Zero-Day Guardian: A Dual Model Enabled Federated Learning Framework for Handling Zero-Day Attacks in 5G Enabled IIoT

Priyanka Verma, *Senior Member, IEEE*, Nitesh Bharot, *Senior Member, IEEE*, John G. Breslin, *Senior Member, IEEE*, Donna O'Shea, *Senior Member, IEEE*, Ankit Vidyarthi, and Deepak Gupta, *Senior Member, IEEE*

***Abstract*—5G emerges as the bedrock for the Industrial Internet of Things (IIoT), it facilitates the seamless, low-latency fusion of artificial intelligence and cloud computing, thereby fortifying the entire industrial procedure within a framework of smart and intelligent IIoT ecosystems. Concurrently, the continuously changing landscape of cybersecurity threats in the realm of the Internet of Things (IoT) is giving rise to unparalleled security complexities. These challenges are particularly pronounced in the context of zero-day attacks, and integration of 5G technology further exacerbates the intricacy of the situation. Thus this paper introduces a cutting-edge 5G-enabled framework for cyberthreat detection leveraging Federated Learning (FL) without the need for data sharing. It employs a dual Autoencoder (AE) based model. Distinctly, our model utilizes two synchronized AEs for each client, integral to FL mechanism. While one AE evaluates the IIoT environment based on normal network patterns, another focuses on attack scenarios. For decisive threat assessment, the system uses the capabilities of a one-class SVM classifier with AEs. Furthermore, our method ensures a synergistic blend of self-learning and collaborative learning by implementing a polling mechanism between overarching AE classifier and those tailored to individual client data and counters zero-day threats and out performs traditional AI/ML techniques.**

***Index Terms*—5G, IIoT, cyberthreat, federated learning, autoencoders, zero-day attack.**

I. Introduction

THE RAPID proliferation of the Internet of Things (IoT) and its industrial counterpart, the Industrial Internet of Things (IIoT), signifies a remarkable evolution in the global technological landscape [1]. This evolution involves the interconnection of billions of entities, from everyday consumer electronics to large-scale industrial apparatus, facilitated by the ability to generate, process, and disseminate colossal volumes of data [2].

In essence, IoT refers to the network of physical entities, or "things", integrated with a variety of technologies such as sensors, software, and connectivity modules, which enable these entities to coordinate and share data over the Internet. These entities encompass a broad spectrum of devices from smart household appliances, such as thermostats and refrigerators, to health-oriented devices like wearable fitness trackers [3].

On the other hand, IIoT, an offshoot of IoT, is focused primarily on the industrial utilization of these technologies. The IIoT framework incorporates elements of machine-to-machine communication, automation techniques, Machine Learning (ML), and real-time data analytics. These elements collectively aim to amplify efficiency, productivity, and safety across multiple sectors, including manufacturing, logistics, energy, and transportation [4].

Wireless connections continue to play a crucial role in the growth of IoT and IIoT, ensuring extensive and robust links between devices, machinery, systems, individuals, and entities. 5G is set to drive the evolution of automated manufacturing, particularly in localized and public 5G solutions. This represents a pivotal opportunity to advance future wireless communication [5].

IIoT systems, in particular, are prime targets for cyber-attacks due to the expanded attack surface and each connected device represents a potential weak spot. The data held within these systems is often a treasure trove of intellectual property and sensitive corporate information, making them attractive targets for cybercriminals [6].

Furthermore, IIoT systems frequently suffer from insufficient security protocols, either from oversight or inherent device limitations, rendering them comparatively easy targets [7]. Moreover, the introduction of 5G in the IIoT expands the potential for cyber attacks. The increased speed and connectivity of 5G networks create more entry points for hackers to exploit vulnerabilities in connected devices and systems, posing greater risks to critical infrastructure, data integrity, and operational continuity.

Manuscript received 18 August 2023; revised 19 October 2023; accepted 19 November 2023. Date of publication 28 November 2023; date of current version 26 April 2024. This work was supported by the Science Foundation of Ireland (SFI) under Grant 16/RC/3918, through EU's MSCA under Agreement 847577 and Agreement SFI 12/RC/2289-P2 (Insight). *(Corresponding author: Priyanka Verma.)*

Priyanka Verma, Nitesh Bharot, and John G. Breslin are with the Data Science Institute, University of Galway, Galway, H91 TK33 Ireland (e-mail: priyanka.verma@universityofgalway.ie).

Donna O'Shea is with the Department of Computer Science, Munster Technological University, Cork, T12 P928 Ireland.

Ankit Vidyarthi is with the Department of Computer Science Engineering and Information Technology, Jaypee Institute of Information Technology, Noida 201301, India.

Deepak Gupta is with the CSE Department, Maharaja Agrasen Institute of Technology, New Delhi 110086, India.

Digital Object Identifier 10.1109/TCE.2023.3335385

Security strategies that are adequately mature for conventional Information Technology (IT) systems may not translate seamlessly to the 5G-enabled IIoT context [8].

To circumvent these limitations, a growing body of recent research is pivoting towards ML and Deep Learning (DL) methodologies [9].

However, their practical application is often restricted due to apprehensions about privacy, security risks associated with data transfer between industrial environments and servers, and the time-demanding nature of the training phase on a singular machine [10].

Therefore, the integration of edge computing with DL can help bring intelligence directly to the source of data creation, thus tackling challenges such as data privacy, high communication costs, the need for vast memory space, shortened training periods, and high latency [11]. Local DL and distributed DL techniques have been developed to foster this edge intelligence, and they function without the need for data aggregation [12] thus reducing issues with singular machines.

However, these techniques frequently fail to accurately identify zero-day attack instances, in which cybercriminals commandeer a network of compromised computing devices to take advantage of previously unknown weaknesses in IoT systems. A lack of existing training samples within individual IoT-edge devices hampers the effectiveness of the above methods in such situations. The detection of zero-day threats is inherently difficult, primarily because of the lack of previous information regarding such incidents [13], [14].

One potential remedy to this issue is Federated Learning (FL) [15]. In the FL paradigm, each data proprietor (referred to as a client) constructs a model using their proprietary data and transmits the model weights to a centralized server. The server's function is to amalgamate these parameters to create a comprehensive model, which can then be deployed across all client environments [16].

Thus based on the aforementioned discussion, this paper elucidates an FL-based framework for zero-day cyber-threat detection within 5G-enabled IIoT ecosystems. This advanced cyberthreat detection paradigm involves training two independent AE models on each client's data, one model learning from normal traffic and another learning from attack traffic, with the models' parameters being shared with a server using a 5G network, thereby eliminating the need to share raw data.

These parameters are then used by the server to build two global components to learn normal and attack profiles. Using these global models, each client maps its personal data into a latent space and trains two classifiers for the received global AE models. Further client trains two more one-class classifiers on their local normal and attack data. Then shares the output of four classifiers with their own polling unit (ϑ) for final prediction. Every client has the ability to assess its condition using the shared learning models for representation and the repository of classifiers. Moreover, the system is equipped with 5G capabilities, enabling it to achieve remarkable data throughput and minimal communication delay. This empowers sensors and devices to seamlessly exchange data in real-time between clients and servers, especially when deployed within a data-intensive 5G framework as exemplified in [17]. This advancement enhances system efficiency compared to earlier iterations where immediate connectivity was restricted to private networks with high-speed links. Consequently, the newly devised system is well-suited for real-time applications. Aiming to tackle the problem of zero-day attack detection, the main contributions of the proposed framework are:

- Proposed a dual-model classification system within a federated framework, designed to identify unfamiliar examples by contrasting them with distinct normal and attack profiles.
- This work provides valuable insights and a robust solution to counter zero-day attacks effectively. It proposes a novel and effective framework that achieves high accuracy, detection rate, and F1-score, outperforming traditional models and hybrid approaches.
- The proposed framework provides the ability to effectively handle imbalanced data sets by using two individual models for attack and normal traces.

The rest of the paper is organized as: Section II describes the existing solutions to the zero-day attack. These include centralized and decentralized ML/DL-based techniques. Next, Section III aims to address the proposed approach, explaining the know-how of the proposed approach. It consists of the workflow of the proposed technique along with the algorithms and the diagrammatic representations. The proposed approach is then validated by experimentation and the results obtained are analyzed in Section IV. It aims to compare the results of various existing solutions with the proposed approach. Lastly, the conclusion of the paper along with some scope for future research is mentioned in Section V.

II. Related Work

This section introduces the present techniques available for cyber threat detection. AI-based IDS have been extensively utilized in device-level detection, marking notable successes [18], [19]. The majority of existing research, such as studies [20], [21] on intrusion detection operate under a closed-set assumption, meaning they only anticipate encountering attack classes that were present in the training data set during testing.

In a 2017 study focusing on water treatment systems, Inoue et al. [22] introduced an anomaly detection model utilizing Deep Neural Networks (DNN). By employing Long Short-Term Memory (LSTM) neural networks within their investigation, they were able to reveal that the DNN model, having been trained on normal data, exhibited performance that surpassed that of the one-class Support Vector Machine (SVM) model. The training process for the one-class SVM model was more rapid compared to the DNN method they proposed.

In 2020, there were numerous studies conducted on cyber-threat detection. Audibert et al. [23] put forth an anomaly detection approach in an unsupervised way and leveraging AEs for multivariate time series. Their method exhibited rapid training time, robustness to parameter selection, and stability. Their result evaluation shows that their method stands up well against other methods in the field.

Abdelaty et al. [24] introduced a modular deep learning-oriented anomaly detection model for IIoT systems. They evaluated their proposed model on two IIoT datasets and found it to have superior performance, especially regarding the F1-score metric, compared to several existing methods. Whereas, Moon et al. [25] proposed a combined use of one-class SVM and LSTM networks for anomaly detection within IIoT systems. Their evaluation revealed that the LSTM-based technique was more effective than the one that utilized one-class SVM.

Moreover, Nagarajan et al. [26] offered an anomaly detection method aimed at maintaining privacy within IIoT networks. They compared their approach to two datasets with traditional ML techniques. The results demonstrated that their method had a higher detection rate than the others.

However, these closed-set AI-based Intrusion Detection Systems (IDS) come with inherent limitations. These systems often fall short when faced with unknown or novel attack vectors, and they tend to generate high false positive rates, potentially leading to alert fatigue. Moreover, maintaining closed-set IDS involves labor(intensive), and frequent updates to incorporate new threat intelligence, making it challenging to keep up with the rapidly evolving threat landscape. These systems struggle to adapt to changes in network configurations and are ill-suited to handling class imbalances or scaling effectively in dynamic network environments. Attackers can exploit the weaknesses of closed-set IDS through evasion techniques, emphasizing the need for more adaptive and proactive security measures.

To address these limitations, the cybersecurity community has recognized the importance of open-set intrusion detection methods. Open-set IDS distinguishes between known and unknown threats, primarily relying on anomaly detection and more advanced machine learning techniques. These methods offer a forward-looking approach by continuously learning from new data and adapting to evolving threats, reducing false positives, and offering better scalability and adaptability. They are designed to be more resilient against evasion techniques and can provide a more robust defense against an ever-changing threat landscape, making open-set intrusion detection an essential component of modern cybersecurity strategies.

Only a handful of studies have explored open-set intrusion detection. For instance, Ibrahim Hairab et al. [27] suggested a method based on CNN for anomaly detection in IoT networks to counteract zero-day attacks. Despite this, their proposed method falls short of providing a detailed classification of known attacks.

Ping and Ye [28] proposed open-set IDS, that addresses the problem of seen and unseen behaviors/traffic through three modules named MinMax autoencoder, the classifier, and pseudo extreme value machine. They conducted experiments on USTC-TFC2016 & CSE_IDS2018+ datasets to establish the efficacy of their proposed approach achieving accuracy of 72% and 89.4% respectively.

Farrukh et al. [29] present a novel framework specifically designed to address the open set recognition challenge within the domain of Network Intrusion Detection Systems, with a particular focus on IoT environments. The proposed framework leverages image-based representations of packet-level data, extracting both spatial and temporal patterns from the network traffic. Furthermore, we incorporate stacking and sub-clustering techniques, which facilitate the identification of previously unknown attacks by effectively capturing the intricate and varied characteristics of legitimate network behavior.

Wu et al. [30], in their study, devised an intrusion detection method based on dynamic ensemble incremental learning. While this approach is capable of adapting to newly discovered local attack variants, it struggles to incorporate knowledge of new attacks that manifest in other IDSs. Given that IDS devices dispersed across various geographical locations might face different attack variants, collaborative model learning can substantially enhance the defense capabilities of smart community systems against unfamiliar attacks.

While the aforementioned methods yield exceptional results, there's a significant hurdle that precludes their widespread adoption in the industrial sector. They are centralized techniques in nature, requiring whole data to be housed on one system for training purposes. It makes the training process both time-consuming and hardware-intensive. Additionally, the need to transfer and store all data samples from industrial operations on one server raises concerns about security and privacy. Various industrialists become hesitant to share their data with other entities to train ML models. In response to these challenges, several studies have devised the use of non-centralized techniques, such as FL, to train models. These methods circumvent the need for data sharing, addressing many of the concerns associated with centralized systems.

Detection methodologies based on FL, as referenced in [10], allow for the sharing of locally learnt parameters instead of actual data. This approach is proved superior in accelerating training, protecting privacy, and reducing latency. Popoola et al. [31] employed this strategy by federating IoT edge devices along with DNN to detect zero-day botnet attacks. Reference [32] introduced an innovative system combining blockchain technology with federated intrusion detection to handle untrustworthy updates. Meanwhile, Ruzafa-Alcázar et al. [33] devised an intrusion detection method leveraging semi-supervised federated methodology. In this arrangement, unlabelled samples were utilized for boosting performance of classification system.

In 2022, [34] introduced the FL technique designed to detect attacks on solar farms. Tests in diverse scenarios and comparisons with traditional ML strategies were noted. The experiment demonstrated that their proposed FL-based model's performance closely mirrored its centralized counterpart but with the advantage of reduced computational and data transfer costs.

Rey et al. [35] suggested both supervised and unsupervised FL-based methods for detecting malware, which they evaluated under various conditions. They compared this model with two other methods, revealing that the FL-based approach was superior to employing multiple local models, one per client.

Even though the above FL-based methods tackle the privacy concerns related to centralized ML methods, they fall somewhere short in detecting unknown attacks on IIoT. These

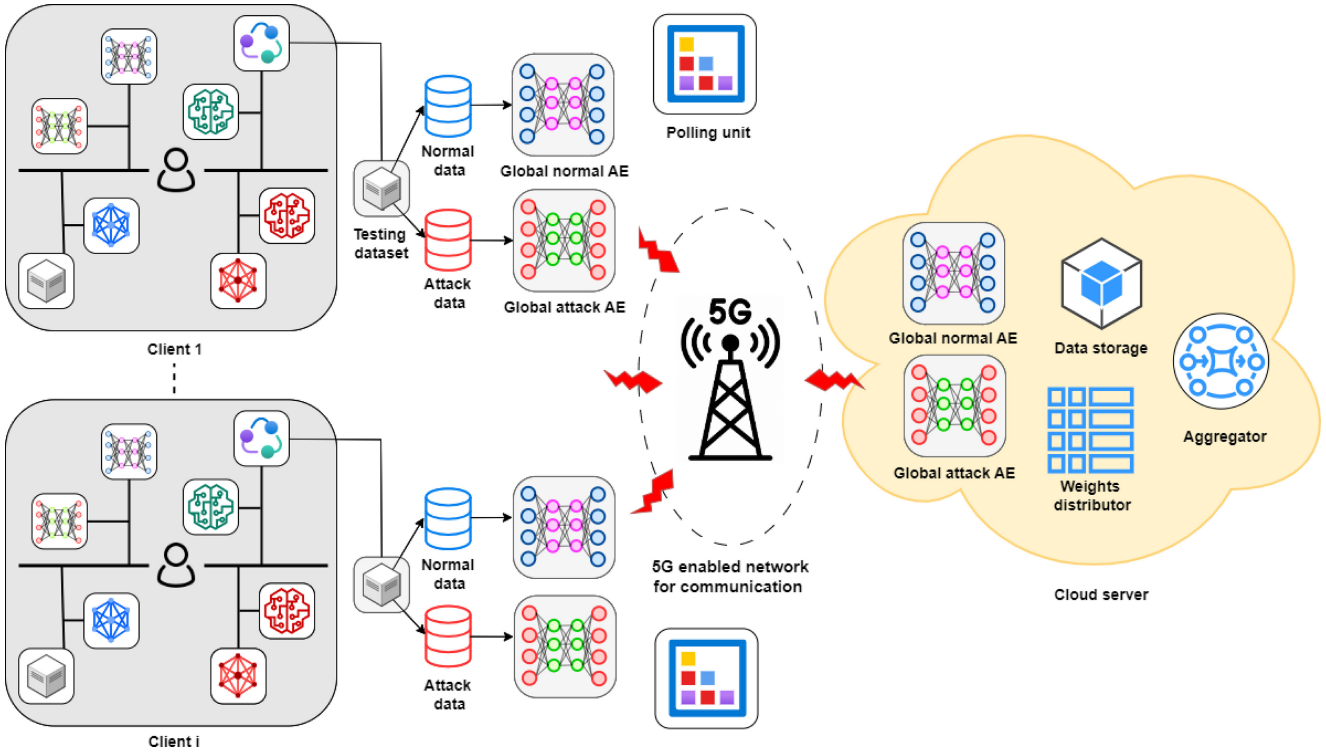

Fig. 1. The proposed zero-day guardian framework with 5G network.

methods generally focus on the creation of general classifier models and thus perform low when encountering zero-day attacks. Moreover, the above methods do not give preference to self-learning and completely rely on the global model, which allows a single attack client to influence the overall global model. Hence, above mentioned strategies do not handle the zero-day scenarios well, thus creating a need for an efficient model.

III. Proposed Zero-Day Guardian Framework

The conventional approach to ML, where the model is trained and tested on the same set of data, restricts individual client's growth for detecting new attack traces. Collaborative learning, on the other hand, offers a better way to enhance individual client progress. However, collaborative learning often involves sharing data, which poses security risks. To address this, FL emerged as a decentralized system that not only ensures client security but also eliminates the need for data sharing among clients. Additionally, FL enables individual clients to participate in a global scenario, promoting better learning outcomes.

FL is a privacy-preserving decentralized method, developed by Google, that reduces the client-side computation and parallelly allows each client to gain global knowledge without globally sharing data with each other or without bringing data to a central location. Let there be C clients $c \in C$, and the number of communication rounds is R and client training data as $client\_t_1, client\_t_2, client\_t_3, \ldots, client\_t_c, \ldots, client\_t_C$. Each $client\_t_c = \{client\_t\_X_c, client\_t\_y_c\}_{c=1}^{C}$ and testing dataset as $client\_tt\_X$, $client\_tt\_y$ for all clients. Each training data has a distinct distribution such that $P(client\_t_a) \neq P(client\_t_b)$. Every client has its individual local model say M and it is trained with loss function as:

$$L(client\_t_c, w) = \frac{1}{|client\_t_c|} \sum L_c(client\_t\_X_c, client\_t\_y_c, w) \quad (1)$$

where $\sum$ varies for $(client\_t\_X_c, client\_t\_y_c) \in client\_t_c$ and $L_c(client\_t\_X_c, client\_t\_y_c, w)$ is a specific function to be minimized. So our goal is to finally aggregate the M_l to obtain global model M_g for each client by maintaining the data privacy:

$$min_{\{M_c\}_{c=1}^{C}} \frac{1}{C} \sum_{c=1}^{C} \frac{1}{|client\_t_k|} * \sum_{i=1}^{client\_t_c} L(M_l^{(c)}(client\_t\_X_c), client\_t\_y_c) \quad (2)$$

In both conventional ML assessment techniques and federated approaches, the model is conditioned and evaluated using identical categories of data. During the training phase, the model assimilates the underlying patterns from each category of data. Subsequently, these learned patterns are employed to recognize samples from the corresponding classes in the testing phase. However, these approaches assume that the training dataset includes all the attack classes that the model will encounter after deployment, which limits the system's ability to detect attacks outside its dataset. This lack of robustness raises concerns about the system's security, as it may allow attack traffic to bypass its defenses. So, we propose a dual AE model-enabled FL framework for handling zero-day attacks in 5G-enabled IIoT systems. The proposed

Algorithm 1 Zero-Day Guardian Framework Workflow

Prerequisites: Clients k with local data D, 2 local classifiers $\mho_b$ & $\mho_m$, 2 global classifiers $\eth_b$ & $\eth_m$, a data differentiator $\beth$ and 2 AEs ϱ_{l_b} & ϱ_{l_m} for normal and attack data respectively, and a voting unit ϑ.
A cloud server with 2 global AEs ϱ_{g_b} & ϱ_{g_m}, a weight distributor ω, a data storage, an aggregator.
Working:

Use $\beth$ to divide the D into the normal and attack dataset D_n & D_a respectively.
$(D_b, D_m) = \beth$ (D)
Use D_b & D_m to participate in federated scenario to obtain ϱ_{g_b} & ϱ_{g_m}
$\varrho_{g_b} = AEFedarated_b(D_b, \varrho_{l_b})$
$\varrho_{g_m} = AEFedarated_m(D_m, \varrho_{l_m})$
Train the $\mho_b$ & $\mho_m$
$\mho_b = \mho_b$.fit(D_b)
$\mho_m = \mho_m$.fit(D_m)
Train $\eth_b$ & $\eth_m$
$\eth_b = \eth_b$.fit(ϱ_{g_b}.predict(D_b))
$\eth_m = \eth_m$.fit(ϱ_{g_m}.predict(D_m))
Testing Phase
Replicate the test data Td to generate 4 test spaces
Td_1 = Td.copy()
Td_2 = Td.copy()
Td_3 = Td.copy()
Td_4 = Td.copy()
Pass Td_b & Td_m through ϱ_{g_b} & ϱ_{g_m} obtained by federated process.
$Td'_1 = \varrho_{g_b}$.predict(Td_1)
$Td'_3 = \varrho_{g_m}$.predict(Td_3)
Use classifiers to predict the probabilities
$y\_pred_g = \eth_p$.score_samples(Td'_1)
$y\_pred'_g = \frac{e^{y\_pred_g}}{\sum_{j=1}^{\ell} e^{y\_pred_{g_j}}}$
$y\_pred_l = \mho_l$.score_samples(Td_2)
$y\_pred'_l = \frac{e^{y\_pred_l}}{\sum_{j=1}^{\ell} e^{y\_pred_{l_j}}}$
where, j = {1,2,3,4}, g = {1,3}, l= {2,4}
ℓ = len($y\_pred_{g\ or\ l}$)
Combine the results of the local and global classifiers individually
y_pred_self_model = list()
for i in range(len(y$_pred_{l_b}$)):
if y$_pred_{l_b}$ [i] ≥ y$_pred_{l_m}$ [i]:
y_pred_self_model.append (0)
else:
y_pred_self_model.append(1)
y_pred_global_model = list()
for i in range(len(y$_pred_{g_b}$)):
if y$_pred_{g_b}$[i] ≥y$_pred_{g_m}$ [i]:
y_pred_global_model.append (0)
else:
y_pred_global_model.append(1)
Predict the outcomes
y_pred = list()
for i in range(len(y_pred_self_model)):
if y_pred_self_model[i] == y_pred_global_model[i]:
y_pred.append(y_pred_global_model[i])
else:
y_pred.append(y_pred_self_model[i])
y_pred is desired output

framework aims to tackle the zero-day attack by separately training classifiers to identify normal and attack traffic. This in turn helps to train the classifiers more precisely to handle only one kind of data empowering it with a higher detection rate. The usage of separate models for normal and attack data also handles the issues caused by the dominant class if the dataset is imbalanced.

The scenario considered within the proposed framework involves clients functioning as edge nodes, symbolizing individual industrial units. Each unit integrates a variety of intelligent sensors, actuators, cameras, robots, machines, IC controllers, and IoT-based chips to gather vital data. This data is then stored in a database to facilitate the training of a local model. Subsequently, these clients collaborate by sharing gradients from their respective local models, which are aggregated on a cloud server. This FL process is facilitated through the utilization of the Internet and the efficiency of 5G infrastructure, with its ultra-low latency, high bandwidth, network slicing capabilities, and advanced security features, 5G facilitates real-time control, machine-to-machine communication, and seamless connectivity for an array of devices and sensors, fostering the growth of interconnected, intelligent systems. It enables real-time remote monitoring and maintenance, enhances mobility for robots and autonomous vehicles, and ensures scalability and energy efficiency in manufacturing environments. Furthermore, 5G's potential to offer global, high-speed connectivity promises to reshape the way industries operate, making them more efficient, responsive, and globally connected.

The major components of the proposed framework are; at the client end it consists of two global AEs and their associated two classifiers, data storage, two more classifiers built on local normal and attack data, a data distinguisher, and a polling mechanism. At the server, we have an aggregator, weights distributor, data storage, and two global AEs.

Algorithm 1 describes the workflow of the proposed framework with Table I describing the parameters involved. The major steps of the proposed framework are:

- At each client it initially begins by initializing the data distinguisher process where the normal traffic data is separated from the attack one.
- These separated datasets D_b for normal & D_m for attack are used to separately train the local AE ϱ_{l_b} for normal

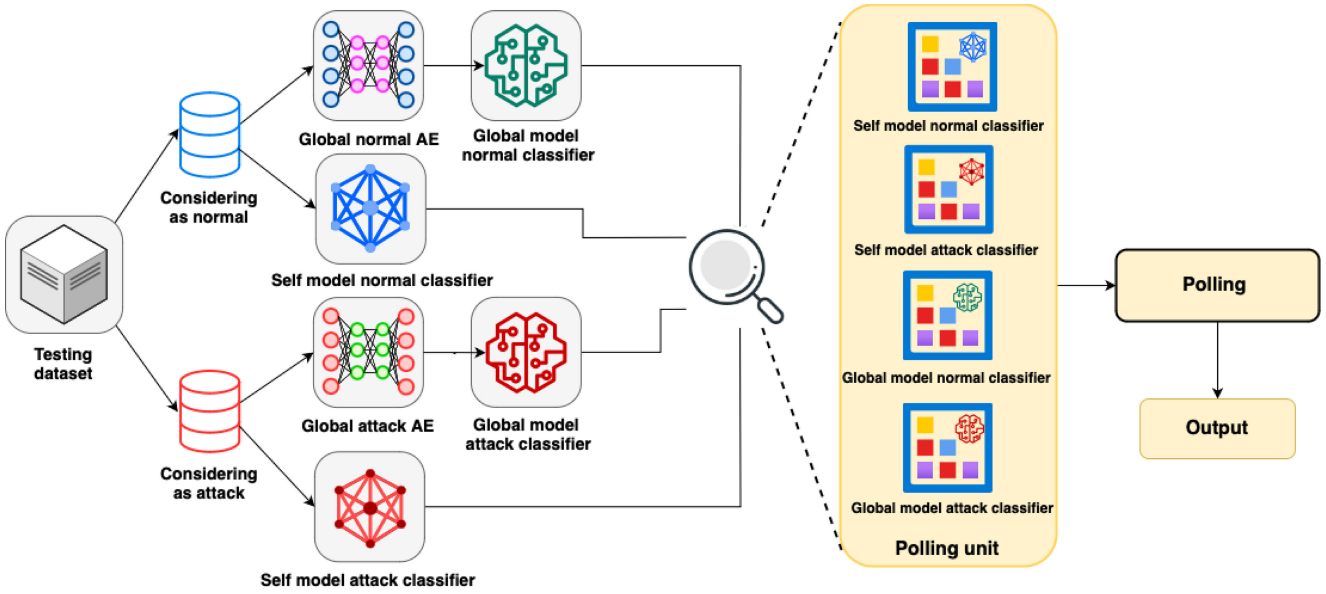

Fig. 2. Testing with the proposed framework at individual client.

TABLE I
DESCRIPTION OF SYMBOLS USED IN PROPOSED APPROACH

Description	Parameter
normal dataset	D_b
attack dataset	D_m
Local AEs	$\varrho_{l_b}, \varrho_{l_m}$
Global AEs	$\varrho_{g_b}, \varrho_{g_m}$
Global one class classifiers	$\eth_b, \eth_m$
Local one class classifiers	$\mho_b, \mho_m$
Voting unit	ϑ
Data differentiator	$\beth$
Encoder function	f
Loss function	ζ

traffic and AE ϱ_{l_m} for attack traffic and shared with the server.

- At the server, the process of aggregation takes place to get the global AEs ϱ_{g_b} and ϱ_{g_m} for normal and attack traffic respectively, and shared back to clients.
- Further on global AEs ϱ_{g_b} and ϱ_{g_m}, each client trains two separate global one class classifiers such as $\eth_b$ & $\eth_m$.
- Then on local data D_b & D_m the clients local classifiers $\mho_b$ & $\mho_m$, named as self-model classifiers are trained.
- In the case of testing, the testing data is first passed through the ϱ_{g_b} & ϱ_{g_m} respectively and then classified using the global classifiers $\eth_b$ & $\eth_m$ respectively.
- The testing data is also passed through the self-model classifiers to generate two more outcomes from $\mho_b$ & $\mho_m$ respectively.
- As a final prediction step, these predictions are combined to generate the desired classification within the polling mechanism.

As mentioned, each client has two local AE ϱ_{l_b} for normal traffic and AE ϱ_{l_m} for attack traffic. This AE is a specially designed network that has the power to transform data through the use of neural structures. It takes the input data, says D with fs features, processes them, and converts them to another output set with the same fs-number of features. It entails the usage of an encoder and decoder which works collaboratively to first reduce the feature set to a specified feature set say fs' (through encoder) and then reconstruct the feature set fs through its decoder set. AE employed in our system mainly uses the same functionality and then uses Mean Squared Error (MSE) as its loss function.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{true} - y_{pred})^2$$

where n is the total number of predictions

Once the local training of ϱ_l is completed, every client shares its AE weights with the global server for the federated process. Once the parameters of the client's local components are shared with the server, it employs the federated averaging method, as described in [15], to amalgamate the components into two global elements, as represented by Eq. (3) and Eq. (4). The resultant global components are depicted in Eq. (5). These consolidated global components are subsequently distributed to the clients, allowing them to refine the components using their individual local data. After this refinement, the parameters are shared back with the server using a 5G network between the client and server. This iterative model weights the training cycle, conveying them to the server, and their subsequent aggregation constitutes the FL process.

$$\mathbb{W}_i = \frac{1}{K}\sum_{k=1}^{K} w_i^k \tag{3}$$

where $\mathbb{W}_i$ is the aggregated weights of components i (normal or attack) and w_i^k are the weights of components i for client k.

$$B_i = \frac{1}{K}\sum_{k=1}^{K} b_i^k \tag{4}$$

Algorithm 2 AEFederated

Input: K clients with their local data & a server

```
for comms_round do
  for client k do
    Train the normal AE ϱ1 present with client
      ζ(D_n^k, r_n^k(f_n^k(D_n^k)));
    Train the attack AE ϱ3 present with client
      ζ(D_a^k, r_a^k(f_a^k(D_a^k)));
    Share the AE parameters ϱ_{g_i} with server for
    both AE's
    ω_i = {ω_i^k | i ∈ {normal, attack}}
    β_i = {β_i^k | i ∈ {normal, attack}}
  end
  Server aggregate the parameters & make the
  global components ϱ_{g_i}
    W_i = (1/K) Σ_{i=1}^{K} w_i^k
    B_i = (1/K) Σ_{i=1}^{K} b_i^k
    η_i = ϱ_{g_i}(D) = σ(w_i D + b_i);
  Server shared global components with clients
end
```

where B_i is the accumulated biases of normal or attack counterparts of i and b_i^k are the biases of component i for client k.

$$\eta_i = \varrho_{g_i}(X) = \sigma(w_i X + B_i) \tag{5}$$

In this context, ϱ_{g_i} signifies the global AE function for component i, η_i compares the new samples with the normal and attack data.

Once the global models are shared with clients, they train four one-class SVM classifiers, two using the new representations η_b and η_m from the global AEs and two on local clients' data for normal and attack traffic. One-class SVM is a type of SVM algorithm used specifically for anomaly detection rather than classic classification tasks. This technique is particularly useful when the data consists mainly of one class (the "normal" class) and the goal is to detect outliers or anomalies, which form a second class that is typically underrepresented in the dataset. As in our proposed approach, we are separately training the classifiers of normal and attack profiles, thus one-class SVM fulfills this purpose.

One-class SVM operates by defining a decision boundary around the normal/attack data in a way that positions this data in a small region while outliers fall outside this region. It achieves this by learning a decision function that is positive for the region of normal/attack data and negative for the region where outliers lie.

However, it's important to note that one-class SVM's performance can be sensitive to the choice of the kernel and the kernel's parameters, as well as the value of the hyperparameter that controls the trade-off between maximizing the distance of the hyperplane from the origin and minimizing the number of instances that fall on the side of the hyperplane with the outliers. Equation (6) shows the training objective of the one-class SVM model. The goal of one-class SVM is to obtain a hypersphere with the center of c and radius of Υ by minimizing the Υ^2.

TABLE II
PARAMETERS USED IN THE ZERO-DAY GUARDIAN MODEL

Model	Parameters
FL Model	Learning rate (0.01), Momentum (0.9), Decay (0.001), Loss function (mean-squared error), Epoch (20), Number of clients (K=10), Validation split= 0.2), Metrics (mean-squared error)
AE Model (normal or attack)	2 Encoder layers (1 input layer with 59 neurons [shape of training data] + 1 relu activated dense layer with 128 no. of neurons), 1 Bottleneck layer (with sigmoid activation and 62 neurons), and 2 Decoder layers (1 mirror layer with 128 neurons and relu activation + 1 output layer with 59 neurons and linear activation) thereby giving total trainable parameters as 31353

$$min_{\Upsilon_i^k, c_i^k} \quad \Upsilon_i^{k^2} + C\sum_{j=1}^{n}\mu_j \tag{6}$$

subjected to:

$$||\varrho_{g_i}(X_i^k) - c_i^k||^2 \leq \Upsilon_i^{k^2} + \mu_j$$
$$\mu_j \geq 0$$

where Υ_i^k and c_i^k are the parameters of the one-class SVM $\mho_i^k$ which is trained for component i of client k, μ_j are slack variables, C is the penalty parameters $\varrho_{g_i}(\cdot)$ is the global AE (Eq. (5)) for component i and X_i^k are the local samples of client k belongs to component i.

IV. RESULTS EVALUATION

In this section, we outline the experimental setup, dataset description, zero-day scenario simulation, and subsequent comparative result analysis. Our approach revolves around a dual AE model-enabled 2-way FL framework designed to counter zero-day attacks. The chosen dataset X-IIoTID represents real cybersecurity incidents to ensure practicality. We meticulously simulate zero-day scenarios to evaluate the efficacy of our zero-day guardian framework. Result analysis encompasses performance metrics, model accuracy, and comparison against traditional methods. This comprehensive evaluation demonstrates the efficacy of our approach in detecting and mitigating zero-day threats, laying the foundation for more robust and proactive cybersecurity measures.

A. Experimental Setup and Parameters

The proposed mechanism was developed and analyzed using Python 3.10,[1] on a MacBook Pro equipped with an Apple M1 Pro processor. The MacBook Pro configuration includes a 10-core CPU, a 16-core GPU, 16 GB of RAM, and a 1TB SSD. Table II gives the parametric description of the proposed framework with Table III describing the performance metrics used for the evaluation.

B. Dataset Description

To evaluate the proposed framework we utilized X-IIoTID dataset [36]. This dataset is specifically designed for IIoT

[1] https://docs.python.org/3/library/

TABLE III
PERFORMANCE METRICS

Accuracy	(TP + TN)/(TP+TN+FP+FN)
Detection Rate (Recall)	TP/(TP+FN)
F1-Score	(2*Precision*Recall)/(Precision+Recall)

applications and captures system activity generated by a range of IIoT devices. With a total of 820,834 traces, the dataset comprises 68 features and encompasses 0-9 different labels. These labels represent various attack types, along with a separate label for normal requests. The whole training dataset was divided among 10 clients for the FL approach. This divided dataset for each client was further bifurcated into normal and attack traffic using the data distinguisher and used to train the individual AEs in a federated way respectively.

C. Building a Zero-Day Scenario

A zero-day attack is characterized by its novelty and the fact that it represents an unfamiliar type of threat that has not been previously encountered or described to the model during its training phase. In order to effectively simulate zero-day attacks in our experiments, a commonly adopted practice involves partitioning the dataset into two distinct groups: one consisting of known network traffic and another comprised of 'unknown' network traffic. In the context of our specific experiments, we implemented this approach by filtering out instances associated with attack labels named 'Lateral Movement' (comprising 31,596 traces) and 'Weapon' (comprising 67,260 traces) from the training dataset. This selective exclusion of attack labels aimed to create a scenario in which the model encounters attacks it has never been exposed to during training. However, it is important to note that these excluded attack labels, namely 'Lateral Movement' and 'Weapon', were reintroduced in the testing dataset, thus ensuring a comprehensive evaluation of the model's performance in the presence of these zero-day attack types.

D. Result Analysis

Here we analyze and compare the performance of the various AEs with the proposed approach, comparison with other federated attack detection DL models, and comparison with other one-class classifiers, traditional ML classifiers, and DL classifiers for individual clients.

1) Comparison With Other AEs: Firstly, we compared the performance of various AE models. Results obtained from the comparison of different AEs on the X-IIOTID dataset with the proposed approach are described in Table IV. The table provides insights into the communication rounds required for training the AEs and the corresponding AE loss for both attack and normal traffic.

For the multilayered AE, it can be observed that for the number of communication rounds from 2 to 10, the AE loss decreases gradually for both attack and normal data. The lowest AE loss values achieved for attack and normal data are 0.000370 and 0.000305, respectively, at 10 communication rounds. Similar trends can be observed in the case of the single-layered AE, sparse AE, and variational AE, where increasing the number of communication rounds results in a decrease in AE loss. However, the AE loss values are higher for all the other AEs in comparison to the multilayered AE.

TABLE IV
COMPARISON OF DIFFERENT AES ON X-IIOTID DATASET

Type	Comm.Rounds	AE Loss	
		attack data	normal data
Multilayered AE	2	0.000463	0.000401
	4	0.000421	0.000353
	6	0.000398	0.000330
	8	0.000382	0.000315
	10	**0.000370**	**0.000305**
Singlelayered AE	2	0.002962	0.003452
	4	0.002465	0.002992
	6	0.002237	0.002774
	8	0.002094	0.002638
	10	0.001993	0.002541
Sparse AE	2	0.001134	0.001195
	4	0.000949	0.000985
	6	0.000866	0.000890
	8	0.000825	0.000831
	10	0.000780	0.000789
Variational AE	2	0.002384	0.002950
	4	0.001470	0.001628
	6	0.001176	0.001249
	8	0.001022	0.001060
	10	0.000925	0.000944

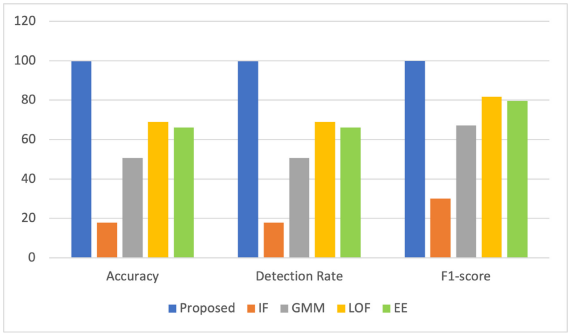

Fig. 3. Comparison of one-class SVM in proposed framework with other one-class classifiers.

Thus it can be concluded from the results that the Multilayered AE demonstrates the lowest AE loss values among all the AEs considered in this comparison for both attack and normal traffic. Therefore, we used multilayered AEs in the proposed approach.

2) Comparison of One-Class SVM Used in Proposed Framework With Other One Class Classifiers Models: This section compares the performance of the opted classifier (one-class SVM) for the proposed approach with other one-class classifiers. The other one-class classifiers considered are; Isolation Forest (IF), Gaussian Mixture Model (GMM), Local Outlier Factor (LOF), and Elliptic Envelope (EE).

The results of the classification performance for different classifiers are presented in the given Figure 3 depicting the accuracy, detection rate, and F1-score for each classifier.

The proposed classifier achieves an accuracy of 99.328%, a detection rate of 99.668%, and an F1-score of 99.844%

TABLE V
COMPARISON MODELS PARAMETERS

MLP	Number of dense layers = 2, number of neurons = (128,128), activation function = (Relu, Relu), optimizer=adam, loss = binary_crossentropy
GRU	Number of GRU layers = 2, number of neurons = (128,108), activation function = (tanh,tanh), optimizer=adam, loss = binary_crossentropy
CNN	Number of CNN layers = 2, number of filters = (128,128), kernel_size = 3, 1 flatten layer, activation function = (Relu, Relu), optimizer=adam, loss = binary_crossentropy
CNN + GRU	Number of CNN layers = 2, number of filters = (128,128), kernel_size = 3, 1 flatten layer, activation functions of CNN layers = (Relu, Relu), number of GRU layers = 2, number of neurons in GRU layers = (128,108), activation function of GRU layers = (tanh,tanh) optimizer=adam, loss = binary_crossentropy

indicating that the one-class SVM used in the proposed framework performs exceptionally well in accurately classifying the data, detecting known and unknown (zero-day) traffic, with high F1-score.

The IF classifier performs poorly, while the GMM, LOF, and EE classifiers achieve varying degrees of effectiveness in identifying zero-day attacks in the dataset. However, the one-class SVM classifier exhibits the highest accuracy, detection rate, and F1-score, indicating its superior performance in zero-day attack detection compared to the other classifiers. Hence, the one-class SVM is chosen for the proposed approach.

3) Comparison of the Proposed Framework With DL-Based FL Models: Table VI presents the performance comparison of different DL models, including MLP, CNN, GRU, CNN + GRU hybrid model, and the proposed model with Table V describing their respective parameters. The results demonstrate that CNN, while effective in capturing spatial patterns, falls short when dealing with sequential information, as evident through the superior performance of the CNN + GRU hybrid system. On the other hand, the GRU model's limitations in handling spatial patterns are compensated by the CNN component of the hybrid model. Zero-day attacks are typically designed to exploit vulnerabilities that are not known to security experts or database systems. Since these models learn from historical data and have not encountered zero-day attack patterns, they lack the ability to detect zero-day attacks effectively. This concept of learning provides them the ability to easily handle and learn the known attack patterns but falls short in case of unknown attacks. Moreover, the proposed approach aims to understand the attack and normal patterns separately therefore they are rendered more power to understand the difference between normal and attack patterns. Table VI demonstrates that the proposed approach significantly outperforms all individual models, including the hybrid CNN + GRU model, across all evaluation metrics. It attains an outstanding accuracy of 99.32%, a detection rate of 99.69%, and an F1-score of 99.84%. These exceptional results

TABLE VI
COMPARISON OF PROPOSED WITH DL-BASED FL MODELS

Type	Accuracy	Detection Rate	F1-score
MLP	75.557	64.371	76.029
CNN	79.469	70.873	80.610
GRU	42.439	38.060	44.308
CNN + GRU	85.761	77.737	86.799
Proposed	**99.328**	**99.688**	**99.844**

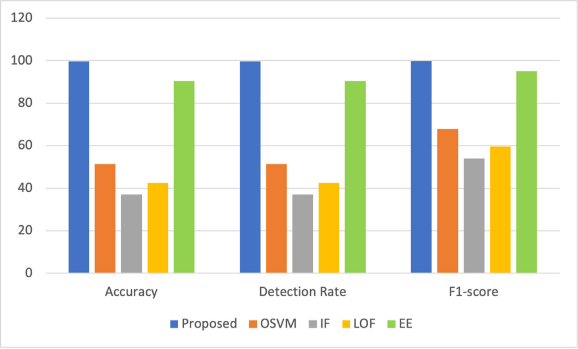

Fig. 4. Comparison of the proposed framework with other one-class classifiers.

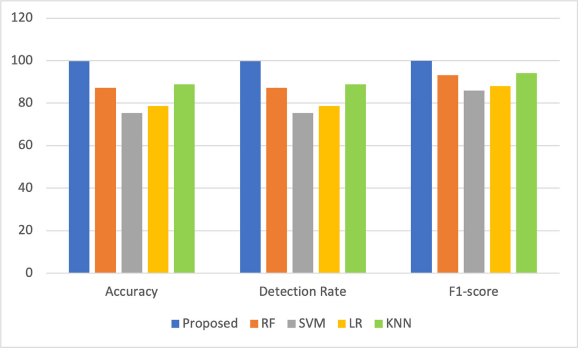

Fig. 5. Comparison of the proposed framework with traditional ML classifiers.

suggest that the proposed approach has successfully addressed the challenge of zero-day attack detection and demonstrates remarkable capabilities in accurately classifying it.

4) Comparison of the Proposed Approach With One Class Classifiers in Centralized Settings: Here the proposed approach is compared with one-class classifiers such as SVM, IF, LOF, EE when applied to the centralized settings where all data is located in one place. As shown in Figure 4 The proposed framework demonstrates the highest accuracy, detection rate, and F1-score among all evaluated classifiers, indicating its superior performance in detecting known and unknown attacks. Moreover, all the other one-class classifier, evaluated on centralized data lags behind the proposed model. However, in centralized settings data privacy is always a concern which is also addressed in the proposed approach while utilizing the FL-based framework.

5) Comparison With ML Models at Individual Client: This section analyzes the effectiveness of the proposed approach in comparison to traditional ML-based techniques for zero-day attack detection. Figure 5 represents the significant difference between the proposed framework and the other ML models

TABLE VII
SERVER SIDE SCALABILITY ANALYSIS IN TERMS OF MEMORY CONSUMPTION AND TIME

Type	Number of Client	Average Memory consumption per client in FL process (in MB)	Average Memory consumption in overall FL process (in MB)	Overall time take in FL process (in sec)	Time taken per epoch per client in FL process (in sec)
normal	2	1814.565	3622.273	163.579	7
	5	1783.715	8918.577	176.256	3
	10	1776.471	17764.712	186.401	3
	15	937.809	14067.139	203.705	2
	20	1206.069	24121.373	250.181	1
attack	2	1858.528	3717.0586	178.5503	9
	5	1936.666	9683.329	166.6853	3
	10	1809.352	18093.515	198.4784	2
	15	1853.859	27807.838	236.979	1
	20	1876.939	37538.783	249.272	1

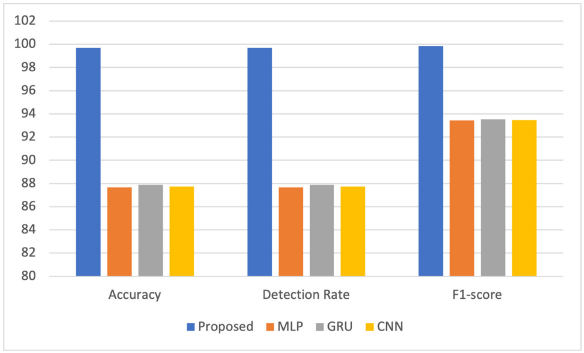

Fig. 6. Comparison of the proposed model with DL classifiers.

in terms of accuracy, showing that the proposed framework outperforms other traditional models with the given dataset. Similar to accuracy, the proposed approach demonstrates the highest detection rate. The high detection rate of the proposed approach indicates its ability to effectively identify zero-day attack instances, making it a promising choice for the application at hand in comparison to traditional ML solutions.

6) Comparison With DL Models at Individual Client: Comparison the performance of the proposed approach with other DL models at individual client level as shown in Figure 6. The proposed model achieves exceptional performance with an accuracy of 99.32%, detection rate of 99.69%, and F1-score of 99.84%, showcasing its superior ability to accurately classify instances and detect anomalies. The MLP, GRU, and CNN models also perform well, achieving accuracies of 87.66%, 87.87%, and 87.72%, respectively. In conclusion, the proposed model outperforms all other models, exhibiting the highest performance across all metrics, while the MLP, GRU, and CNN models also achieve high performance in the classification task.

7) Scalability Analysis: This section outlays the concept of scalability with our proposed framework. It displays various results of the proposed framework, analyzing it with multiple clients. Moreover, it provides insights into the time taken and memory consumption with varying numbers of clients.

Table VII shows results for five different scenarios, by varying the number of clients as 2, 5, 10, 15, and 20 to show the effect of scalability on the proposed approach. The metrics include average memory consumption per client ranging from 937.809 MB to 1814.565 MB for normal scenarios and 1089.352 MB to 1936.666 MB for attack scenarios. Overall memory consumption for the FL process ranges from 3622.273 MB to 24121.373 MB for the normal scenario and 9683.329 MB to 37538.783 MB for the attack scenario. Moreover, the overall time taken for the complete FL process ranges from 163.579 seconds to 250.181 seconds for the normal scenario and 166.6853 seconds to 249.272 seconds for the attack scenario. The time taken per epoch per client ranges from 1 to 7 seconds for the normal scenario and 1 to 9 for the attack scenario. These numbers provide a comprehensive overview of how server-side performance metrics change with varying client numbers in the FL process for different scenarios. Thus, from the above results it is observed that even with an increased number of clients (more connected IoT devices), the proposed approach is able to deal with them without adding much complexity and resource consumption to the system.

V. CONCLUSION

In conclusion, our research introduces an innovative approach to enhance cybersecurity defenses against zero-day attacks and address data imbalance within the context of a 5G network. The proposed framework, which leverages a dual Autoencoder (AE) model-enabled Federated Learning (FL) system, has yielded remarkable results. It achieved an exceptional accuracy rate of 99.32%, a detection rate of 99.69%, and an F1-score of 99.84%. These results clearly surpass the performance of traditional models and hybrid architectures, underscoring the framework's effectiveness in accurately identifying and classifying zero-day attacks. Moreover, the incorporation of separate AEs during training significantly improved the handling of data imbalance, particularly benefiting underrepresented classes.

Furthermore, our adoption of the dual model FL framework facilitated efficient collaboration and knowledge sharing among distributed nodes, leading to enhanced model generalization and scalability. These outcomes collectively establish our approach as a robust and promising solution to bolster cybersecurity defenses in the face of dynamic and evolving threats in real-world scenarios. Nevertheless, it is important to acknowledge that this approach does introduce increased complexity and computation costs. In our forthcoming research efforts, we intend to focus on optimizing the FL process

and explore its applicability in various domains, continuing to push the boundaries of advanced threat detection and data handling techniques. Concurrently, we will explore the practical viability and implementation of integrating edge computing with deep learning to harness edge intelligence within intrusion detection systems, addressing the need for real-world applicability and optimization.

ACKNOWLEDGMENT

The research presented emanated from the European Union's Horizon 2020 program for research and innovation, which provided funding under Grant Number 847577 (SMART 4.0 Marie Skłodowska–Curie Actions COFUND), and also by a grant from Science Foundation Ireland under Grant Number SFI/12/RC/3918 and SFI 12/RC/2289-P2 (Insight). In order to promote open access, the author has chosen to apply a CC BY public copyright license to any version of the Author Accepted Manuscript that results from this submission.

REFERENCES

[1] C. F. Strnadl, "End-to-end architectures for data monetization in the Industrial Internet of Things (IIoT) concepts and implementations," in *Monetization of Technical Data: Innovations from Industry and Research*. Berlin, Germany: Springer, 2023, pp. 149–183.

[2] S. Muthunagai and R. Anitha, "CTS-IIoT: Computation of time series data during index based de-duplication of Industrial IoT (IIoT) data in cloud environment," *Wireless Pers. Commun.*, vol. 129, no. 1, pp. 433–453, Mar. 2023.

[3] S. Ding, A. Tukker, and H. Ward, "Opportunities and risks of Internet of Things (IoT) technologies for circular business models: A literature review," *J. Environ. Manag.*, vol. 336, Jun. 2023, Art. no. 117662.

[4] P. Verma and N. Bharot, "A review on security trends and solutions against cyber threats in industry 4," in *Proc. 3rd Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, 2023, pp. 397–402.

[5] A. Mahmood et al., "Industrial IoT in 5G-and-beyond networks: Vision, architecture, and design trends," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4122–4137, Jun. 2022.

[6] R. Pareriya, P. Verma, and P. Suhana, "An ensemble Xgboost approach for the detection of cyber-attacks in the Industrial IoT domain," in *Big Data Analytics Fog-Enabled IoT Networks: Towards a Privacy Security Perspective*. Boca Raton, FL, USA: CRC Press, 2023, p. 125.

[7] P. Verma, J. G. Breslin, and D. O'Shea, "FLDID: Federated learning enabled deep intrusion detection in smart manufacturing industries," *Sensors*, vol. 22, no. 22, p. 8974, Nov. 2022.

[8] S. S. Mathew, K. Hayawi, N. A. Dawit, I. Taleb, and Z. Trabelsi, "Integration of blockchain and collaborative intrusion detection for secure data transactions in industrial IoT: A survey," *Clust. Comput.*, vol. 25, no. 6, pp. 4129–4149, 2022.

[9] M. Nuaimi, L. C. Fourati, and B. B. Hamed, "Intelligent approaches toward intrusion detection systems for Industrial Internet of Things: A systematic comprehensive review," *J. Netw. Comput. Appl.*, vol. 215, Jun. 2023, Art. no. 103637.

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

[11] X. Wang, Y. Han, V. C. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[12] N. Bharot, P. Verma, S. Sharma, and V. Suraparaju, "Distributed denial-of-service attack detection and mitigation using feature selection and intensive care request processing unit," *Arab. J. Sci. Eng.*, vol. 43, pp. 959–967, Feb. 2018.

[13] T. Zoppi, A. Ceccarelli, and A. Bondavalli, "Unsupervised algorithms to detect zero-day attacks: Strategy and application," *IEEE Access*, vol. 9, pp. 90603–90615, 2021.

[14] Y. Guo, "A review of machine learning-based zero-day attack detection: Challenges and future directions," *Comput. Commun.*, vol. 198, Jan. 2023, pp. 175–185.

[15] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[16] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.

[17] M. Anisetti, F. Berto, and M. Banzi, "Orchestration of data-intensive pipeline in 5G-enabled edge continuum," in *Proc. IEEE World Congr. Services (SERVICES)*, 2022, pp. 2–10.

[18] P. Verma, S. Tapaswi, and W. W. Godfrey, "A request aware module using CS-IDR to reduce VM level collateral damages caused by DDoS attack in cloud environment," *Clust. Comput.*, vol. 24, pp. 1–17, Sep. 2021.

[19] P. Verma, S. Tapaswi, and W. W. Godfrey, "An impact analysis and detection of HTTP flooding attack in cloud using bio-inspired clustering approach," *Int. J. Swarm Intell. Res. (IJSIR)*, vol. 12, no. 1, pp. 29–49, 2021.

[20] P. Kumar et al., "PPSF: A privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2326–2341, Jul.-Sep. 2021.

[21] Y. Guo, T. Ji, Q. Wang, L. Yu, G. Min, and P. Li, "Unsupervised anomaly detection in IoT systems for smart cities," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2231–2242, Oct.–Dec. 2020.

[22] J. Inoue, Y. Yamagata, Y. Chen, C. M. Poskitt, and J. Sun, "Anomaly detection for a water treatment system using unsupervised machine learning," in *Proc. IEEE Int. Conf. Data Min. Workshops (ICDMW)*, 2017, pp. 1058–1065.

[23] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: Unsupervised anomaly detection on multivariate time series," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2020, pp. 3395–3404.

[24] M. Abdelaty, R. Doriguzzi-Corin, and D. Siracusa, "DAICS: A deep learning solution for anomaly detection in industrial control systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 2, pp. 1117–1129, Apr.–Jun. 2021.

[25] J.-H. Moon, J.-H. Yu, and K.-A. Sohn, "An ensemble approach to anomaly detection using high- and low-variance principal components," *Comput. Elect. Eng.*, vol. 99, Apr. 2022, Art. no. 107773.

[26] S. M. Nagarajan, G. G. Deverajan, A. K. Bashir, R. P. Mahapatra, and M. S. Al-Numay, "IADF-CPS: Intelligent anomaly detection framework towards cyber physical systems," *Comput. Commun.*, vol. 188, pp. 81–89, Apr. 2022.

[27] B. Ibrahim Hairab, H. K. Aslan, M. S. Elsayed, A. D. Jurcut, and M. A. Azer, "Anomaly detection of zero-day attacks based on CNN and regularization techniques," *Electronics*, vol. 12, no. 3, p. 573, Jan. 2023.

[28] G. Ping and X. Ye, "Open-set intrusion detection with MinMax autoencoder and pseudo extreme value machine," in *Proc. Int. Jt. Conf. Neural Netw. (IJCNN)*, 2022, pp. 1–8.

[29] Y. A. Farrukh, S. Wali, I. Khan, and N. D. Bastian, "Detecting unknown attacks in IoT environments: An open set classifier for enhanced network intrusion detection," 2023, *arXiv:2309.07461*.

[30] Z. Wu, P. Gao, L. Cui, and J. Chen, "An incremental learning method based on dynamic ensemble RVM for intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 1, pp. 671–685, Mar. 2021.

[31] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3930–3944, Mar. 2022.

[32] M. Abdel-Basset, N. Moustafa, H. Hawash, I. Razzak, K. M. Sallam, and O. M. Elkomy, "Federated intrusion detection in blockchain-based smart transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2523–2537, Mar. 2022.

[33] P. Ruzafa-Alcázar et al., "Intrusion detection based on privacy-preserving federated learning for the industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1145–1154, Feb. 2023.

[34] L. Zhao, J. Li, Q. Li, and F. Li, "A federated learning framework for detecting false data injection attacks in solar farms," *IEEE Trans. Power Electron.*, vol. 37, no. 3, pp. 2496–2501, Mar. 2022.

[35] V. Rey, P. M. S. Sánchez, A. H. Celdrán, and G. Bovet, "Federated learning for malware detection in IoT devices," *Comput. Netw.*, vol. 204, Feb. 2022, Art. no. 108693.

[36] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for Industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3962–3977, Mar. 2022.