Local Outlier Factor (LOF), One Class SVM, and Shallow AE, had lower F2 scores than the ensemble model. Note that the F2 score is particularly suitable in intrusion detection as it gives more weight to Recall (the ability to identify actual attacks correctly), which is critical in minimizing potential harm and ensuring system security.

B. Federated ensemble IDS

In this section, we simulate a federated intrusion detection system (IDS) composed of a coordinator and three workers. Initially, the coordinator learns a baseline model using 100 benign data points and shares its statistics with the workers. The coordinator actively participates with the workers in processing network traffic data. We use the NSL-KDD dataset (approximately 120,000 rows), resulting in roughly 30 epochs for our simulation. Each epoch involves processing 1,000 data points per entity, totaling 4,000. This allows us to assess the baseline K-means algorithm's effectiveness in a distributed environment where all parties process and learn from the



(a) Federated baseline Kmeans without local ensemble.



(b) Federated baseline Kmeans with local ensemble.

Fig. 4: Evolution of performance metrics for the coordinator and three workers over 30 epochs, each comprising a window of 1,000 data points. The vertical dashed red lines denote points where merge operations were conducted.

dynamic network traffic data. The simulation runs until fewer than 1,000 elements remain in both the normalized data and the ground-truth labels. Performance metrics are calculated and recorded in a Python dataframe after each epoch.

Our simulation study explored two distinct scenarios. In the first scenario, we employed the federated baseline K-means algorithm without the integration of an ensemble. This scenario included three merging operations. The coordinator underwent its primary training phase, following which it shared its initial statistical outcomes with the workers. The second scenario also involved three merging operations but differed by incorporating our proposed federated ensemble method.

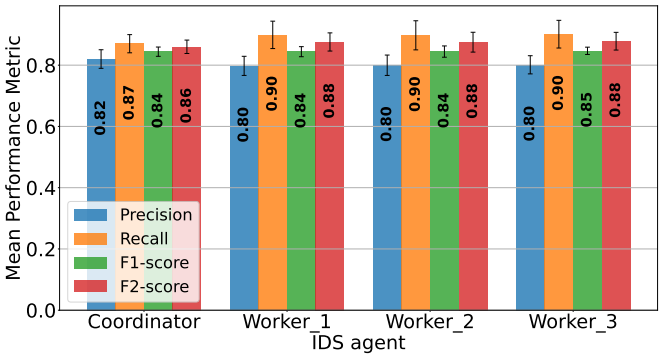
The results from the first scenario (Figure 4-a) demonstrate a performance improvement in both the coordinator and the workers through the federated baseline Kmeans approach. The Federated baseline Kmeans approach, with its distributed data processing, has positively influenced the performance metrics over the various windows. Starting with the baseline data points, the coordinator and the workers' initial performance shows a reasonably high precision and recall, indicating a good ability to correctly classify true positives and effectively

identify actual positives from the total predicted positives. This was reflected in the initial F1 and F2 scores, which are measures of the test's accuracy considering both precision and recall. As the experiment progresses, the coordinator and workers' performances generally improve. The F2 score, which gives more weight to recall than precision, shows a significant increase in most of the iterations. This suggests the system's ability to correctly identify true positives from the total actual positives (recall) has improved over time. Nevertheless, each merge operation has performance fluctuations across the precision and F1 score metrics. Notably, a negative trend is observed in precision throughout the windows/epochs, suggesting an increasing false-positive rate with each merge. This can be attributed to baseline Kmeans' prioritization of the true-positive rate over the false-positive rate. After each merging operation, the workers become more strict regarding rejecting distances greater than the dynamically evolving threshold. Overall, these results underscore the potential of federated learning in intrusion detection, with the system demonstrating a generally improving trend in performance over time. However, the drawback of increasing false positives with time in the federated baseline Kmeans approach should be noted.

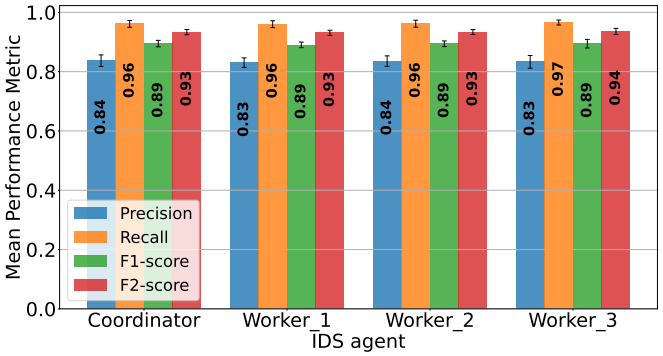
Implementing an ensemble approach, which combines federated baseline k-means, KNN, and KDE, mitigates the shortcomings inherent in the standalone baseline k-means technique. As illustrated in the second scenario (Figure 4-b), this ensemble strategy often results in a noticeable enhancement in precision for all participating workers and the coordinator, as opposed to the outcomes achieved with a non-ensemble strategy. Importantly, it is evident that each worker attains a precision score surpassing 0.8, a clear contrast to the precision decline observed below 0.8 in the non-ensemble approach. While some temporal fluctuations in precision persist, the trend remains predominantly stable, a clear divergence from a deteriorating trend. Moreover, there is a noticeable increase in recall for the ensemble approach, especially regarding the coordinator's results. The F1 score has improved slightly in the ensemble approach, and the F2 score, which emphasizes recall, is also consistently higher in the ensemble approach compared to the non-ensemble method, further demonstrating the efficacy of the ensemble strategy. The differences in the average performance metrics for the ensemble and non-ensemble approaches are illustrated in Figure 5.

C. Cross-layer federated learning

In this section, we explore the concept of cross-layer federated IDS, where a cluster-based sampling technique proposed in our previous work [2] is applied before intrusion detection. We consider a scenario involving a coordinator and a worker with two sampling rates: 0.60 and 0.20. The coordinator trains on a baseline data, exports its statistics to the worker, and then the worker utilizes the same statistics for 10 epochs, processing 5,000 data points per epoch. Two merging operations are added at epochs 2 and 6.



(a) Average performance metrics for the federated baseline Kmeans without local ensemble after 30 epochs.



(b) Average performance metrics for the federated baseline Kmeans with local ensemble after 30 epochs.

Fig. 5: Average performance metrics for the coordinator and three workers with three merging operations.

In Intrusion Detection Systems (IDS) context, the sampling rate signifies the volume of data retained in memory for analysis. A high sampling rate, such as 0.60, equates to a larger data retention, while a lower rate, exemplified by 0.20, results in fewer data being preserved. Retaining historical data in memory can aid in detecting and preventing cyber threats, such as replay attacks, where an attacker intercepts, artificially delays, or retransmits valid data to deceive the recipient system into performing unauthorized operations. Our analysis of the IDS performance metrics (Precision, Recall, and F1-score) across varying sampling rates (Figure 6) reveals a slight degradation in performance at a lower sampling rate (0.20) as compared to a higher rate (0.60). This outcome is predictable, considering that a reduced volume of data equates to less information available to the IDS for making accurate decisions. For instance, at the third window, following the initial merge operation, there is a decrease of approximately 0.05 in precision without ensemble when the sampling rate is reduced from 0.60 to 0.20. Interestingly, the diminished sampling rate does not adversely impact the recall metric. The IDS retains its ability to identify attack packets with comparable efficiency as it exhibited at the higher sampling rate of 0.60. In contrast, the ensemble methodology consistently demonstrates superior performance metrics across all sampling rates and

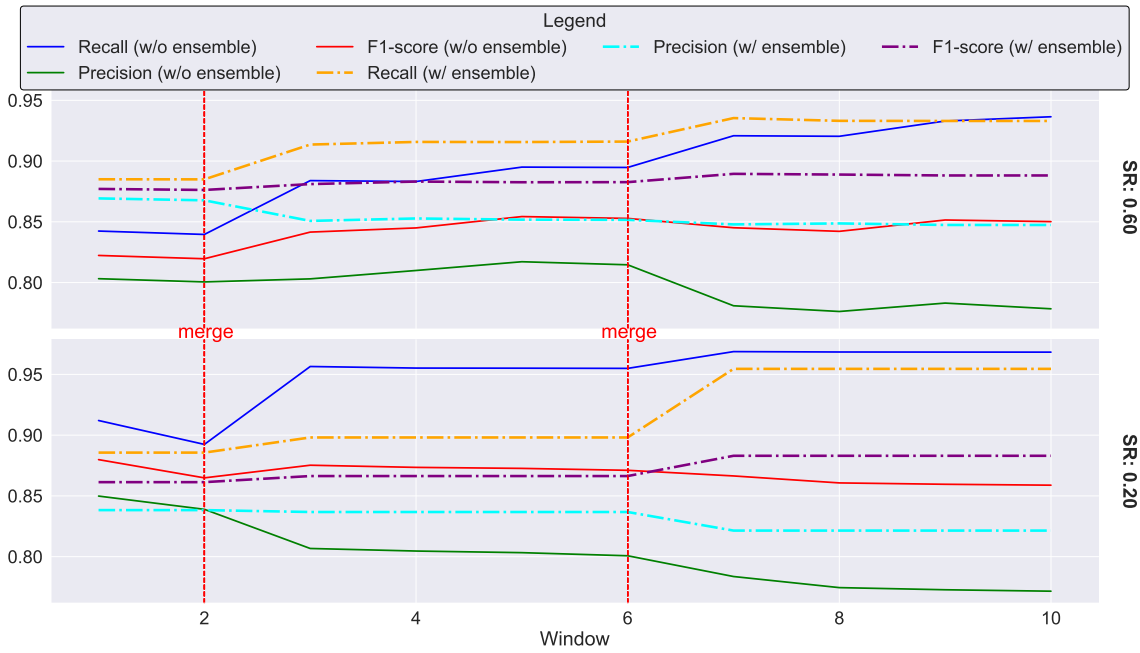


Fig. 6: Performance evaluation of Intrusion Detection Systems (IDS) under various sampling rates and Ensemble Learning conditions. This figure illustrates the comparative performance of IDS with and without ensemble learning for two distinct sampling rates, 0.60 and 0.20, across ten successive windows. The three performance metrics considered are Precision, Recall, and F1-score, represented by distinct color-coded lines. Solid lines indicate the performance without ensemble learning, while dashed lines represent the ensemble learning scenario. The vertical dashed red lines denote points where merge operations were conducted. It can be observed that the ensemble approach consistently outperforms the non-ensemble condition across all metrics and both sampling rates. The performance slightly degrades with the reduction in sampling rate, especially for the non-ensemble condition, further highlighting the robustness of the ensemble approach.

windows compared to non-ensemble methods. This superior performance can be attributed to the inherent characteristics of ensemble learning, which combines multiple models to make a decision, typically resulting in more robust and accurate outcomes. For example, at the first window with a sampling rate of 0.60, the precision utilizing the ensemble approach is 0.869, notably higher than the precision of 0.803 achieved without ensemble. This trend of enhanced performance with the ensemble approach is consistently observed across all windows and both the sampled rates.

VI. Conclusion

This research presented a novel Intrusion Detection System (IDS) employing a heterogeneous federated ensemble approach, combining weighed and voting-based strategies to enhance anomaly detection. The baseline K-means algorithm, supported by assisting local models, showcased superior accuracy over standalone IDS under varying sampling rates (0.60 and 0.20). Integrating local models into the decision-making process has proven to be a significant advancement, enhancing the system's capability to identify anomalous packets accurately. When the baseline K-means predicts a benign class, the system relies solely on this prediction, showcasing its weighed aspect. However, when a potential anomaly is detected, a voting process amongst local models is initiated, thus implementing a voting-based strategy. This fusion of strategies enhances the system's reliability and precision in identifying cyber threats. Moreover, the federated design of our approach allows for the periodic statistics merging and sharing of local models, effectively increasing the detection capacity of each worker. Importantly, this does not lead to a higher false positive rate due to the ensemble's ability to validate and consolidate predictions. In conclusion, our research underlines the effectiveness of heterogeneous federated ensembles in IDS, offering promising advancements for network security in IoT applications.

References

[1] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, and A. Skarmeta, "Evaluating federated learning for intrusion detection in internet of things: Review and challenges," *Computer Networks*, vol. 203, p. 108661, 2022.

[2] S. Hajj, R. El Sibai, A. Barada, J. Bou Abdo, J. Demerjian, C. Guyeux, A. Makhoul, and D. Ginhac, "Cluster-based sampling algorithm for lightweight iot intrusion detection system," in *2022 20th International Conference on Security and Management*. Springer, 2022.

[3] S. Hajj, J. Azar, J. Bou Abdo, J. Demerjian, C. Guyeux, A. Makhoul, and D. Ginhac, "Cross-layer federated learning for lightweight iot intrusion detection system," *IEEE Access*, 2023 (Under review).

[4] N. Kaja, A. Shaout, and D. Ma, "An intelligent intrusion detection system," *Applied Intelligence*, vol. 49, pp. 3235–3247, 2019.

[5] A. Teramoto, H. Fujita, O. Yamamuro, and T. Tamaki, "Automated detection of pulmonary nodules in pet/ct images: Ensemble false-positive reduction using a convolutional neural network technique," *Medical physics*, vol. 43, no. 6Part1, pp. 2821–2827, 2016.

[6] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, no. 2, p. 217, 2021.

[7] C. Chio and D. Freeman, *Machine learning and security: Protecting systems with data and algorithms*. " O'Reilly Media, Inc.", 2018.

[8] S. Hajj, R. El Sibai, J. Bou Abdo, J. Demerjian, A. Makhoul, and C. Guyeux, "Anomaly-based intrusion detection systems: The requirements, methods, measurements, and datasets," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 4, p. e4240, 2021.

[9] D. Oh, D. Kim, and W. W. Ro, "A malicious pattern detection engine for embedded security systems in the internet of things," *Sensors*, vol. 14, no. 12, pp. 24 188–24 211, 2014.

[10] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, no. 6, pp. 310–317, 2020.

[11] A. Belenguer, J. Navaridas, and J. A. Pascual, "A review of federated learning in intrusion detection systems for iot," *arXiv preprint arXiv:2204.12443*, 2022.

[12] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, 2022.

[13] B. Ghimire and D. B. Rawat, "Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things," *IEEE Internet of Things Journal*, 2022.

[14] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani, "A survey on iot intrusion detection: Federated learning, game theory, social psychology and explainable ai as future directions," *IEEE Internet of Things Journal*, 2022.

[15] S. Hajj, R. El Sibai, J. Bou Abdo, J. Demerjian, C. Guyeux, A. Makhoul, and D. Ginhac, "A critical review on the implementation of static data sampling techniques to detect network attacks," *IEEE Access*, vol. 9, pp. 138 903–138 938, 2021.

[16] T. G. Dietterich *et al.*, "Ensemble learning," *The handbook of brain theory and neural networks*, vol. 2, no. 1, pp. 110–125, 2002.

[17] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[18] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," *Frontiers of Computer Science*, vol. 14, pp. 241–258, 2020.

[19] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, "Ensemble learning for data stream analysis: A survey," *Information Fusion*, vol. 37, pp. 132–156, 2017.

[20] R. Polikar, "Ensemble learning," *Ensemble machine learning: Methods and applications*, pp. 1–34, 2012.

[21] Y. Sun, K. Tang, L. L. Minku, S. Wang, and X. Yao, "Online ensemble learning of data streams with gradually evolved classes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 6, pp. 1532–1545, 2016.

[22] J. N. van Rijn, G. Holmes, B. Pfahringer, and J. Vanschoren, "The online performance estimation framework: heterogeneous ensemble learning for data streams," *Machine Learning*, vol. 107, pp. 149–176, 2018.

[23] Y. Zhang and X. Jin, "An automatic construction and organization strategy for ensemble learning on data streams," *ACM SIGMOD Record*, vol. 35, no. 3, pp. 28–33, 2006.

[24] Z. Ahmadi and H. Beigy, "Semi-supervised ensemble learning of data streams in the presence of concept drift," in *Hybrid Artificial Intelligent Systems: 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part II 7*. Springer, 2012, pp. 526–537.

[25] A. Abbasi, A. R. Javed, C. Chakraborty, J. Nebhen, W. Zehra, and Z. Jalil, "Elstream: An ensemble learning approach for concept drift detection in dynamic social big data stream learning," *IEEE Access*, vol. 9, pp. 66 408–66 419, 2021.

[26] B. Krawczyk and A. Cano, "Online ensemble learning with abstaining classifiers for drifting and noisy data streams," *Applied Soft Computing*, vol. 68, pp. 677–692, 2018.

[27] H. Du, Y. Zhang, K. Gang, L. Zhang, and Y.-C. Chen, "Online ensemble learning algorithm for imbalanced data stream," *Applied Soft Computing*, vol. 107, p. 107378, 2021.

[28] H. Li, Y. Wang, H. Wang, and B. Zhou, "Multi-window based ensemble learning for classification of imbalanced streaming data," *World Wide Web*, vol. 20, pp. 1507–1525, 2017.

[29] P. Zhang, X. Zhu, Y. Shi, L. Guo, and X. Wu, "Robust ensemble learning for mining noisy data streams," *Decision Support Systems*, vol. 50, no. 2, pp. 469–479, 2011.

[30] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet, "A survey on ensemble learning for data stream classification," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–36, 2017.

[31] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," *Journal of Big Data*, vol. 1, no. 1, pp. 1–16, 2014.

[32] L. Li, R. Sun, S. Cai, K. Zhao, and Q. Zhang, "A review of improved extreme learning machine methods for data stream classification," *Multimedia Tools and Applications*, vol. 78, pp. 33 375–33 400, 2019.

[33] X. Fei, N. Shah, N. Verba, K.-M. Chao, V. Sanchez-Anguix, J. Lewandowski, A. James, and Z. Usman, "Cps data streams analytics based on machine learning for cloud and fog computing: A survey," *Future generation computer systems*, vol. 90, pp. 435–450, 2019.

[34] A. A. Benczúr, L. Kocsis, and R. Pálovics, "Online machine learning in big data streams," *arXiv preprint arXiv:1802.05872*, 2018.

[35] B. A. Tama and S. Lim, "Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation," *Computer Science Review*, vol. 39, p. 100357, 2021.

[36] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur, and S. Garg, "Securing fog-to-things environment using intrusion detection system based on ensemble learning," in *2019 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2019, pp. 1–7.

[37] A. Verma and V. Ranga, "Elnids: Ensemble learning based network intrusion detection system for rpl based internet of things," in *2019 4th International conference on Internet of Things: Smart innovation and usages (IoT-SIU)*. IEEE, 2019, pp. 1–6.

[38] Q. Abu Al-Haija and A. Al-Badawi, "Attack-aware iot network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, p. 241, 2021.

[39] M. Mohy-Eddine, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "An ensemble learning based intrusion detection model for industrial iot security," *Big Data Mining and Analytics*, vol. 6, no. 3, pp. 273–287, 2023.

[40] A. Alhowaide, I. Alsmadi, and J. Tang, "Ensemble detection model for iot ids," *Internet of Things*, vol. 16, p. 100435, 2021.

[41] O. Abu Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, "An improved pio feature selection algorithm for iot network intrusion detection system based on ensemble learning," *Expert Systems with Applications*, vol. 213, p. 118745, 2023.

[42] B. Gopalakrishnan and P. Purusothaman, "A new design of intrusion detection in iot sector using optimal feature selection and high ranking-based ensemble learning model," *Peer-to-Peer Networking and Applications*, vol. 15, no. 5, pp. 2199–2226, 2022.

[43] Q. Abu Al-Haija and M. Al-Dala'ien, "Elba-iot: an ensemble learning model for botnet attack detection in iot networks," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, p. 18, 2022.

[44] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, "lids-sioel: intrusion detection framework for iot-based smart environments security using ensemble learning," *Cluster Computing*, pp. 1–15, 2022.

[45] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of ids for iot: recent updates, security issues, and challenges," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3211–3243, 2021.

[46] K. A. Da Costa, J. P. Papa, C. O. Lisboa, R. Munoz, and V. H. C. de Albuquerque, "Internet of things: A survey on machine learning-based intrusion detection approaches," *Computer Networks*, vol. 151, pp. 147–157, 2019.

[47] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.

[48] "Baseline k-means github project." [Online]. Available: https://github.com/josephazar/baselineKmeans

[49] S. Revathi and A. Malathi, "A detailed analysis on nsl-kdd dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[50] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, pp. 1–6.