Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060935

ARTICLE

An Intrusion Detection System Based on HiTar-2024 Dataset Generation from LOG Files for Smart Industrial Internet-of-Things Environment

Tarak Dhaouadi[1], Hichem Mrabet[1,2,*], Adeeb Alhomoud[3] and Abderrazak Jemai[1,4]

[1]SER'Com Laboratory, Tunisia Polytechnic School, University of Carthage, Tunis, 2078, Tunisia
[2]Computer Sciences Department, Faculty of Sciences of Tunis, University of Tunis El Manar, Tunis, 1001, Tunisia
[3]Department of Science, College of Science and Theoretical Studies, Saudi Electronic University, Riyadh, 11673, Saudi Arabia
[4]Computer Sciences Department, INSAT, University of Carthage, Tunis, 1080, Tunisia

*Corresponding Author: Hichem Mrabet. Email: hichem.mrabet@fst.utm.tn

Received: 13 November 2024; Accepted: 13 January 2025; Published: 06 March 2025

ABSTRACT: The increasing adoption of Industrial Internet of Things (IIoT) systems in smart manufacturing is leading to raise cyberattack numbers and pressing the requirement for intrusion detection systems (IDS) to be effective. However, existing datasets for IDS training often lack relevance to modern IIoT environments, limiting their applicability for research and development. To address the latter gap, this paper introduces the HiTar-2024 dataset specifically designed for IIoT systems. As a consequence, that can be used by an IDS to detect imminent threats. Likewise, HiTar-2024 was generated using the AREZZO simulator, which replicates realistic smart manufacturing scenarios. The generated dataset includes five distinct classes: Normal, Probing, Remote to Local (R2L), User to Root (U2R), and Denial of Service (DoS). Furthermore, comprehensive experiments with popular Machine Learning (ML) models using various classifiers, including BayesNet, Logistic, IBK, Multiclass, PART, and J48 demonstrate high accuracy, precision, recall, and F1-scores, exceeding 0.99 across all ML metrics. The latter result is reached thanks to the rigorous applied process to achieve this quite good result, including data pre-processing, features extraction, fixing the class imbalance problem, and using a test option for model robustness. This comprehensive approach emphasizes meticulous dataset construction through a complete dataset generation process, a careful labelling algorithm, and a sophisticated evaluation method, providing valuable insights to reinforce IIoT system security. Finally, the HiTar-2024 dataset is compared with other similar datasets in the literature, considering several factors such as data format, feature extraction tools, number of features, attack categories, number of instances, and ML metrics.

KEYWORDS: Intrusion detection system; industrial IoT; machine learning; security; cyber-attacks; dataset

1 Introduction

The emergence of the Industrial Internet-of-Things (IIoT) has accompanied in a new generation of connectivity, seamlessly linking machines, sensors, and control systems [1]. Despite the unprecedented opportunities for application real-time boosting and decision-making based on data, this integration brings forth profound cybersecurity challenges. Security stands as a paramount concern in the IIoT landscape, with interconnected systems facing escalating risks of cyber-attacks that span from operational disruptions to compromising sensitive data and worker safety [2]. Furthermore, cyber-attacks on manufactories can be very harmful, like German Steel Mill Thyssenkrupp in 2014 and Norsk Hydro Aluminum incidents in 2019 [3]. Recently, in 2023, a proposed architecture can prevent session hijacking and address resolution protocol

(ARP) spoofing attacks. The latter attack is prevented by mapping all MAC addresses automatically to their respective IP addresses in a flexible manufacturing simulator network [4].

In the state-of-the-art, various solutions are proposed to fix cyber-attack issues in the smart manufacturing environment, such as blockchain (BC) and machine learning (ML) approaches. On the one hand, BC components based on smart contracts is proposed to guarantee data integrity related to the sensors access control system in the IIoT context [5–8]. On the other hand, ML is very useful in cyber-attack detection based on supervised classifiers [9–12].

To address these challenges, we introduce the HiTar-2024 dataset, a meticulously crafted LOG-based dataset designed to fortify IIoT security. HiTar-2024 captures detailed IIoT device activities, events, and anomalies, facilitating rapid attack detection, performance monitoring, and understanding of abnormal behaviors. This research propels us to the forefront of IIoT cybersecurity, with a primary focus on the state-of-the-art construction of the HiTar-2024 dataset. Our exploration involves meticulous scrutiny of methodologies for generating robust datasets, underscoring best practices [13]. Additionally, we delve into various attack scenarios encountered by Industrial IoT systems, shedding light on attack mechanisms and potential impacts on operations [14]. To evaluate HiTar-2024's effectiveness in classifying cyberattacks, we employ supervised ML techniques by using the WEKA tool [15]. Our research unfolds in two pivotal phases: the first involves the meticulous generation of LOG files from the AREZZO industrial simulator, creating a realistic environment [16]; the second entails leveraging HiTar-2024 for attack classification. The approach involves sophisticated supervised ML algorithms within the WEKA tool, assessing detection capabilities within a simulated industrial context [17]. The experimental phase unfolds as a meticulous analysis of algorithmic performance, representing a crucial juncture in our pursuit of bolstering security for IIoT systems. This phase is designed to provide nuanced insights into the capabilities of our approach, focusing on the precise identification and classification of simulated cyber-attacks within the generated HiTar-2024 dataset. Employing sophisticated supervised ML algorithms within the WEKA tool, we subject the HiTar-2024 dataset to rigorous evaluation in a simulated industrial context. The chosen algorithms undergo training with the dataset to develop robust intrusion detection models. Subsequently, the models are rigorously tested against various attack scenarios, assessing their ability to accurately identify and classify anomalies.

It is crucial to clarify that while the environment is simulated, the interactions within it are real. This distinction is essential for several reasons. Firstly, it ensures that the HiTar-2024 dataset accurately reflects real-world scenarios, thereby enhancing its applicability and reliability in practical settings. By simulating the environment, we can control and replicate various conditions to test the robustness of the intrusion detection system (IDS). However, the interactions being real means that the data generated from these interactions is authentic and representative of actual network behavior and potential threats. This approach combines the benefits of a controlled experimental setup with the authenticity of real-world data, thereby ensuring the scientific rigor and validity of the HiTar-2024 and its applications.

The assessment encompasses diverse attack mechanisms and potential impacts on industrial operations, mirroring real-world challenges faced by IIoT systems. Through this process, we aim to not only quantify the accuracy of detection but also understand the models' resilience in distinguishing between routine activities and security threats. The results obtained from this in-depth exploration serve as a beacon guiding the viability and effectiveness of our approach in realistic scenarios. The findings shed light on the strengths and limitations of the HiTar-2024 dataset and its integration with supervised ML techniques. As a testament to its innovative contributions, HiTar-2024 emerges as a pivotal tool in the arsenal of IIoT cybersecurity, ensuring a robust defense against evolving cyber threats in industrial settings.

Our approach is based on two major axes: the first step involves the careful generation of LOG files from the AREZZO industrial simulator while simulating an industrial fabrication process. This simulator offers a

realistic simulation environment, faithfully reproducing authentic industrial scenarios. The goal is to capture a variety of activities and events reflecting the complexity of a real-world industrial context. The LOG files thus obtained constitute the cornerstone of our dataset, and we take particular care to guarantee the diversity of the data generated to ensure the representativeness of the whole. In the second phase of our study, we focus on using this newly created dataset for attack classification. By exploiting sophisticated supervised ML algorithms integrated through the WEKA tool, we aim to evaluate the attack detection capacity in a simulated industrial context. The obtained results can be used by an IDS based on ML algorithms to detect threats in the context of the IIoT environment. Then, an evaluation of the performance of the IDS is performed through several metrics such as accuracy, precision, sensitivity and F-measure.

Existing datasets used for cyber-attack detection like CAIDA [12], UNSW-NB15 [13], KDD99 [17], NSLKDD [18], CICIDS17 [19], IoTHIDS [20] and IoT-SH [21] offer valuable insights into network security, but they primarily focus on general IoT or legacy systems and fail to capture the complexities of modern **IIoT** environments. While datasets such as **TON-IoT** and **Edge-IIoTset** are more relevant to IIoT, they still lack specific industrial attack scenarios. **HiTar-2024**, generated using the **AREZZO simulator**, fills this gap by focusing on **smart manufacturing** environments. With five attack classes, including Normal, **Probing**, Remote to Local **(R2L)**, **User to Root (U2R)**, and **Denial of Service (DoS)** as intrusion types, it provides a more realistic and diverse dataset for training **IDS** in IIoT systems, addressing the unique challenges faced in industrial cybersecurity.

The main contributions of the paper are given as follows:

- Traffic classification is based on five classes: Normal, Probing, R2L, U2R and DoS.
- Traffic labelling description by checking conditions for each traffic class.
- HiTar-2024 dataset generation process from an industrial environment simulator.
- Performance analysis of the HiTar-2024 dataset with various ML classifiers.
- Comparison between the HiTar-2024 dataset and similar one in the literature.

The paper is organized into the following sections: The second section presents the state of the art of the existing dataset in the IoT environment. Section 3 exhibits the AREZZO industrial simulator. Section 4 introduces the IDS algorithm description, followed by the HiTar-2024 dataset process generation in Section 5. Then, a performance evaluation of the generated HiTar-2024 dataset was performed using a supervised ML technique, as shown in Section 6. Finally, a conclusion is drawn at the end of this contribution.

2 State of the Art for Existent IoT Dataset

Table 1 exhibits the existent Datasets for IoT applications from 1999 until 2023 as function of the year of generation, name, features number, attack classes and description.

The first dataset is KDD'99, which was created by DARPA 98 IDS evaluation program in 1999 [17]. The latter was generated from tcpdump data based on seven weeks of network traffic and contains 41 features and four attack classes. The four attack classes include Probing, R2L, U2R, and DoS attacks. Then, the NSLKDD is known as an enhanced version of KDD'99 dataset by removing duplication and reducing the size [18]. It contains 42 features and the same four attack classes, such as Probing, R2L, U2R and DoS attack.

Fig. 1 represents the attack classification through four classes: Probing, R2L, U2R and DoS attacks.

Probing: Attackers use a variety of tools to discover open entry points, identify active services, determine system types, and pinpoint software specifics. They may take advantage of common oversights such as unchanged default security settings, known weaknesses in software, and setup mistakes to infiltrate protected networks. Additionally, they might use deceptive tactics to extract information from individuals or system overseers.

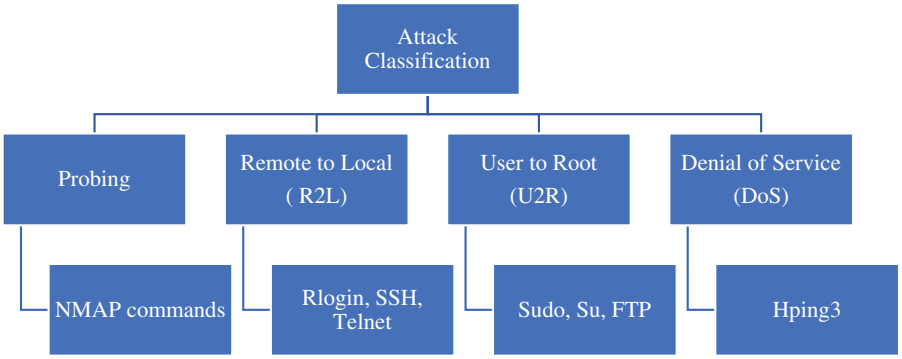

Figure 1: Attack classification used during attack scenario in AREZZO environment

R2L: Attackers explore networks using MiM attacks to intercept and manipulate network traffic. They often exploit weaknesses in network protocols, insecure connections, and misconfigurations to gain access to internal networks. They can also use malware to establish persistent connections, thereby compromising network security. Common techniques include identity theft, interception of sensitive data, and compromise of data integrity.

U2R: Attackers explore target systems to identify software vulnerabilities, configuration errors, and design weaknesses that could allow them to take control of the system as a privileged user. They often use techniques such as privilege escalation, executing malicious code, manipulating file permissions, and creating backdoors to ensure continued access. Common techniques include using malware, exploiting known vulnerabilities, and using brute force attacks to compromise credentials.

DoS: Attackers use botnets and distributed DoS (DDoS) attacks to overwhelm the resources of target systems, causing downtime and disruption. They often exploit vulnerabilities in network protocols, web services and applications to generate massive traffic. They can also use flood attacks to overload servers with malicious requests. Common techniques include sending malformed data packets, misusing legitimate protocol requests, and consuming excessive system resources.

CICIDS dataset is characterized by wide attack diversity and complete network configuration [19]. In addition, CICIDS contains BENIGN, Bot, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, PortScan, SQL Injection, Brute Force attack classes. The IoTHIDS dataset was introduced in 2018 and proposed a dataset based on domestic IoT network traffic infected by IoT botnet malwares [20]. The covered botnet related to DDoS attacks on IoT devices are Mirai, Hajime, Aidra, Bashlite, Dofloo, Tsunami, and Wroba. The authors propose a three-layer based-IDS using a supervised approach to detect cyber-attacks on IoT smart home networks [21]. Nonetheless, the range of popular network-based cyber-attacks detected covered DoS, MiM/spoofing, reconnaissance, and replay attacks. Also, the generated dataset extracts 121 features from TCPDUMP captured packet. In 2020, Alsaedi et al. [22] proposed a new dataset for both IoT and IIoT IDS composed of 127 features and 8 attack classes. The latter attack classes include Normal, DDoS, DoS, injection, Man in the middle (MiM), Password, XSS, and Scanning attacks. The IoT-23 dataset consists of 23 attack scenarios for various IoT network traffic created by Avast AIC laboratory [23]. The IoT-23 dataset contains 8 classes attack such as C&C, DDoS, FileDownload, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan and Torii.

The HIKARI-2021 dataset is generated from tcpdump traffic capture [24]. It contains benign and encrypted network traffic. Also, the dataset contains 86 features, two benign traffic categories and four attack categories. The four attack categories are Bruteforce, Bruteforce-XML, Probing and XMRIGCC CryptoMiner.

In 2022, Ferrag et al. generated an Edge-IIoTset dataset that could be used by an IDS based on centralized and federated learning for IoT and IIoT applications [25]. The generated dataset contains five classes' attacks, such as DoS/DDoS attacks, Information gathering, MiM, Injection and Malware attacks. Finally, a comparison of centralized model performance and federated deep learning approach is performed in the paper in terms of Fl-score, Recall, and Precision under multi-class classification approach.

Likewise, the authors published a gathered dataset for Linux-based IoT devices [26] in 2022. The IoT Linux dataset is generated based on the simulation of random attacks. Finally, for CICIoT dataset generation [27], two steps were used, including benign data generation and attack framework execution. In addition, the attack framework involves DDoS, DoS, Reconnaissance, Web-based, brute force, ARP spoofing, and Mirai attacks to an IoT topology.

Table 1: Related works to existent dataset for IoT

Years	Dataset name	Features number	Attack classes	Description
1999	KDD [17]	41	Probing, R2L, U2R, DoS.	The dataset is generated from tcpdump data based on 7 weeks of network traffic.
2009	NSLKDD [18]	42	Probing, R2L, U2R, DoS.	An enhanced version of KDD'99 dataset.
2017	CICIDS [19]	83	BENIGN, Bot, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, PortScan, Sql Injection, Brute Force.	CICIDS dataset is characterized with wide attack diversity and Complete Network configuration.
2018	IoTHIDS [20]	6	Mirai, Hajime, Aidra, Bashlite, Dofloo, Tsunami, and Wroba.	IoTHIDS proposes a dataset based on domestic IoT network traffic infected by IoT botnet malwares.
2019	IoT-SH [21]	121	DoS, MiM/spoofing, reconnaissance, and replay.	The authors propose a three-layer IDS that uses a supervised approach to detect a cyber-attack on IoT networks.
2020	TON-IoT [22]	127	Normal, DDoS, DoS, Injection, MiM, Password, XSS, Scanning.	A telemetry dataset is proposed for both IoT and IIoT applications.
2020	IoT-23 [23]	21	C&C, DDoS, FileDownload, HeartBeat, Mirai, Okiru, PartOfAHorizontalPortScan, Torii.	The IoT-23 dataset consists of twenty-three attacks scenarios of different IoT network traffic created by Avast AIC laboratory.
2021	HIKARI [24]	86	Background, Benign, Bruteforce, Bruteforce-XML, Probing, XMRIGCC CryptoMiner.	The dataset contains benign and encrypted network traffic.

(Continued)

Table 1 (continued)

Years	Dataset name	Features number	Attack classes	Description
2022	Edge-IIoTset [25]	61	DoS/DDoS attacks, Information gathering, MiM attacks, Injection attacks, Malware attacks.	The generated Edge-IIoTset datset could be used by an IDS based on centralized and federated learning.
2022	IoT Linux [26]	–	Simulation of random attacks.	Authors publish a gathered dataset for Linux-based IoT devices.
2023	CICIoT2023 [27]	47	Benign, DDoS, DoS, Reconnaissance, Web-based, brute force, spoofing, and Mirai.	Seven attack classes are performed to generate the CICIoT dataset.

The HiTar-2024 stands out from existing intrusion detection datasets by focusing specifically on IIoT manufacturing scenarios using the AREZZO simulator to generate realistic and detailed traffic logs. Unlike older datasets like KDD CUP 99 [17] and NSL-KDD [18], which are outdated and lack relevance to modern IIoT environments, or more general-purpose datasets like CICIDS2017 [19] and BoT-IoT [28], HiTar-2024 addresses unique industrial challenges by categorizing attacks into five IIoT-relevant classes: Normal, Probing, R2L, U2R, and DoS. It bridged the gaps left by other datasets, such as limited focus on industrial applications, outdated traffic patterns, or insufficient granularity, making it a critical resource for developing robust and realistic intrusion detection systems tailored to smart manufacturing environments.

3 IIoT Environment Description

Alternative simulation tools such as NS-3 and OMNeT++ are commonly used for network simulation but suffer from a lack of industrial-specific scenario simulation required for IIoT environments. For this reason, AREZZO was chosen for its high flexibility, realism, and ability to accurately model industrial network traffic, making it ideal for simulating smart manufacturing scenarios and generating realistic benign and malicious traffic for HiTar-2024.

AREZZO is a project created by LAMIH Lab [29], and it is defined as an emulator of a flexible manufacturing system (FMS) satisfying to the "Bench4star" benchmark. The latter is composed of two main components, including "Emulator", which emulates a cell and "Pilot", which permits handling the emulator through client-server communication, as shown in (Fig. 2a and b).

AREZZO is a simulator build to simulate industrial components and process representing a complete manufacturing environment. It incorporates the monorail conveying system with eleven transfer blocks allowing the flexible routing of up to fifteen identified shuttles via a radio frequency identification (RFID) tagged across seven workstations in charge of achieving particular operations such as robotic assembly and monitor on manufactured products. Each product is linked with a dedicated shuttle during its manufacturing cycle and has to be conveyed on this shuttle from one station to another, respecting a specific plan.

Fig. 3a depicts a view of the real set up composed of workstations and their relative locations, Fig. 3b presents the automation network architecture.

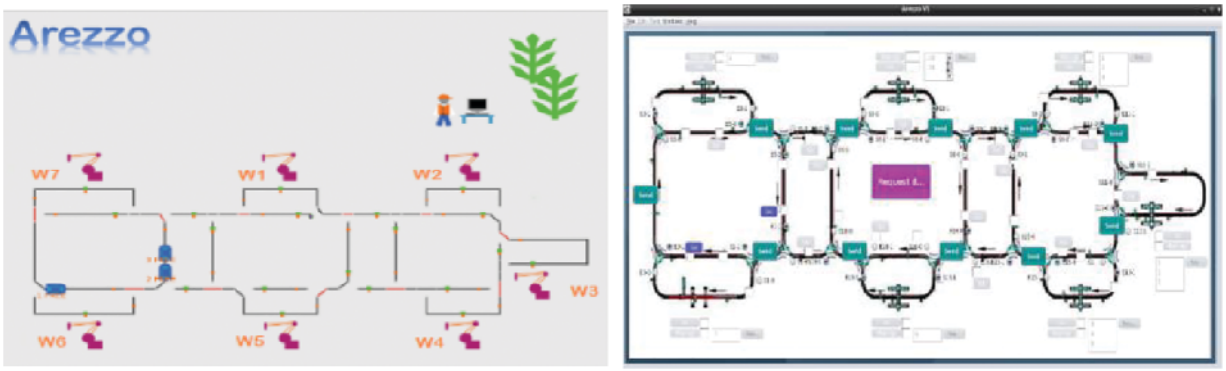

Figure 2: The AREZZO Tool: (a) Arezzo emulator, (b) Arezzo pilot

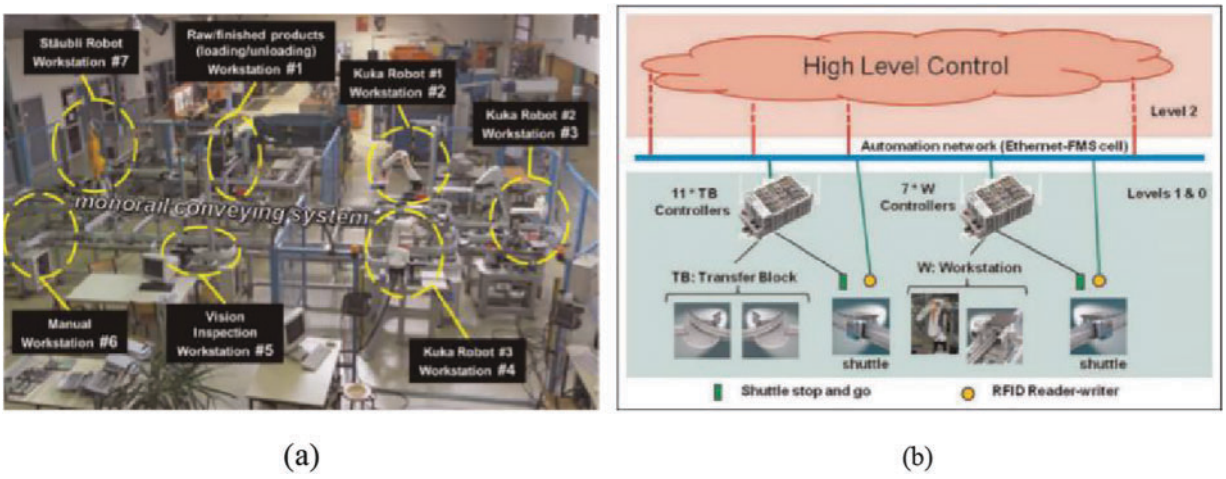

Figure 3: (a) General view of workstations and their locations, (b) Aip-Primeca cell physical architecture

According to Fig. 3b, levels L0 and L1 called also shop floor are composed of controllers, workstations, shuttle stop and go, RFID Reader/Write device, and an Ethernet-switched automation network [30]. The physical topology used to connect all equipment is called Modbus through the automation network (Ethernet-FMS cell). The Ethernet-FMS cell involves up to 26 Modbus/Transport Control Protocol (Modbus/TCP) servers. On the other hand, level L2 is defined as the high-level control (HLC) layer. The latter level is responsible for resolving conflict among workstations when accessing the common Modbus and defining the routing of the work in progress over time through shuttles, leading to the good execution of the work according to the process plan on the available resources.

HiTar-2024 includes five classes: **Normal, Probing, R2L, U2R,** and **DoS**. Among these, **Probing, R2L, U2R,** and **DoS** are intrusion classes, representing malicious activities such as scanning, unauthorized access, privilege escalation, and service disruption. **The normal** class represents benign traffic.

4 Dataset Generation Process

Fig. 4 represents the proposed HiTar-2024 dataset generation process. The first step in the process starts with running an IIoT scenario in the AREZZO environment. The second step involves benign and attack

scenario generation. Benign scenario leads to benign LOG. However, the attack scenario leads to probing LOG, R2L LOG, U2R LOG and DoS LOG. The last step consists to group the various LOGs into one dataset, as shown in Fig. 4.

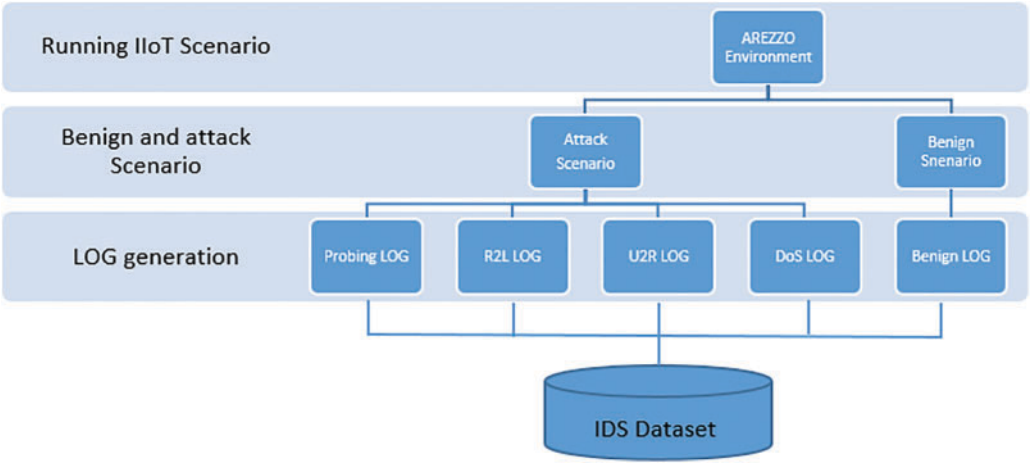

Figure 4: The proposed HiTar-2024 dataset generation process

The HiTar-2024 dataset is created using the AREZZO simulator, which provides a controlled, simulated environment for smart manufacturing scenarios. Despite the simulated setting, the interactions captured within this environment are based on real-world network behaviors and threats. This approach ensures that the dataset combines the benefits of a controlled experimental setup with the authenticity of real-world data. By doing so, it enhances the dataset's relevance and reliability for developing effective security measures in IIoT systems, highlighting its scientific and practical value.

As shown in Table 2, the HiTar-2024 dataset is composed of 39 extracted features compared to 7 extracted features in Hitar-2023 [31]. The HiTar-2024 dataset employs a comprehensive feature selection approach to accurately capture and label network traffic data. The dataset includes 39 attributes, each serving a specific purpose in identifying and classifying network behaviors. The Timestamp attribute records the connection time in seconds, while IP_SRC and IP_DST capture the source and destination IP addresses, respectively. PORT_SRC and PORT_DST denote the port numbers of the source and destination machines. The PROTOCOL attribute specifies the type of protocol used, such as TCP, UDP, or ICMP. The Flag attribute indicates the corresponding flag used in the connection, with additional Boolean attributes (e.g., f-SYN, f-SYNACK, f-ACK) representing specific flag conditions. The Service attribute identifies the service type, with Boolean attributes (e.g., sr_FTP, sr_SSH, sr_HTTP) indicating specific services. Finally, the Attack_type attribute labels the data as Normal, Probing, R2L, U2R, or DoS. This detailed feature selection ensures that the dataset captures all relevant aspects of network traffic, facilitating accurate detection and classification of various attack types.

Table 2 gives the attributes number, the attribute's name, the attribute's type and the attribute description for the HiTar-2024 dataset.

Table 2: HiTar-2024 dataset features description

Attribute's number	Attribute's name	Attribute's type	Description
1	Timestamp	Time	Connexion time in seconds
2	IP_SRC	Nominal	IP address version 4 of source workstation
3	IP_DST	Nominal	IP address version 4 of destination workstation
4	PORT_SRC	Integer	Port number of source workstation
5	PORT_DST	Integer	Port number of destination workstation
6	PROTOCOL	Nominal	Protocol's type (TCP, UDP, ICMP)
7	Flag	Nominal	The correspondent used flag
8	f-SYN	Boolean	Equals to 1 when the flag is [S] else 0
9	f-SYNACK	Boolean	Equals to 1 when the flag is [S.] else 0
10	f-ACK	Boolean	Equals to 1 when the flag is [.] else 0
11	f-PUSHACK	Boolean	Equals to 1 when the flag is [P.] else 0
12	f-FIN	Boolean	Equals to 1 when the flag is [F] else 0
13	f-FINACK	Boolean	Equals to 1 when the flag is [F.] else 0
14	f-RST	Boolean	Equals to 1 when the flag is [R] else 0
15	f-RSTACK	Boolean	Equals to 1 when the flag is [R.] else 0
16	f-NONE	Boolean	Equals to 1 when the flag is [NONE] else 0
17	f-FSPU	Boolean	Equals to 1 when the flag is [FSPU] else 0
18	f-SEW	Boolean	Equals to 1 when the flag is [SEW] else 0
19	f-SACKE	Boolean	Equals to 1 when the flag is [S.E] else 0
20	f-FPU	Boolean	Equals to 1 when the flag is [FPU] else 0
21	f-PACKU	Boolean	Equals to 1 when the flag is [P.U] else 0
22	Service	Nominal	The correspondent used service
23	f-other	Boolean	1 if other flag value
24	sr_RSH	Boolean	Equals to 1 when the Service is Remote shell else 0
25	sr_Rlogin	Boolean	Equals to 1 when the Service is Remote Login else 0
26	sr_Ruser	Boolean	Equals to 1 when the Service is Remote User else 0
27	sr_FTP	Boolean	Equals to 1 when the Service is FTP else 0
28	sr_SSH	Boolean	Equals to 1 when the Service is Secure shell else 0
29	sr_Telnet	Boolean	Equals to 1 when the Service is Telnet else 0
30	sr_SMTP	Boolean	Equals to 1 when the Service is SMTP shell else 0
31	sr_DNS	Boolean	Equals to 1 when the Service is DNS shell else 0
32	sr_DHCP	Boolean	Equals to 1 when the Service is DHCP else 0
33	sr_HTTP	Boolean	Equals to 1 when the Service is HTTP else 0
34	sr_HTTPS	Boolean	Equals to 1 when the Service is HTTPS else 0
35	sr_NTP	Boolean	Equals to 1 when the Service is NTP else 0
36	sr_Modbus	Boolean	Equals to 1 when the Service is Modbus else 0
37	sr_BGP	Boolean	Equals to 1 when the Service is BGP else 0
38	sr_Other	Boolean	Equals to 1 when the Service is unknown else 0
39	Attack_type	Nominal	The added label: Normal, Probing, R2L, U2R, DoS

5 HiTar-2024 Labelling Process

This part encapsulates the overall procedure of the construction of our **HiTar-2024** labelling process threw the presentation of the meticulous programming of the "Attack_labelling.sh" algorithm, which is a practical approach to network traffic analysis and intrusion detection, providing a robust method for identifying and classifying malicious network attacks.

The HiTar-2024 labelling process, as discussed above, is presented in Fig. 5.

The "Attack_labelling.sh" algorithm is a practical and technical approach to label the HiTar-2024 Dataset. It operates on AREZZO's LOG file, which comprises network traffic data from benign packets and malicious one as well. The output is a CSV file that encapsulates the HiTar-2024 dataset.

The algorithm initiates by defining the path to AREZZO's LOG file and the output file. Subsequently, a function named "identify_attack" is defined. This function ingests a line from the LOG file as input and yields a labelled line indicating the attack type or class. The generated IDS outputs include five classes such as Normal_behaviour, Probing, R2L, U2R and DoS attacks.

The "identify_attack" function operates by extracting pertinent fields from the LOG line, such as TIME, IP_SRC, IP_DST, PORT_SRC, PORT_DST, PROTOCOL, and flag. It then defines the flag types and service types and assigns values to them. The function then scrutinizes conditions to identify the type of attack or classify it as normal behaviour. These conditions are defined for normal behaviour, probing, R2L, U2R, and DoS attacks.

The algorithm then processes each line of the LOG file. It reads each line, invokes the "identify_attack" function to label the line, and writes the labelled line to the output file. Upon completion of attack labelling, the algorithm displays a message "Attack labelling completed".

It's important to note that the "Attack_labelling.sh" algorithm is developed using Bash shell scripting. As shown in Fig. 5, the HiTar-2024 labelling process involves five steps as follows:

- The first step specifies the path to Arezzo's LOG file containing both normal and attack network traffic.
- The second step specifies the path to the output file where the attack labelling will be saved.
- The third step extracts relevant features from LOG file and writes on the Dataset generated.
- The fourth step calls **identify_attack**() function to check conditions for identifying the attack class.
- The fifth step adds the labelled line containing the attack type according to features gathered.

Start Algorithm: **Attack_labelling.sh** for labelling the HiTar-2024 Dataset
Input:
- Arezzo's log file containing collected network traffic data from both benign & malicious profiles
Output:
- output csv file for storing the HiTar-2024 dataset csv file
Procedure:
I. Specify the path to the Arezzo's log file containing both normal and attack network traffic.
II. Specify the path to the output file where the attack labelling will be saved.
For each instance in the log file do
III. Define a function called "identify_attack" to check for specific attack types and add labels based on predefined conditions.
Function **identify_attack**():
{
Input: A line from the log file
Output: A labelled line indicating the attack type
1. Extract relevant features from the log line (TIME, IP_SRC, IP_DST, PORT_SRC, PORT_DST, PROTOCOL, flag)
2.
a. Defining the flags types (f-SYN, f-SYNACK, f-ACK, f-PUSHACK, f-FIN, f-FINACK, f-RST, f-RSTACK, f-NONE, f-FSPU, f-SEW, f-SACKE, f-FPU, f-PACKU)
b. Attributing the values for the flag's types.
3.
a. Defining the services types (Service, sr_FTP, sr_SSH, sr_Telnet, sr_SMTP, sr_DNS, sr_DHCP, sr_HTTP, sr_HTTPS, sr_NTP, sr_Modbus, sr_BGP)
b. Attributing the values for the services types.
4. Check conditions to identify the type of attack or classify as normal behaviour:
a. Defining conditions for normal behaviour:
a.1. Port Condition: The Port Source or Port Destination must be the privileged port 502, as Arezzo uses the MODBUS protocol.
a.2. IP Condition: Both IP Source and IP Destination must be within the network and have specific IPs allowed from the AREZZO's INTERFACE folder.
a.3 Flags Condition: The flags used must be the same as those used during AREZZO's normal traffic.
b. Defining conditions for Probing attacks.
b.1. Flag Condition: The flag must be either [echo], [S], [R.], [none], or [FPU].
c. Defining conditions for R2L.
c.1. Port and Flag Condition: The ports used as Port Source or Port Destination must be specified depending on the service used, and flags must be different from [S], echo, or [R.].
c.2. Specific Port Condition: The flag must be [S.] and the Port Source or Port Destination must be port 21 or 533.
c.3. Unprivileged Ports Condition: The flag must be [S.] and the Port Source or Port Destination must be unprivileged ports.
d. Defining conditions for U2R.
d.1. Port and Flag Condition: Using non-privileged and privileged ports, and the flag is other than "length".
d.2. Specific Port Condition: Using ports 20 and 21, and the flags are other than [S.], [S], [R.], or [R].
e. Defining conditions for DoS.
e.1. Flooding Condition: When port 502 is flooded with the flags [F], [R], [.], or [S].
e.2. Reachability Condition: When the reachability field is "unreachable".
e.3. Specific Flag Condition: Specifically, when the flag field is "length".
5. Write the labelled line to the output file based on identified conditions.
}
End do

Figure 5: The HiTar-2024 labelling process description

Building HiTar-2024 involved significant challenges, such as accurately simulating IIoT scenarios in the AREZZO simulator to reflect real-world manufacturing environments, ensuring sufficient data diversity to cover a wide range of attack types, and generating realistic attack traffic while minimizing noise in the logs. These challenges were addressed by leveraging AREZZO's advanced simulation capabilities, employing meticulous data labelling processes, and designing scenarios that capture complex IIoT interactions. Detailed validation steps ensured the dataset's reliability and applicability, making it a robust resource for IDS training.

6 Results and Discussion

The process used to perform the various ML metrics is shown in Fig. 6. Therefore, the life cycle of ML evaluation metrics involves 6 steps. The first step is the selection of the generated dataset. The second step is the pre-processing. Then, the selection of the classifier model is followed by the model training and test phase. The last step is the generation of ML performance evaluation metrics. In the pre-processing step, the analyst removes the duplication and adds the missing information. Also, in test phase subsets are evaluated with 10 folds cross-validation on the training model.

Figure 6: ML metrics evaluation life cycle

In fact, the different steps requested for simulation experiments in the Weka tool to carry out performance metrics based on various classifiers are defined as follows:

1. The first step is started with uploading the HiTar-2024 dataset.
2. The second step is called data pre-processing.
3. The third step is to select the classifier family and sub-classifier type. Then, configure the classifier parameters.
4. Model construction via training.
5. Test phase by selecting test options. The cross-validation option is selected with 10 folds as parameter.
6. Start the simulation to generate predictive accuracy and other performance evaluation metrics.

However, Class imbalance is a common issue in ML, where certain classes are underrepresented compared to others, leading to biased models that perform poorly on minority classes. This is particularly relevant for the HiTar-2024 dataset, which includes various attack types with potentially imbalanced distributions. To address this, techniques like SMOTE (Synthetic Minority Over-sampling Technique) can be used to balance the dataset by generating synthetic instances for the minority classes. Additionally, employing 10-fold cross-validation helps ensure robust model evaluation. In this method, the dataset is divided into 10 subsets, and the model is trained and validated 10 times, each time using a different subset as the validation set and the remaining subsets for training. This approach provides a more reliable estimate of the model's performance by reducing the variance associated with a single train-test split, ensuring that the model generalizes well to unseen data.

In this consolidated evaluation phase, the Weka tool serves as the linchpin for a comprehensive comparative analysis of ML classifiers across diverse families, including Bayesian, Functions, Tree, Lazy, Meta, and Rules. The evaluation focuses on specific classifiers within these families: BayesNet, Logistic, J48, IBK, Multiclass Classifier, and PART.

The performance of a classification model in ML is commonly evaluated using specific measures and metrics: our assessment for the HiTar-2024 dataset effectiveness incorporated key metrics like accuracy, precision, recall, F1-score, the Receiver Operating Characteristic Area, the Precision Recall Curve, Kappa Statistic, Time Taken to Build Model and Mean Absolute Error outlined as follows:

True Positive (TP): corresponds to cases where the model correctly predicted positive examples.

False Positive (FP): matches to cases where the model incorrectly predicted examples as positive when they are negative in the label.

False Negative (FN): matches to cases where the model incorrectly predicted examples as negative when, they are positive.

True Negative (TN): corresponds to cases where the model correctly predicted negative points.

Accuracy: measures the total number of instances correctly classified by the model relative to the total number of instances as in (1).

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \times 100 \tag{1}$$

Precision: measures the accuracy of positive predictions made by the model as in (2).

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \times 100 \tag{2}$$

Sensitivity (also known as Recall): measures the model's capacity to capture all TP instances as in (3).

$$Sensitivity = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \times 100 \tag{3}$$

F-measure (also known as F1-score): defined as the relationship between the inverse of precision and recall, providing a balance between the two metrics as in (4).

$$F - measure = \frac{2\mathrm{xTP}}{2\mathrm{xTP} + \mathrm{FP} + \mathrm{FN}} \times 100 \tag{4}$$

Matthews Correlation Coefficient (MCC): defined as the interdependence between predicted and actual values in the dataset. It considers all elements of the confusion matrix to assess the overall quality of a classification model as in (5).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt[2]{(TP + TN) \times (TP + FN) \times (TN + FP) \times (FP + FN)}} \times 100 \tag{5}$$

Receiver Operating Characteristic (ROC): represents a model's performance at various probability thresholds by plotting the TP rate *vs.* the FP rate.

Precision Recall Curve (PRC): illustrates the relationship between precision and recall for different probability thresholds.

Kappa Statistic (KS): measures the agreement between observed and expected classifications, considering the possibility of agreement occurring by chance.

Time Taken to Build Model (TTBM): measures the duration, in seconds, required for a ML model to be constructed during the training phase.

Mean Absolute Error (MAE): calculates the average absolute error among actual and predicted values as in (6).

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \widehat{y_i} | \tag{6}$$

where:

n is the number of points.

y_i is the actual value for the i-th point.

$\widehat{y_i}$ is the predicted value for the i-th point.

Table 3 provides a comprehensive overview of the performance metrics for various classifiers within distinct families, as analysed through the powerful WEKA tool. These classifiers are evaluated based on key metrics, which collectively offer valuable insights into the effectiveness and efficiency of each classifier, aiding in the informed selection of models tailored to specific data characteristics and classification goals.

Table 3: Performance Analysis with various classifiers

Family	Bayesian	Functions	Lazy	Meta	Rules	Tree
Classifier	Bayesnet	Logistic	IBK	Multiclass	PART	J48
Accuracy	99.00	99.96	99.99	99.95	99.90	99.96
Precision	0.991	1.000	1.000	1.000	0.999	1.000
Recall	0.990	1.000	1.000	1.000	0.999	1.000
F1-score	0.990	1.000	1.000	1.000	0.999	1.000
MCC	0.990	0.999	1.000	0.999	0.998	0.999
ROC Area	1.000	1.000	1.000	1.000	1.000	1.000
PRC Area	0.999	1.000	1.000	0.999	1.000	1.000
KS	0.9803	0.9992	0.9999	0.9991	0.9981	0.9992
TTBM (seconds)	0.46	8.33	–	13.51	1.44	0.5
MAE	0.0004	0.0002	0.0001	0.0004	0.0006	0.0003

This evaluation is conducted within the specific context of validating the generated HiTar-2024 dataset.

Interpretation of the classifiers' performances: in the rigorous evaluation of classifiers across distinct families using the given dataset, several noteworthy observations emerge:

The default parameters for the used classifiers in performing the performance analysis are like so:

BayseNet Classifier: -D -Q weka.classifiers.bayes.net.search.local.K2-P1 -S BAYES -E weka.classifiers.bayes.net.estimate.SimpleEstimator – -A 0.5

In the BayesNet classifier, debugging information is enabled with the -D parameter, while the search algorithm is specified as weka.classifiers.bayes.net.search.local.K2, a local search algorithm using the K2 heuristic, through the -Q parameter. The number of parents each node can have is set to one with -P1, and the scoring function is set to Bayes with -S BAYES. The estimator algorithm used is weka.classifiers.bayes.net.estimate.SimpleEstimator, as specified by -E, and the alpha value for estimating probabilities is set to 0.5 with – -A 0.5.

Logistic Classifier: -R 1.0E- 8 -M-1 -num-decimal-places 4.

For the Logistic classifier, the ridge in the log-likelihood is set at 1.0E-8 with -R, and the maximum number of iterations is unlimited, as indicated by -M -1. Numeric output is formatted to four decimal places with -num-decimal-places 4.

IBK Classifier: -K 1 -W 0 -A "weka.core.neighbourseach.LinearNNSearch -A\"

The IBK classifier, which uses k-nearest neighbors, sets the number of nearest neighbors to one with -K 1 and uses equal weighting as denoted by -W 0. The nearest neighbor search algorithm and distance measure are specified as LinearNNSearch and EuclideanDistance respectively, with -A "weka.core.neighbourseach.LinearNNSearch -A\" weka.core.EuclideanDistance -R first-last\"".

MultiClass Classifier: -M 0 -R 2.0 -S 1 -W weka.classifiers.functions.Logistic – -R 1.0E-8 -M -1 -num-decimal-places4

The MultiClassClassifier sets no method with -M 0, uses a random number seed of two with -R 2.0, and handles multi-class problems with scheme one as indicated by -S 1. It specifies Logistic as the base classifier along with its options using -W weka.classifiers.functions.Logistic – -R 1.0E-8 -M -1 -num-decimal-places4.

weka.core.EuclideanDistance -R first-last\""

PART Classifier: -C 0.25 -M 2/**J48 Classifier:** -C 0.25 -M 2

Both PART and J48 classifiers use a confidence threshold for pruning set at 0.25 with -C and require a minimum of two instances per rule or leaf, as specified by -M 2.

These parameters are adjusted to ensure that each classifier operates efficiently within the analytical framework, optimizing performance on the dataset.

Bayesian (BayesNet Classifier): exhibits strong performance with an accuracy of 99.00% and consistent precision, recall, F1-score, MCC, and ROC Area values. While it demonstrates reliability across metrics, it exhibits a slightly lower Ks statistic compared to other classifiers.

Functions (Logistic Classifier): stands out with an exceptional accuracy of 99.96% and perfect precision 100%, recall, F1-score, MCC, ROC Area, and PRC area values. Its performance underscores its effectiveness across various evaluation criteria.

Lazy (IBK Classifier): Demonstrates outstanding performance with the highest accuracy of 99.99% among all classifiers. It achieves perfect precision, recall, F1-score, MCC, ROC Area, and PRC area values of 100%, highlighting its robust classification capabilities.

Meta (Multiclass Classifier): shows strong performance with an accuracy of 99.95% and excellent precision, recall, F1-score, MCC, and ROC Area values. Its performance is consistently high across most evaluation metrics, emphasizing its reliability.

Rules (PART Classifier): while slightly trailing in overall accuracy at 99.90%, the Rules classifier contributes valuable insights, particularly in terms of TTBM and MAE. It demonstrates strong precision, recall, F1-score, and ROC Area values, showcasing its efficacy in classification tasks.

Tree (J48 Classifier): similarly to Logistic, the Tree classifier achieves an accuracy of 99.96% and perfect precision 100%, recall, F1-score, MCC, ROC Area, and PRC area values. It stands out as a robust performer across various evaluation criteria.

In summary, all classifiers exhibit exceptional performance, each with its own strengths and contributions to the classification task. While Lazy (IBK) appears as the top performer in terms of accuracy, other classifiers such as Functions (Logistic), Meta (Multiclass), and Tree (J48) also demonstrate remarkable

capabilities across multiple metrics. These insights can guide practitioners in selecting the most appropriate classifiers based on specific requirements and priorities in their applications.

This comprehensive and nuanced analysis provides a clear understanding of the diverse strengths and limitations of each classifier within the specific context of the HiTar-2024 dataset, offering valuable guidance for informed model selection.

The HiTar-2024 dataset, enriched with realistic IIoT scenarios, plays a crucial role in enhancing the effectiveness of classifiers in modelling intricate relationships within the dataset. Its inclusion of a diverse set of attack scenario presents a formidable challenge to classifiers, pushing them to excel in detecting various cyber threats with remarkable precision, recall, and F1-score values. The dataset's deliberate incorporation of complex attack instances further empowers classifiers to discern intricate patterns, culminating in superior anomaly detection performance. In summary, the classifiers' outstanding success on the HiTar-2024 dataset underscores its significance in furnishing a challenging yet representative environment for training and evaluating ML models. The realism and diversity of attack instances encapsulated in HiTar-2024 substantially contribute to the classifiers' ability to achieve exceptional results in the realm of industrial network security.

The detailed performance results subsequently highlight the robustness of the used classifiers: Lazy (IBK) appears as the top-performing classifier with an exceptional accuracy of 99.99%, making it the standout choice for classification tasks. Functions (Logistic) follows closely with an accuracy of 99.96% and perfect precision; logistic exhibits remarkable classification prowess, particularly in terms of precision and recall. Tree (J48) demonstrates robust performance with an accuracy of 99.96%; its classification capabilities make it a prime choice for accurate attack detection. Meta (Multiclass) exhibits strong performance with an accuracy of 99.95%; Meta consistently impresses with its classification metrics. On the other hand, Rules (PART) is slightly lower in overall accuracy at 99.90%, also, it provides valuable insights and contributes valuable information for model selection and analysis. Bayesian (Bayesnet) demonstrates a slightly lower accuracy at 99.00% and shows reliability across metrics but shows room for improvement compared to other classifiers.

This comprehensive evaluation reveals the diverse strengths of each classifier within the HiTar-2024 dataset context, aiding practitioners in making informed choices for effective attack detection. The analysis, conducted with the Weka tool, provides valuable insights for selecting the most suitable algorithm for applications leveraging the HiTar-2024 dataset in industrial settings. This holistic evaluation, facilitated by the Weka tool, not only validates the robustness of classifiers in the face of the complex challenges posed by HiTar-2024 but also serves as a guiding compass for practitioners. The realism and diversity encapsulated in HiTar-2024 prove pivotal in advancing the capabilities of classifiers, heralding a new era of intelligent and adaptive industrial network security solutions tailored to real-world industrial complexities.

In our rigorous exploration of the HiTar-2024 dataset, which presents complex IIoT scenarios, we meticulously calibrated our classifiers to ensure robust performance without succumbing to overfitting. We employed 10-fold cross-validation for test options that partitions the data into 10 subsets or folders, training the model on 9 subsets while testing on the remaining subset allowing the model to demonstrate its predictive prowess on previously unseen data.

Furthermore, we embraced model simplicity as a virtue, selecting algorithms that are inherently less prone to overfitting due to their straightforward nature. The construction of these models was executed with remarkable efficiency, with none exceeding a 14-s build time threshold, thus exemplifying the synergy between expeditious model development and their inherent simplicity.

In our analytical endeavor, we took the helm in executing a comprehensive data analysis, spearheading the attribute selection process with precision and expertise. We meticulously sifted through the dataset to

identify and retain the most predictive features. This hands-on approach not only bolstered model interpretability but also fortified its generalization capabilities. Through this discerning process, we systematically eradicated redundant or irrelevant attributes, thereby purifying the feature space. This purification was instrumental in mitigating the risk of model overfitting and ensuring that our classifiers remained attuned to the underlying patterns within the HiTar-2024 dataset without being misled by noise or spurious correlations.

In conclusion, our methodical approach to analyzing the HiTar-2024 dataset for IIoT scenarios involved strategic measures to prevent overfitting. By implementing the 10-fold cross-validation to ensure robust validation across unseen data, prioritizing model simplicity to reduce complexity, and conducting a thorough attribute selection to focus on the most predictive features. These steps collectively enhanced model accuracy and generalizability, ensuring our classifiers are both efficient and effective in discerning true patterns from noise, which led us to successfully overcome the overfitting problem.

The KDD99, UNSW-NB15, CICIDS-2017, and our proposal through the HiTar-2024 dataset are often to be used for network intrusion detection research, but they differ in key aspects, as shown in Table 4. In terms of the number of unique IP addresses, KDD99 has the fewest (i.e., 11), while CICIDS-2017 leads with 16,960, followed by UNSW-NB15 (i.e., 45) and HiTar-2024 (i.e., 20). All datasets, except CICIDS-2017, are fully simulated; CICIDS-2017 is only partially simulated. The data collection format also varies, with KDD99 and HiTar-2024 using TCPDUMP, while UNSW-NB15 and CICIDS-2017 rely on PCAP files. For feature extraction, KDD99 uses the Bro-IDS tool, UNSW-NB15 employs Argus and Bro-IDS, CICIDS-2017 uses CICFlowmeter, while HiTar-2024 has a custom script called attack_labelling.sh.

Table 4: Comparison of HiTar-2024 dataset with other similar datasets in the literature

	Parameters	KDD [17]	UNSW-NB15 [13]	CICIDS [19]	HiTar-2024
	Number of unique IP address	11	45	16,960	20
	Simulation	Yes	Yes	Partial	Yes
	Format of the data collected	Tcpdump, BSM, dumpfile	pcap files	pcap files	Tcpdump
	Feature extraction tools	Bro-IDS tool	Argus, Bro-IDS	CICFlowmeter	attack_labelling.sh
	Number of features	42	42	80	39
	Attack categories	4	9	7	5
	Numbers of instances	825,050	175,341	225,745	15,842
	Domain of application	IoT	IoT	IoT	IIoT
	Accuracy %	NA	84.11	99.00	99.9937
	Precision %	NA	78.34	99.00	100
MLP	Recall %	NA	98.31	99.00	100
	F1-score %	NA	87.20	99.00	100
	FPR %	NA	33.28	NA	0

Feature count differs among datasets, with CICIDS-2017 providing the most features equal to 80, compared to 42 for KDD99 and UNSW-NB15, and 39 for HiTar-2024. Regarding attack categories, UNSW-NB15 has the highest number of attacks with nine, while KDD99 has four, CICIDS-2017 has seven, and HiTar-2024 has five. KDD99 has the greatest number of instances, equal to 825,050, with HiTar-2024 having the fewest, equal to 15,842. All datasets focus on IoT, except HiTar-2024, which is designed for IIoT applications.

The ML performance metrics for the Multi-Layer Perceptron (MLP) model indicate that HiTar-2024 demonstrates excellent accuracy (99.9937%), precision (100%), recall (100%), and F1-score (100%), with a 0% FP rate. These figures surpass UNSW-NB15, which has an accuracy of 84.11%, a precision of 78.34%, recall of 98.31%, an F1-score of 87.20%, and an FP rate of 33.28%. While CICIDS-2017 shows high accuracy, precision, recall, and F1-score (all at 99%), it lacks data on FP rates. Overall, HiTar-2024 stands out for its

near-perfect MLP metrics and low FP rate, although it has fewer instances and attack categories compared to other datasets, suggesting that HiTar-2024 excels in specific performance aspects.

7 Conclusion

The HiTar-2024 dataset is a groundbreaking contribution to IIoT security, particularly notable for its innovative approach to attack labelling. Developed using the AREZZO simulator, the dataset captures real-world network behaviors within a controlled environment, ensuring both authenticity and replicability. One of the key novelties of the HiTar-2024 dataset is its detailed attack labelling, which categorizes network traffic into five distinct classes: Normal, Probing, R2L, U2R, and DoS. This comprehensive labelling is achieved through the use of a sophisticated script, "attack_labelling.sh", which meticulously analyses network traffic based on specific conditions and flags. The use of various supervised ML classifiers further enhances the dataset's utility exploited by an IDS during the training process. Additionally, the HiTar-2024 dataset is compared with other similar datasets in terms of data format, feature extraction tools, number of features, attack categories, number of instances, and ML metrics. This comprehensive approach underscores the scientific rigor and practical value of the HiTar-2024 dataset, providing critical insights for developing robust security measures in IIoT systems.

The performance analysis carried out based on the generated HiTar-2024 reveals notable accuracy levels. Within the Bayesian family, IBK classifier achieves the highest accuracy of 99.99%. For the J48-based tree classifier, it stands out with an accuracy of 99.96%. The Logistic-based functions classifier follows closely with an accuracy of 99.96%. Regarding Meta based Multiclass classifier: Demonstrates strong performance with an accuracy of 99.95%. Finally, the Rules-based PART family classifier maintains a respectable accuracy of 99.90%, and the Bayesian-based Bayesnet family classifier achieves an accuracy of 99.00%.

The high performance of ML models on HiTar-2024, as reflected in Tables 3 and 4 with metrics such as accuracy, precision, and recall exceeding 0.99, indicates the dataset's quality and the robustness of current algorithms due to a rigorous process, including data pre-processing, features extraction, fixing the class imbalance problem and using a careful cross-validation test option for model robustness. However, this should not suggest that intrusion detection in IIoT systems is an inherently easy problem. The evolving nature of cyber threats, the variability in industrial environments, and the sophistication of modern attacks present significant challenges. To address these complexities, future research should explore solutions for detecting unknown or zero-day attacks, which remain a critical vulnerability. Additionally, real-time detection methods should be prioritized to ensure timely responses in dynamic IIoT ecosystems. Finally, techniques such as federated learning can also be applied to enhance scalability and privacy in distributed industrial systems, paving the way for more adaptive and resilient IDS in IIoT environments.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Tarak Dhaouadi, Hichem Mrabet; data collection: Tarak Dhaouadi; analysis and interpretation of results: Hichem Mrabet, Abderrazak Jemai; draft manuscript preparation: Hichem Mrabet, Adeeb Alhomoud. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in "HiTar-2024 dataset" at https://github.com/H-Mrabet/HiTar-2024-dataset (accessed on 12 January 2025).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Nomenclature

CAIDA	Center for Applied Internet Data Analysis
UNSW-NB15	University of New South Wales-Network-Based 15
KDD99	Knowledge Discovery in Databases 1999
NSLKDD	Network Security Laboratory-KDD
CICIDS17	Canadian Institute for Cybersecurity Intrusion Detection System 2017
IoTHIDS	Internet of Things Host-based Intrusion Detection System
IoT-SH	Internet of Things-Smart-Home
TON-IoT	Telemetry, Operating Systems, and Network-Internet of Things
Edge-IIoTset	Edge Computing-Industrial Internet of Things Dataset

References

1. Xu LD, Xu EL, Li L. Industry 4.0: state of the art and future trends. Int J Prod Res. 2018;56(8):2941–62. doi:10.1080/00207543.2018.1444806.
2. Ustundag A, Cevikcan E. Industry 4.0: managing the digital transformation. Cham: Springer International Publishing; 2018. doi:10.1007/978-3-319-57870-5.
3. Nargiza A, Ransomware. Analysis of 2019 LockerGoga cyber-attack to Norsk Hydro multinational company and its countermeasures. Eurasian J Med Commun. 2022;9:1–9.
4. Wali A, Mrabet H, Jemai A. A secure IoT architecture for industry 4. 0. In: Mosbah M, Kechadi T, Bellatreche L, Gargouri F, Guegan CG, Badir H et al., editors. Advances in model and data engineering in the digitalization era. MEDI 2023. Communications in computer and information science. vol. 2071. Cham: Springer; 2024. p. 210–23. doi:10.1007/978-3-031-55729-3_17.
5. Umran SM, Lu S, Ameen Abduljabbar Z, Tang X. A blockchain-based architecture for securing industrial IoTs data in electric smart grid. Comput Mater Contin. 2023;74(3):5389–416. doi:10.32604/cmc.2023.034331.
6. Hasan MR, Alazab A, Joy SB, Uddin MN, Uddin MA, Khraisat A, et al. Smart contract-based access control framework for Internet of Things devices. Computers. 2023;12(11):240. doi:10.3390/computers12110240.
7. Makhdoom I, Zhou I, Abolhasan M, Lipman J, Ni W. PrivySharing: a blockchain-based framework for privacy-preserving and secure data sharing in smart cities. Comput Secur. 2020;88(3):101653. doi:10.1016/j.cose.2019.101653.
8. Adnan Hussain H, Mansor Z, Shukur Z, Jafar U. Ether-IoT: a realtime lightweight and scalable blockchain-enabled cache algorithm for IoT access control. Comput Mater Contin. 2023;75(2):3797–815. doi:10.32604/cmc.2023.034671.
9. Vourganas IJ, Michala AL. Applications of machine learning in cyber security: a review. J Cybersecur Priv. 2024;4(4):972–92. doi:10.3390/jcp4040045.
10. Yavanoglu O, Aydos M. A review on cyber security datasets for machine learning algorithms. In: 2017 IEEE International Conference on Big Data (Big Data); 2017 Dec 11–14; Boston, MA, USA: IEEE; 2017. p. 2186–93. doi:10.1109/BigData.2017.8258167.
11. Shaukat K, Luo S, Varadharajan V, Hameed IA, Xu M. A survey on machine learning techniques for cyber security in the last decade. IEEE Access. 2020;8:222310–54. doi:10.1109/ACCESS.2020.3041951.
12. Mohd Yusof NN, Sulaiman NS. Cyber attack detection dataset: a review. J Phys: Conf Ser. 2022;2319(1):012029. doi:10.1088/1742-6596/2319/1/012029.
13. Al-Daweri MS, Zainol Ariffin KA, Abdullah S, Md Senan MFE. An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. Symmetry. 2020;12(10):1666. doi:10.3390/sym12101666.
14. Sasi T, Lashkari AH, Lu R, Xiong P, Iqbal S. A comprehensive survey on IoT attacks: taxonomy, detection mechanisms and challenges. J Inf Intell. 2024;2(6):455–513. doi:10.1016/j.jiixd.2023.12.001.
15. Bouazza A, Debbi Lakhlef H. Machine learning-based intrusion detection system against routing attacks in the Internet of Things. In: Tunisian-Algerian Joint Conference on Applied Computing (TACC 2022); 2022 Dec 13–14; Constantine, Algeria.

16. Lippmann RP, Fried DJ, Graf I, Haines JW, Kendall KR, McClung D, et al. Evaluating intrusion detection systems: the 1998 *DARPA* off-line intrusion detection evaluation. In: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00; 2000 Jan 25–27; Hilton Head, SC, USA: IEEE; 2000. p. 12–26. doi:10.1109/DISCEX.2000.821506.
17. Mishra P, Varadharajan V, Tupakula U, Pilli ES. A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun Surv Tutor. 2019;21(1):686–728. doi:10.1109/COMST.2018.2847722.
18. Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications; 2009 Jul 8–10; Ottawa, ON, Canada: IEEE; 2009. p. 1–6. doi:10.1109/CISDA.2009.5356528.
19. Sharafaldin I, Habibi Lashkari A, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of the 4th International Conference on Information Systems Security and Privacy; 2018 Jan 22–24; Funchal, Madeira, Portugal: SCITEPRESS-Science and Technology Publications; 2018. p. 108–16. doi:10.5220/0006639801080116.
20. Bezerra VH, da Costa VGT, Martins RA, Barbon S, Miani RS, Zarpelão BB. Providing IoT host-based datasets for intrusion detection research. In: Anais do XVIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg 2018). Brasil. Sociedade Brasileira de Computação—SBC; 2018. p. 15–28. doi:10.5753/sbseg.2018.4240.
21. Anthi E, Williams L, Słowińska M, Theodorakopoulos G, Burnap P. A supervised intrusion detection system for smart home IoT devices. IEEE Internet Things J. 2019;6(5):9042–53. doi:10.1109/JIOT.2019.2926365.
22. Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A. TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access. 2020;8:165130–50. doi:10.1109/ACCESS.2020.3022862.
23. Garcia S, Parmisano A, Erquiaga MJ. IoT-23: a labelled dataset with malicious and benign IoT network traffic (Version 1.0.0). Zenodo. 2020. doi:10.5281/zenodo.4743746.
24. Ferriyan A, Thamrin AH, Takeda K, Murai J. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. Appl Sci. 2021;11(17):7868. doi:10.3390/app11177868.
25. Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access. 2022;10:40281–306. doi:10.1109/ACCESS.2022.3165809.
26. Adamczyk B, Brzeczek M, Michalak M, Kostorz I, Wawrowski L, Hermansa M, et al. Dataset generation framework for evaluation of IoT linux host-based intrusion detection systems. In: IEEE International Conference on Big Data (Big Data); 2022 Dec 17–20; Osaka, Japan: IEEE; 2022. p. 6179–87. doi:10.1109/bigdata55660.2022.10020442.
27. Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors. 2023;23(13):5941. doi:10.3390/s23135941.
28. Ashraf J, Keshk M, Moustafa N, Abdel-Basset M, Khurshid H, Bakhshi AD, et al. IoTBoT-IDS: a novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. Sustain Cities Soc. 2021;72(10):103041. doi:10.1016/j.scs.2021.103041.
29. Laboratory of industrial and human automation control, mechanical engineering and computer science (LAMIH) [cited 2024 Apr 20]. Available from: http://www.uphf.fr/lamih/.
30. Berger T, Deneux D, Bonte T, Cocquebert E, Trentesaux D. Arezzo-flexible manufacturing system: a generic flexible manufacturing system shop floor emulator approach for high-level control virtual commissioning. Concurr Eng. 2015;23(4):333–42. doi:10.1177/1063293X15591609.
31. Dhaouadi T, Mrabet H, Jemai A. The HiTar-23 dataset construction and validation for securing industrial Internet of Things environment. In: 2024 IEEE 27th International Symposium on Real-Time Distributed Computing (ISORC); 2024 May 22–25; Tunis, Tunisia: IEEE; 2024. p. 1–6. doi:10.1109/ISORC61049.2024.10551372.