

ENHANCING IOT SECURITY: INTEGRATING MACHINE LEARNING INTO STATIC AND STREAMING IDS

Amrani Ayoub ¹
Ettazi Haitam

Received 20.08.2024.
Revised 10.02.2025.
Accepted 20.02.2025.

Keywords:

Intrusion detection systems, Internet of things, Machine Learning, IOT20ID, Streaming mode

ABSTRACT

The Internet of Things (IoT) has brought incredible advancements in healthcare, industry, smart cities, and home automation. But with these innovations come serious security challenges. Many IoT devices, which often have limited resources, are attractive targets for cyberattacks due to their extensive connectivity. This situation puts the confidentiality, integrity, and availability of data at risk. To tackle these issues, technologies like artificial intelligence (AI) and machine learning (ML) are becoming increasingly important. AI, especially through ML and deep learning (DL), provides powerful tools for spotting and preventing attacks. By analyzing the massive amounts of data generated by IoT devices, ML algorithms can learn to identify normal behavior patterns and detect any anomalies that might indicate malicious activity. Intrusion detection systems are a key part of this security strategy, spotting suspicious behaviors and supporting traditional measures. For example, using different ML algorithms on a database of IoT communication frames (IOTID20) highlights how effective these approaches can be in protecting IoT systems from emerging threats.

© 2025 Published by Faculty of Engineering



1. INTRODUCTION

The rapid growth in connected devices and the huge amount of data they exchange have created amazing opportunities to improve our lives through the Internet of Things (IoT). However, this increased connectivity also brings significant data security challenges. As more devices come online, new vulnerabilities and security gaps appear, making it easier for attackers to compromise these devices and steal sensitive information.

Today's sophisticated cyberattacks often outsmart traditional security measures, which struggle to detect

and prevent these new threats. This is where machine learning comes in. By analyzing unusual patterns and behaviors in the data exchanged between IoT devices, machine learning can detect advanced attacks. Using machine learning algorithms, we can identify emerging threats early and take initiative-taking steps to protect our data and systems.

This paper proposes combining static-mode and streaming-mode Intrusion Detection Systems (IDS) with machine learning to boost IoT security. This approach ensures quick, adaptive responses to new threats while maintaining high accuracy.

¹ Corresponding author: Amrani Ayoub
Email: amrani.ayoub@uit.ac.ma

1.1 Contribution

Our Contribution in this paper are as follows:

- Building a static intrusion detection system using Machine Learning techniques. By leveraging machine learning models, including Random Forest, LGBM, and GaussianNB, trained on historical data (the IOTID20 dataset), the system can identify known and sophisticated attack patterns while also detecting potential anomalies that may indicate new forms of attacks.
- Proposing a real-time intrusion detection system (IDS) that operate in streaming mode, moving away from static IDS. This means we are handling live data streams using algorithms like Hoeffding Tree, Adaptive Random Forest, and Online Naive Bayes from the River library.

1.2 Organization

The remainder of this paper is structured as follows: Section 2 explores several types of IDS. Section 3 introduces related works. Section 4 details our proposed methods. Section 5 examines limitations of the Streaming IDS. Finally, Section 6 presents the conclusion.

2. PRELIMINARIES

Techniques that involve passive traffic gathering and evaluation are useful for tracking and evaluating user, network, and service data, as demonstrated by (Elrawy et al., 2018). The potential threats to the confidentiality, integrity, and availability of an information system from intrusions underscore the critical role of an intrusion detection system (IDS, figure 1) in identifying and thwarting these threats (Ghorbani et al., 2010).

When talking about IDS, the following terms are frequently used:

- False positive: An alert that an IDS generates that is not in line with a genuine attack (False Alarm).
- False negative: An actual intrusion that the IDS failed to identify.
- True positives and True negatives: Match desired actions.

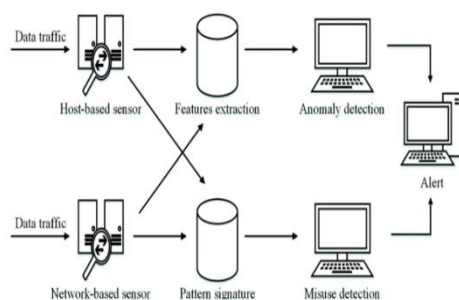


Figure 1. IDS Steps

There are three phases to an intrusion detection system's operation. The initial phase involves monitoring through sensors located either on the host or the network. The analysis stage, which comes in second, makes use of techniques for either feature extraction or pattern recognition. The last step is the detection stage, which looks for unusual activity or malicious intrusions. Within an information system, an intrusion detection system (IDS) intercepts and copies data traffic, which it then examines to look for potentially dangerous activity.

2.1 Types of IDS

The implementation of an IDS is influenced by the environment, leading to two main families of IDS (Alzahrani & Alenazi, 2021):

- NIDS: systems for detecting network intrusions that concentrate on network security.
- HIDS: Intrusion Detection Systems that operate at the host level.

HIDS:

Designed for installation on a specific system, a host-based intrusion detection system (HIDS, figure 2) is a safeguard against malicious attacks and intrusions that pose threats to the system's data or operating system.

Host environment metrics, like log files from a computer system, are usually used by HIDS.

Hence, the fundamental element for any HIDS is the extraction of features from the environment. The operational framework and positioning of a HIDS within the network are HIDS benefits include:

- Attack detection in encrypted traffic.
- Damage prevention, which leads to improved attack response.
- Accurate tracking of system operations.
- HIDS is a "last line of defense" tool that prevents attacks that NIDS is unable to detect.

The following are some of HIDS's drawbacks:

- HIDS utilizes system resources, including CPU and memory, which may impact the overall responsiveness of the host environment.
- The challenge of identifying scans.
- Since HIDS relies on the host's resources, it is susceptible to Denial of Service (DoS) attacks.

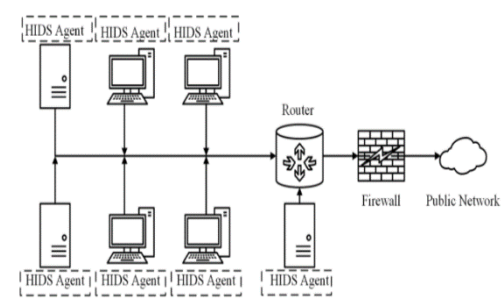


Figure 2. HIDS location

IDS: internet-centric intrusion detection system (NIDS, figure 3) examines traffic on the network packets to find intrusions and malicious activity. These systems can manifest in either hardware or software forms. For instance, Snort NIDS is an example of a software-based NIDS. Figure 3 illustrates how a NIDS operates and its positioning within the network.

One of NIDS's benefits is:

- its discreet network monitoring, which keeps it hidden from attackers.
- Scan detection made simpler by signatures.
- Screening of traffic.
- A database for attacks is not required.
- Minimal false negatives.
- Protects against intrusions because it runs covertly.

NIDS's shortcomings include.

- its inability to identify attacks in encrypted traffic.
- Problems with bandwidth if there is a lot of network traffic.
- Tolerance for false positives: sizable system changes may result in an ongoing barrage of alerts.
- Ineffectiveness if a NIDS is placed incorrectly.

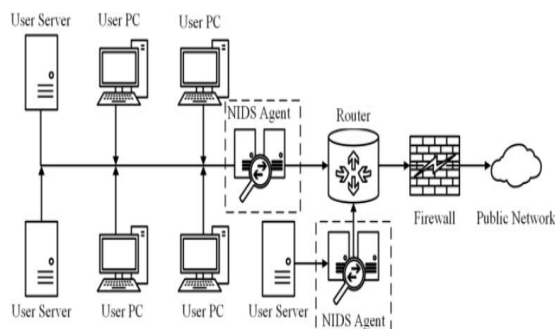


Figure 3. NIDS Location

3. RELATED WORKS

An intrusion detection system (IDS) for 6LoWPAN was presented by Kasinathan et al. (2013) and has the capacity to detect Denial of Service (DoS) crimes in Internet of Things networks. Their architecture comprises a protection manager for handling DoS incidents, an IDS system leveraging Suricata, and an IDS probe, all working collaboratively to identify and address DoS attacks. This system was developed based on an analysis of IP-based WSN vulnerabilities. The Suricata IDS, when operating on a host computer, offers the advantage of addressing energy-related issues and safeguarding the energy resources of WSNs.

Jun and Chi (2014) proposed an integrated Intrusion Detection System (IDS) based on Complex Event Processing (CEP). CEP technology is known for its capabilities in real-time data processing and advanced

pattern recognition. The IDS architecture includes key components such as an action engine module, a CEP module, an event database module, and a filtering module. The foundation of the system's intrusion detection lies in an event processing model with predefined rules. This real-time system excels in detecting intrusions within an Internet of Things (IoT) environment (Elrawy et al., 2018; Jun & Chi, 2014).

A Network Intrusion Detection System (NIDS) with statistical and rule-based techniques was proposed by Butun et al. (Butun, 2015) specifically designed with networks of wireless sensors. The architecture of the system is designed around both uplink and downlink IDS components, aligning with the hierarchical structure of WSNs. The downlink IDS is tasked with detecting unusual activities within the nodes of the network, while the uplink IDS focuses on detecting anomalies among cluster heads. This system stands out for its reliance on Wireless Sensor Network (WSN) clustering and its suitability for use in hierarchical WSN setups.

Surendar and Umamakeswari (2016) proposed an intrusion detection system (IDS) specification driven by constraints, tailored for IoT networks employing the 6LoWPAN standard. This system effectively detects sinkhole attacks and maintains Quality of Service measurements. The network reconstructs itself and isolates itself in reaction to hostile nodes, guaranteeing continuous operation. Built upon specifications and adopting a protocol model approach, this the behavior of an intrusion detection system is determined by its guidelines.

A specification-driven attack detection system (IDS) targeting 6LoWPAN-based Internet of Things connections was presented by Le et al. (2016). This IDS aims to identify various topology threats within the IPv6 Routing Protocol for Lightweight and Lossy Networks, which encompasses attacks in the Directed Acyclic Graph (DODAG) such as hole, status, local repair, peer solicitation, and destination information solicitation (DIS). Consequently, other nodes generate Directed Acyclic Graph (DODAG) Information Object (DIO) messages (Alzubaidi & Al-Samawi, 2017). This technique escalates the network's overhead. The suggested IDS examines protocol activity from log files to develop techniques for route building and upkeep to establish a stable topology. Its key features are its capacity to adapt to large-scale networks and its exceptional efficacy in detecting RPL topological hazards in a way that saves energy.

An Intrusion Detection System (IDS) relies on algorithms to implement various intrusion detection steps. There is a wide range of algorithms used in IDS for diverse types and methods of intrusion detection.

Intrusion Detection Based on Principal Component Analysis (PCA):

A common strategy in many intrusions detection system (IDS) systems is PCA. Mori et al. (2016) highlight PCA as a versatile multivariate descriptive method adept at handling quantitative data, with the capability to extend to mixed-level measurement data. In the realm of IDS, PCA serves the dual purpose of dimensionality reduction and detection. Elrawy et al. (Elrawy et al., 2016) utilized principal component analysis (PCA) to segregate major and minor components, forming the basis for a statistical IDS and anomaly-based data exploration system. The scores derived from these components play a crucial role in determining detection stages through PCA. Moreover, PCA has proven beneficial in various domains, including payload modeling, statistical modeling, data exploration, and intrusion detection methods based on machine learning (Elrawy et al., 2016).

Intrusion Detection Based on Abuse:

To recognize typical attacks, an intrusion detection method based on abuse utilizes a repository of recognized malicious code and patterns of infiltration (Bul'ajoul et al., 2015). Nevertheless, there are several drawbacks to misuse-based intrusion detection systems, including as an increased likelihood of false alarms, packet overload, and the computational expense of matching signatures (Meng et al., 2014). Additionally, in networks like wireless sensor networks (WSNs) (Abduvaliyev et al., 2013), where storing extensive databases of attack signatures becomes challenging, the efficacy of misuse-based intrusion detection systems (IDS) can be hindered by memory constraints. Additionally, signature and pattern databases for pattern-matching and intrusion detection systems using signatures that employ a misuse-based methodology must be updated on a regular basis. According to past data, these technologies are intended to detect malicious intrusions and assaults (Elrawy et al., 2018).

Anomaly-based Intrusion Detection:

By creating a normal data model with information from typical users, this kind of intrusion detection compares real-time data models to find anomalies (Bhuyan et al., 2014) (Agarwal, 2023). Hacking tools may produce noise or other phenomena, which is what causes these anomalies. Thus, anomalies are distinct actions brought about by attackers who leave evidence in the network environment (Sharma et al., 2014). The purpose of finding these traces is to find and identify unidentified attacks. An anomaly-based intrusion detection system (IDS) creates a model of typical computer environment behavior using data from typical users, which is updated regularly to detect any deviations from the (Mishra et al., 2017; Elrawy et al., 2018).

Data Mining Approach:

Using a data mining technique, one can extract knowledge from enormous amounts of data. The term "extracted knowledge" refers to intriguing patterns

discovered in the data (Duque & bin Omar, 2015). A pattern of this nature can elucidate the functioning of user or network data within a computer system. The ability of the data mining technique to automatically generate pattern-dependent models from traffic descriptions is one of its advantages. Additionally, this technique works with any type of IDS and in any kind of computer environment. The data mining technique works well when working with an infinite, continuous, and rapidly growing online data stream (Alseiari & Aung, 2015). This approach is used to design an IDS through a series of steps that involve rule learning, clustering, classification, and regression (Alseiari & Aung, 2015).

Machine Learning:

The finding or checking stage and the training or process of learning are the fundamental phases of machine learning. To understand the features of the machine's environment, computational algorithms or functions need regular data provides like reference input during the training phase. These learned characteristics are then utilized for both detection and classification in the subsequent detection step (Nishani, & Biba, 2016). A machine learning method known as "supervised learning" creates a classification model based on the training dataset's features during the learning phase. Subsequently, this model is applied to classify new instances (Namdev et al., 2015). Machine learning techniques that rely on data features instead of grouped training data are known as unsupervised learning (Namdev et al., 2015).

IoT-Based IDS Using Machine Learning:

Starting with a swift exploration of existing methodologies addressing IoT intrusions, this section will pivot towards a focused examination of machine learning driven IoT intrusion detection systems. An array of machine learning techniques has been applied to diverse forms of IoT networks, encompassing several variations, with the primary goal of detecting and responding to anomalies and intrusions (Alsheikh et al., 2014). One of the most popular intrusion detection techniques is traffic filtering, which separates malicious from legitimate packets using batch or packet analysis. Even though traffic classification works well, there are a lot of false positives that come from it, which reduces the method's reliability.

Shukla (2017) presented an IDS designed especially for 6LoWPAN-based low-power IoT networks. Their method used lightweight machine learning approaches to detect threats in Internet of Things networks that resembled worms. Three machine learning approaches—supervised learning using decision trees, unsupervised learning using K-means clustering, and hybrid learning—were incorporated into the IDS mechanism. The authors demonstrated the effectiveness of these centralized machine learning techniques. Based on how big the network is, the K-means-based intrusion detection system (IDS) achieved a detection rate ranging from 70%

to 93%, whereas the decision tree approach produced a detection rate of 71-80%. In IoT networks, the hybrid IDS mechanism eventually attained a 71–75% detection rate. Despite having a lower detection rate than the other two methods, the hybrid mechanism showed higher accuracy.

In contrast, two machine learning techniques were used by Canedo & Skjellum (2016) to identify intrusions at the IoT gateways. For Internet of Things security, the authors employed genetic algorithms and artificial neural networks (ANN). The authors examined data anomalies originating from peripheral devices during their experiments. Using both training samples and the ANN's input neurons, anomalies were predicted over 99% of the time. Three input neurons were employed in the author's additional experiments, and the prediction rate stayed above 99% with a false negative rate of 1%.

An Artificial Inflammatory Device (AID) made exclusively for IoT of IoT networks was presented by Liu et al. (2011). This creative system exhibits flexibility to the Internet of Things by learning about novel assault patterns on its own. Based on a signatures concept and machine learning, the system's development is rooted in a machine learning strategy inspired by the mechanisms of the artificial immune system. This system's primary goal is to improve IoT network security by acting as a network intrusion detection system. A standout feature includes its capability to dynamically adapt to new environments and autonomously learn emerging attack patterns.

Krimmling and Peter (2014) introduced a Network Intrusion Detection System (NIDS) based on machine learning, with a particular focus on anomalies and signatures. This system is unique in that it is made to work with CoAP applications and is optimized for use with lightweight algorithms in intelligent public transportation applications.

A system to detect network intrusions (NIDS) for Wireless Sensor Networks (WSNs) was suggested by Garcia-Font et al. (2017). This system combines a machine learning method with a model based on signatures. Utilizing two distinct detection engines—one for anomalies and the other for signatures—the system aims to enhance the detection rate while managing false positives. It is tailored to assist smart city administrators in identifying intrusions through an intrusion detection system and an attack classification scheme, concentrating on Wireless Sensor Networks in various smart city settings. Among its noteworthy aspects is its suitability for large-scale WSNs.

Deng et al. (2018) introduced a Network Intrusion Detection System (NIDS) employing the Principal Component Analysis (PCA) and fuzzy c-means (FCM) clustering algorithms. By combining data mining and machine learning approaches, this technology improves

the accuracy of intrusion detection in Internet of Things (IoT) networks. The FCM algorithm serves as a clustering method, while the PCA method is employed for feature reduction and selection. The system's effectiveness in boosting detection efficiency while minimizing false positives is demonstrated through testing on the KDD-CUP99 dataset.

Amouri et al. (2018) proposed a NIDS that integrates the machine learning technique using the process model. There are two detection stages in this system. During the initial phase, referred to as local detection, skilled sniffers gather information about network activity to employ decision trees and supervised learning to generate a collection of accurately classified instances (CCI). Accumulated Fluctuation Measures (AMoF) are generated by super nodes using the CCIs collected during the second phase, which is called global detection. These are distinct temporal profiles for malicious and normal nodes. This system's low resource requirements and low computational complexity are its key characteristics.

4. PROPOSED METHODS

The development and implementation of a sophisticated static Machine Learning (ML)-based intrusion detection system is crucial. This system utilizes a combination of advanced ML algorithms trained using the IOTID20 dataset. This dataset represents a wide range of scenarios, enabling the model to learn to identify a broad spectrum of anomalies.

Dataset:

IoT devices and interconnection structures are combined in the IoTID20 dataset (figure 4). To create the IoTID20 dataset (table 1, figure 5), a standard smart home setup was made up of the EZVIZ Wi-Fi camera and the SKT NGU smart home device. The IoTID20 dataset's testbed environment is depicted in the figure. The smart home's Wi-Fi router is linked to these two Internet of Things devices.

Smartphones, tablets, and laptops are among the additional gadgets linked to the smart home router. The SKT NGU and the EZVIZ Wi-Fi camera are target devices, whereas all other connected devices are attacking equipment.

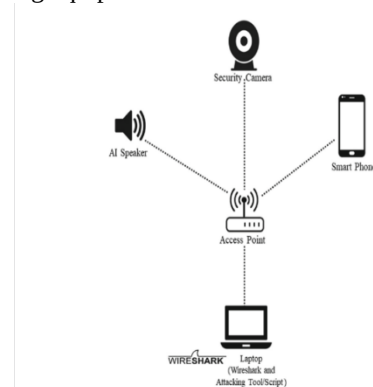


Figure 4. IOTID20

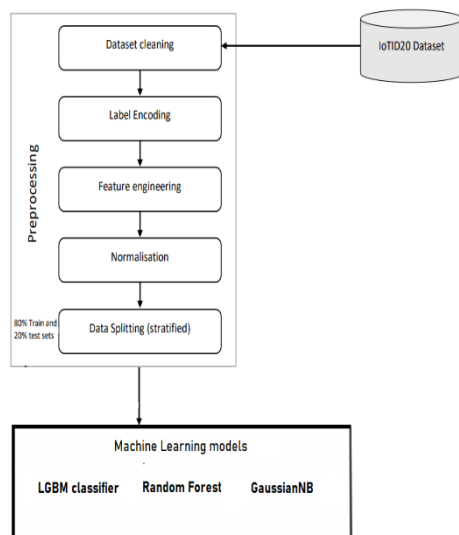
Table 1. Binary, Categorical, and Sub categorical Data in the IoTID20 Dataset (Ullah et al., 2022)

Binary	Category	Subcategory
Normal, Anomaly	Normal, DoS, Mirai, MITM, Scan	Normal, Syn Flooding, BruteForce, HTTPFlooding, UDPFlooding, ARP Spoofing, Host Port., OS

Binary label distribution		Subcategory distribution	
Normal	40073	Type	Instances
Anomaly	585710	Normal	40073
		DoS	59391
		Mirai Ack Flooding	55124
		Mirai Brute force	121181
		Mirai HTTP Flooding	55818
		Mirai UDP Flooding	183554
		MITM	35377
		Scan Host Port	22192
		Scan Port OS	53073

Figure 5. Normal and Attacked Instances in the IoTID20 Dataset (Ullah et al., 2022)

The data preprocessing step for machine learning involves preparing and cleaning raw data before using it to train a machine learning model. This step is essential to improve the quality and reliability of the results obtained from the model. Here is an illustrative diagram of all the crucial steps in the preprocessing phase (figure 6):

**Figure 6.** Preprocessing phase**Data Cleaning:**

Before training a model, a dataset must be examined for empty and undefined instances. The dataset in this experiment was validated using the built-in Python library (Pandas). There are missing values in the IoTID20 dataset. We eliminated each occurrence of a missing value to clean up the dataset.

Label Encoding:

In the context of our dataset, label encoding becomes essential for transforming categorical data into a numerical format. This technique assigns a unique numerical label to each category, facilitating the compatibility of categorical features with machine learning algorithms. Through label encoding, we enable the algorithms to effectively interpret and derive insights from these categorical variables during the modeling process.

Feature Engineering:

Every dataset has a distinct collection of features. A dataset must be stripped of any irrelevant features that don't affect the output label if it has several features. These features can cause overfitting and underfitting, which can seriously impair the runtime and classifier performance. The filtering approach was applied in this investigation.

Normalization:

Normalization is a critical preprocessing step that focuses on scaling numerical features to a standard range, typically between 0 and 1. By doing so, we mitigate issues arising from variations in the scale of distinctive features. This ensures that each feature contributes proportionally to the machine learning model, preventing any feature from exerting undue influence due to its scale.

Data Splitting:

One of the frequently used preprocessing techniques for assessing the effectiveness of ML/DL models is to divide the data into training and testing sets. Random data splitting can result in an uneven distribution of data in an unbalanced dataset, making it difficult to assess model performance with precision. The stratified method in our case divides the data at an 80% training and 20% testing ratio for each class into training and testing sets.

Results:

Here is a table (table 2) showing the results of the models used in this work: LightGBM Classifier, Random Forest, GaussianNB, along with the one that achieved the highest overall accuracy compared to the others:

Table 2. Performance Results of Each Model in Static Mode

	Accuracy	Precision	Recall	Time
LightGBM	99,897%	99,951%	99,939%	6,53908
Random Forest	99,928%	99,946%	99,976%	30,55096
GaussianNB	39,091%	99,697%	34,995%	4,03808

After evaluating the performance of each model, it's evident that the Random Forest model stands out with its highest accuracy. This exceptional accuracy is attributed to its proficiency in handling complex data and detecting intrusion patterns with remarkable precision.

Furthermore, Random Forest's ability to resist overfitting makes it a more reliable choice for real-world scenarios.

In contrast, the Naive Bayes model (GaussianNB) exhibited lower accuracy, primarily due to our intrusion data's adherence to a Gaussian (normal) distribution. While GaussianNB may perform well with numeric and continuous data, it may not reach the performance levels achieved by more complex models like Random Forest and LGBM, especially when dealing with intricate and diverse data.

The importance of this performance becomes evident when considering sophisticated attacks on industrial infrastructure, such as Stuxnet (2010), Triton (2017), and other incidents like the 2015 Ukrainian power grid attack. These incidents emphasize the critical role of cybersecurity in the industrial sector, where industrial control systems (SCADA) and critical infrastructure face increasing vulnerability to cyber threats. The consequences can range from major outages to economic disruptions and threats to national security.

Many of these attacks, particularly those targeting industrial infrastructure, involve extended periods of observation and preparation by attackers to go unnoticed. In this context, our static ML-based IDS system proves ideally suited. It excels in learning from historical data, identifying subtle signals indicative of malicious activities, and promptly generating real-time alerts upon detecting any suspicious behavior. Achieving over 99% accuracy, this model's exceptional performance can be attributed to its capability to detect weak signals and malicious behaviors, even when attacks unfold over an extended period. By enhancing the security of industrial infrastructures, this project contributes to bolstering their resilience against emerging threats, thereby preserving the integrity of critical industrial operations.

However, while static IDS systems are effective, they come with inherent limitations, especially in comparison to their real-time streaming counterparts. One major limitation is their ability to respond to emerging threats, as previously discussed. Static IDS relies on historical data and predefined models to detect known anomalies and attack patterns, which can hinder their quick adaptation to evolving attack strategies and zero-day vulnerabilities.

Moreover, in static mode, IDS may struggle to cope with high-speed data streams generated in modern IoT environments. The sheer volume and velocity of data can overwhelm traditional IDS systems, leading to delays and missed detections.

Additionally, static IDS systems are less effective in scenarios where immediate threat responses are crucial. Some cyberattacks demand real-time reactions to prevent substantial damage, a feature that static IDS systems may lack.

These limitations encountered in static mode highlight the importance of exploring the utility of streaming IDS. Transitioning to a real-time IDS approach offers a more adaptive and responsive threat detection mechanism, serving as a valuable complement to static mode. This shift is an initiative-taking measure to prevent threats, particularly in dynamic, high-speed IoT environments where emerging threats necessitate immediate attention and action.

In the second phase of our project, we turn our focus towards building an online Intrusion Detection System (IDS), a process that relies on distinct methodologies and tools compared to our static IDS. To accomplish this, we undertook several crucial and significant steps.

A key step involved dividing our IOTID20 dataset into data streams. This operation simulated the dynamic nature of data generated in modern IoT environments. To better understand this process, envision our initial dataset being split into continuous data segments, with each considered as a stream. This segmentation mirrors the constant flow of data in a real IoT environment, where current information is continually generated.

Regarding the models, we leveraged several algorithms specifically designed for data streams, which we sourced from the river library. This library is a valuable resource in the fields of online learning and real-time intrusion detection. Among the models we used are the Hoeffding Tree, the Adaptive Random Forest, and the Online Naive Bayes (figure 7).

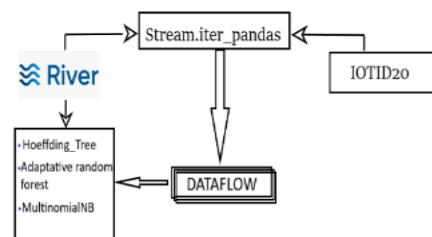


Figure 6. Scheme models

Hoeffding Tree: This is an adaptive decision tree model designed to manage continuous data streams. The Hoeffding Tree model can make decisions on ever-evolving data using statistics based on the Hoeffding test. It adapts to new data as it arrives, making it an ideal choice for environments where information is constantly being updated.

Adaptive Random Forest: This model is a variant of the classic Random Forest adapted for data streams. It relies on the idea of creating and adjusting decision tree forests incrementally as new data arrives. The Adaptive Random Forest allows for dynamic adaptation to changes in data, making it suitable for ever-changing IoT environments.

MultinomialNB: or Multinomial Naive Bayes, is an adapted version of the classic Naive Bayes algorithm for data streams. It is particularly useful for online

classification and can adapt to new data while maintaining good accuracy.

Results: By employing these models tailored for data streams, our online IDS can detect threats as they emerge, providing rapid and adaptive responses. The results are presented in the form of graphs, with each model having a dedicated graph depicting changes in accuracy based on the number of records (figure 8, figure 9 and figure 10):

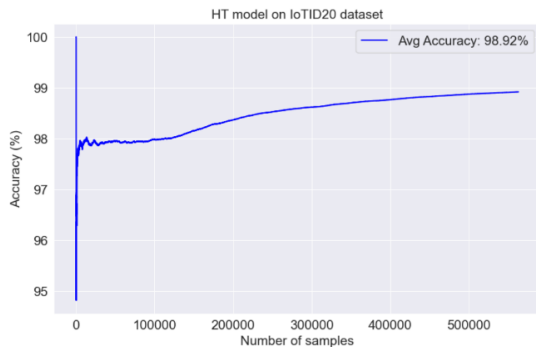


Figure 8. Hoeffding tree model

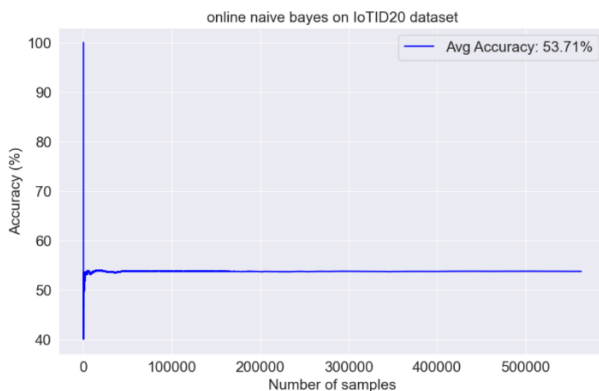


Figure 9. Multinomial Naïve Bayes model

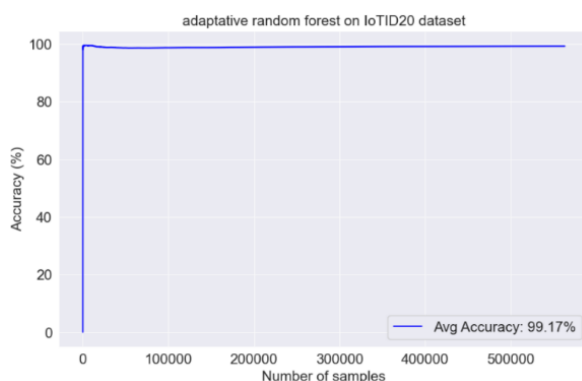


Figure 7. Adaptive Random Forest model

The Adaptive Random Forest model stands out with significantly higher accuracy compared to the other models. It manages to achieve an exceptional level of accuracy of 99%. This remarkable performance is of paramount importance for our streaming intrusion detection system (table 3).

An accuracy of 99% means that our online IDS is extremely dependable in identifying real-time threats. As data continues to flow, the Adaptive Random Forest model maintains an elevated level of accuracy. This reliability is essential to ensure a swift and adaptive response to emerging threats in dynamic and ever-evolving IoT environments.

Table 3. Performance Results of Each Model in Static Mode

	Accuracy	Precision	Recall
Hoeffding Tree	98.916%	99.385%	99.457%
Adaptive Random Forest	99.17%	99.273%	99.844%
Online Naive Bayes	53.711%	99.304%	50.9%

5. LIMITATIONS

Streaming IDS, while highly promising, also have some limitations that are crucial to consider. Firstly, these systems require rigorous management of real-time data streams, which can be complex to set up. High data rates and the variety of sources within IoT environments can lead to issues such as latency, data loss, or false alerts if the systems are not properly configured.

Furthermore, streaming IDS systems are typically tailored to detect immediate threats, but they may struggle to identify more subtle or slowly evolving attacks that require in-depth analysis over a longer period. Detection of these threats can become a challenge, and effective integration of machine learning models and deep learning algorithms is necessary to address these challenges.

Another challenge is resource management. Streaming IDS systems consume significant computing resources, especially when processing massive volumes of real-time data. The use of powerful hardware and servers is often required to ensure optimal performance, which can be costly.

6. CONCLUSION

In conclusion, this article has delved into the fascinating realm of Intrusion Detection Systems (IDS) in the context of IoT environments. As the Internet of Things continues to expand, the need for robust and lightweight security becomes increasingly urgent. IoT networks form the backbone of smart environments, and any security gap can have direct consequences on the services and applications relying on them. Threats such as Denial of Service attacks, Distributed Denial of Service attacks, probing attacks, and those targeting the RPL routing protocol have implications for IoT network security, making IoT security a paramount concern.

To address this challenge, Intrusion Detection Systems have emerged as a potential solution. We thoroughly examined IDS systems specifically designed for IoT environments, with a particular focus on their

implementation in the industrial domain. We utilized the IOD20 database to assess the effectiveness of our static-mode IDS, achieving promising results with machine learning models such as Random Forest and Light Gradient Boosting Machine.

However, we also acknowledged the limitations of static-mode IDS, particularly concerning their responsiveness to emerging threats and their ability to manage high-speed data streams. This led us to explore streaming-mode IDS, which can detect threats in real-time and provide rapid, adaptive responses. We shared the steps involved in constructing our online IDS, highlighting the use of data-stream-adapted models, including the Hoeffding Tree, Adaptive Random Forest, and Online Naive Bayes from the River library.

The results obtained demonstrated that the Adaptive Random Forest model achieved exceptional accuracy of 99%, reinforcing our confidence in intrusion detection in streaming mode. As data continues to flow, this model maintains an elevated level of accuracy, ensuring a swift response to threats.

Our exploration of streaming-mode IDS has opened exciting prospects for intrusion detection in IoT environments. However, we also emphasized the challenges associated with these systems, such as the management of real-time data streams and high resource requirements. Nevertheless, by combining the strengths of static-mode and streaming-mode IDS, it is possible to enhance IoT environment security and ensure their resilience against emerging threats.

References:

- Abduvaliyev, A., Pathan, A. S. K., Zhou, J., Roman, R., & Wong, W. C. (2013). (2013). On vital domains of intrusion detection systems in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 15(3), 1223–1237.
- Agarwal, N., Pandey, R., & Rajagopal, S. (2023). Research on IIoT security: Novel machine learning-based intrusion detection using TCP/IP packets. *Proceedings on Engineering Sciences*, 5(1), 63–68. <https://doi.org/10.24874/PES.SI.01.008>
- Alseieri, F. A. A., & Aung, Z. (2015). Real-time anomaly-based distributed intrusion detection systems for advanced metering infrastructure utilizing stream data mining. In *Proceedings of the 2015 International Conference on Smart Grid and Clean Energy Technologies* (pp. 148–153). IEEE.
- Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P.. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, 16(4), 1996–2018.
- Alzahrani, A. O., & Alenazi, M. J. (2021). Designing a network intrusion detection system based on machine learning for software defined networks. *Future Internet*, 13(5), 1–18. <https://doi.org/10.3390/fi13050111>
- Alzubaidi, M., & Al-Samawi, A. M. (2017). A review of sinkhole detection mechanisms in RPL. In *Proceedings of the 8th International Conference on Information Technology* (pp. 369–374). IEEE.
- Amouri, A., & Alapati, V. (2018). Multilayer anomaly-based network intrusion detection for IoT. In *Proceedings of the IEEE 19th Wireless and Microwave Technology Conference* (pp. 1–4). IEEE.
- Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: Methods, systems, and tools. *IEEE Communications Surveys & Tutorials*, 16(1), 303–336.
- Bul'ajoul, W., James, A., & Pannu, M. (2015). Improving network intrusion detection system performance through quality of service configuration and parallel technology. *Journal of Computer and System Sciences*, 81(6), 981-999.
- Butun, I., & Rahebi, I. (2015). Multi-level clustering-based intrusion detection system for hierarchical wireless sensor networks. *Sensors*, 15(11), 28960–28978.
- Deng, L., Li, D., Yao, X., & Wang, H. (2018). Mobile network intrusion detection for IoT system based on transfer learning algorithm. *Cluster Computing*, 21(4), 1–6.
- Duque, S., & bin Omar, M. N. (2015). Using data mining algorithms for developing a model for intrusion detection system (IDS). *Procedia Computer Science*, 61, 46–51.
- Elrawy, M. F., Awad, A. I., & Hamed, H. F. (2018). Intrusion detection systems for IoT-based smart environments: A survey. *Journal of Cloud Computing*, 7(21), 1–21. <https://doi.org/10.1186/s13677-018-0123-6>
- Elrawy, M. F., Awad, A. I., & Hamed, H. F. (2016). Flow-based features for robust intrusion detection system targeting mobile traffic. In *Proceedings of the 23rd International Conference on Telecommunications* (pp. 1–6). IEEE.
- Garcia-Font, V., Garrigues, C., & Rifà-Pous, H. (2017). Attack classification schema for smart city WSNs. *Sensors*, 17(4), 771.
- Ghorbani, A. A., Lu, W., & Tavallaee, M. (2010). *Network intrusion detection and prevention: Concepts and techniques*. Springer.
- Jun, C., & Chi, C. (2014). Designing complex event processing-based IDS in Internet of Things. In *Proceedings of the Sixth International Conference on Measuring Technology and Mechatronics Automation* (pp. 226–229). IEEE.

- Kasinathan, P., Pastrone, C., Spirito, M. A., & Vinkovits, M. (2013). Denial of service detection in 6LoWPAN-based Internet of Things. In *Proceedings of the IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications* (pp. 600–607). IEEE.
- Krimmling, J., & Peter, S. (2014). Integration and evaluation of intrusion detection for CoAP in smart city applications. In *Proceedings of the IEEE Conference on Communications and Network Security* (pp. 73–78). IEEE.
- Le, A., Loo, J., Chai, K. K., & Aiash, M. (2016). A specification-based IDS for detecting attacks on RPL-based network topology. *Information*, 7(2), 1–19.
- Liu, C., Yang, J., Chen, R., Zhang, Y., & Zeng, J. (2011). Research on immunity-based intrusion detection technology for the Internet of Things. In *Proceedings of the Seventh International Conference on Natural Computation* (pp. 212–216). IEEE.
- Meng, W., Li, W., & Kwok, L. F. (2014). Enhancing the performance of signature-based network intrusion detection systems using an improved filtering mechanism. *Computer Security*, 43, 189–204.
- Mishra, P., Pilli, E. S., Varadharajan, V., & Tupakula, U. (2017). Intrusion detection techniques in cloud environment. *Journal of Network and Computer Applications*, 77, 18–47.
- Mori, Y., Kuroda, M., & Makino, N. (2016). *Non-linear principal component analysis and its applications*. Springer.
- Namdev, N., Agrawal, S., & Silkari, S. (2015). Recent advancement in machine learning-based internet traffic classification. *Procedia Computer Science*, 60, 784–791.
- Nishani, L., & Biba, M. (2016). Machine learning for intrusion detection in MANET: A state-of-the-art survey. *Journal of Intelligent Information Systems*, 46(2), 391–407.
- Sharma, R. K., Pandey, A. K., Jayabalan, B., & Naval, P. (2014). Utilizing machine learning-based intrusion detection technologies for network security. *Proceedings on Engineering Sciences*, 6(2), 311–320. <https://doi.org/10.24874/PES.SI.24.02.014>
- Shukla, P. (2017). ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things. In *Proceedings of the Intelligent Systems Conference* (pp. 234–240). IEEE.
- Canedo, J., & Skjellum, A. (2016). Using machine learning to secure IoT systems. In *Proceedings of the 14th Annual Conference on Privacy, Security and Trust* (pp. 219–222). IEEE.
- Surendar, M., & Umamakeswari, A. (2016). An intrusion detection and response system for Internet of Things with 6LoWPAN. In *Proceedings of the International Conference on Wireless Communications, Signal Processing and Networking* (pp. 1903–1908). IEEE.
- Ullah, S., Ahmad, J., Khan, M. A., Alkhamash, E. H., Hadjouni, M., Ghadi, Y. Y., ... & Pitropakis, N. (2022). A new intrusion detection system for the Internet of Things via deep convolutional neural network and feature engineering. *Sensors*, 22(14), 1–16.

Amrani Ayoub

Ibn Tofail University, Faculty of
Science, LaRIT
Kenitra,
Morocco
amrani.ayoub@uit.ac.ma
ORCID 0000-0001-6301-0976

Ettazi Haitam

Hassan Premier University, FSTS
Settat,
Morocco
ettazi.haitam@gmail.com
ORCID 0000-0002-7303-3044
