Eksplorium *p*-ISSN 0854-1418
Volume 46 No. 1, May 2025: 1082–1101 *e*-ISSN 2503-426X

Agile Intrusion Recognition Scheme using Federated Learning for SCADA Systems

[1]T. John Sunder Singh, [2]Dr. J. I. Sheeba, [3]Dr. S. Pradeep Devaneyan
[1]Research Scholar, Department of Computer Science and Engineering,
Puducherry Technological University, Puducherry – 605014, India
email: johnsundersingh@ptuniv.edu.in
[2]Associate Professor, Department of Computer Science and Engineering,
Puducherry Technological University, Puducherry – 605014, India
email: sheeba@ptuniv.edu.in
[3]Professor, Department of Mechanical Engineering,
Sri Venkateswara College of Engineering and Technology, Puducherry – 605012, India
email: pr.signs@gmail.com

Article Received: 22 Feb 2025, Revised: 25 April 2025, Accepted: 07 May 2025

Abstract: Supervisory Control and Data Acquisition (SCADA) systems provide itinerary assimilation of industrial hardware and software for remote handling and control. Intrusions in such systems are vulnerable to seizing the legitimate device control for adversarial purposes. To handle such intrusions, an Agile Intrusion Recognition Scheme (AIRS) is presented in this article. This scheme is designed to identify and mitigate layered attacks in SCADA systems. The entry and control points of the intrusions in the system layers are identified using accumulated data logs at the end of disseminated controls. Such logs are analyzed using federated learning at different layer synchronization points. If the synchronization fails then the changes caused entry is marked as an intrusion. The federated learning is responsible for validating the synchronous points between control broadcasting and data acquisition intervals. The synchronization failure in the least intervals is reverted with new control and entry points. This process is optimal for detecting random and frequent intrusions in any control interval of the SCADA systems.

Keywords—Control Broadcast, Federated Learning, Intrusion Detection, SCADA, Synchronization Check

INTRODUCTION

Maintaining SCADA system security is essential for safeguarding crucial infrastructure. Strong user authentication and access restrictions are implemented as part of robust security techniques [1]. An additional line of protection against possible cyber threats is provided by the use of encryption in communication [2]. To fix known vulnerabilities and preserve general security, regular system upgrades and patching are crucial [3]. Segmenting a network makes it easier to prevent unauthorized access while using intrusion detection systems makes it easier to spot and address such breaches [4]. System logs and user activity are continuously monitored to give real-time insights and enable quick fixes for security vulnerabilities. Programs for employee training help create a culture that is aware of cybersecurity issues and encourages proactive defense against changing threats [3, 5]. A thorough security plan is ensured by cooperation between cybersecurity specialists and SCADA operators, and regular security audits assist in evaluating and fortifying the system's defenses [6].

Intrusion detection is crucial for protecting SCADA systems and vital infrastructure. The system uses methods such as anomaly detection and user behavior analysis to identify both known and unknown cyber threats [7]. Systems like Network Intrusion Detection

Systems (NIDS) are used to maintain the safety and security of the SCADA network [8]. Real-time data is acquired by continually monitoring system logs, network traffic, and user behavior, which enables you to respond quickly to any security concerns [9]. Limiting access, encrypting communications between SCADA components, and making sure that only authorized users may use the system are further security precautions [10]. Vulnerabilities need to be updated and corrected often to maintain system security. Constant training and cooperation between cybersecurity experts and SCADA operators provide proactive threat control. Constant training and cooperation between cybersecurity experts and SCADA operators provide proactive threat control [11].

Integrating machine learning into SCADA systems for intrusion detection improves cybersecurity by instantly examining network traffic patterns [12]. Even in complicated circumstances, these algorithms can recognize abnormalities and possible assaults in response to developing threats [13]. While unsupervised learning finds new threats without specified labels, supervised learning increases accuracy by training on labeled datasets. This flexibility is essential for dealing with the ever-changing landscape of cyber threats [14]. Through continual development over time, reinforcement learning further optimizes intrusion detection systems based on user input [15]. Machine learning's proactive approach reduces reaction times, enhancing the critical infrastructure's overall resilience. Machine learning is more successful at protecting SCADA systems against new cybersecurity threats when it is supported by ongoing monitoring and updates based on the most recent threat data [16].

CONTRIBUTIONS

- The proposal of an agile intrusion recognition scheme for detecting intrusions in SCADA systems comprising interoperation controllers and devices
- The assimilation of federated learning for identifying synchronization failures through large data acquisitions and control dissemination
- The real-time data incorporated analysis using self-metrics and performing a comparative analysis using distinct metrics and methods

RELATED WORKS

Ahakonye et al. [17] devised a method to identify intrusions in SCADA networks. The method combines a potent machine learning classifier with an unbiased feature selection approach, including key phases like data preparation. A modified decision tree and Chi-square feature selection contribute to the effectiveness of the proposed intrusion detection system. The approach enhances the security of real-time SCADA networks by developing an advanced intrusion detection system.

Ndonda et al. [18] investigated the temporal patterns of state transitions to identify intrusions in ICS/SCADA. The goal was to enhance Industrial Control Systems (ICS) security by introducing a novel time-based detection system. The method involves creating a learning mechanism to recognize and detect unusual activities within the physical processes of ICS by leveraging temporal data features. The method significantly elevates the level of security for Industrial Control Systems.

Öztürk et al. [19] created a machine learning-driven intrusion detection system tailored for SCADA systems in healthcare. The aim is to boost the security of energy distribution and cyber-physical systems vulnerable to cyber threats. The system utilizes artificial intelligence and machine learning techniques to identify and categorize potential attack threats. This approach contributes to the heightened security of SCADA systems in the healthcare sector.

Ahakonye et al. [20] introduced CH-DT, a method aimed at detecting intrusions in high-dimensional data within SCADA networks. The goal is to enhance the security of real-time SCADA networks within the industrial Internet of Things (IIoT) environment. CH-DT proves to be an effective technique for bolstering the security of SCADA networks in the context of the IIoT. Empirical results demonstrate the model's reliability in accurately detecting anomalies with minimal computational requirements.

Al Ghazo et al. [21] identified critical attack sets within attack graphs, specifically focusing on computer and SCADA/ICS networks. The primary objective is to enhance security by detecting highly critical cyber-attacks in SCADA/ICS networks. The proposed model demonstrates superior performance, surpassing previous models in terms of both accuracy and speed. The approach proves effective in identifying and addressing highly critical cyber threats in SCADA/ICS networks.

Nguyen et al. [22] introduced a stacking ensemble of tree-based models for intrusion detection in SCADA systems. The introduced model is used to enhance the precision level of intrusion detection in SCADA systems. A meta-classifier is used here to classify the types of intrusions. It minimizes the computational complexity and cost during the detection process. The introduced model elevates the performance and efficiency range of the systems.

Rabie et al. [23] developed a perceptron stochastic neural network (PSNN) based intrusion detection model for SCADA systems. The developed model is used to detect the vulnerabilities and attacks that reduce the performance level of the systems. The PSNN-based model analyzes the dimensional features that produce the optimal dataset for further processes. The developed model reduces the latency and complexity ratio in intrusion detection.

Barsha and Hubballi [24] proposed an anomaly detection model for SCADA systems. The proposed model is used to evaluate and analyze the cyber-attack. It detects the sequencing anomalies which increase the computational cost of the systems. The proposed model also minimizes the energy consumption ratio while performing the detection process. The proposed model elevates the quality of services (QoS) and feasibility range of the SCADA systems.

Saheed et al. [25] developed a hybrid ensemble learning method for anomaly detection in industrial sensor networks and SCADA systems. The method introduces a unique hybrid Ensemble Learning Model designed for intrusion detection in SCADA systems integrated with Industrial Sensor Networks (ISNs). The method plays a key role in enhancing the overall cybersecurity of SCADA systems. The approach contributes to strengthening the cybersecurity measures for SCADA systems.

Diaba et al. [26] created a SCADA security system using deep learning to prevent cyber infiltrations (CI). The method blends a Genetically Seeded Flora algorithm with a

Transformer Neural Network to spot changes in how things operate. The approach offers an advanced solution to strengthen the cybersecurity of SCADA systems by effectively identifying and preventing potential CI. The developed system proves effective in bolstering the security of SCADA systems against CI.

Zheng et al. [27] presented a semi-supervised multivariate time series anomaly detection method for wind turbines. The primary objective is to reduce wind turbine maintenance costs and minimize unplanned downtime. The model integrates a reconstruction model and an auxiliary discriminator, collectively focusing on efficient pattern extraction from multivariate time series data. The method enhanced anomaly detection performance and operational efficiency in wind turbine systems.

Shlomo et al. [28] developed a method to detect malicious activity patterns in SCADA systems based on temporal patterns. The primary objective is to enhance the security of SCADA systems by addressing concerns related to the manipulation of temporal patterns. The method offers a robust solution for improving SCADA security by identifying and mitigating malicious activities. The method demonstrated effectiveness in strengthening SCADA security against various threats.

Oyucu et al. [29] introduced an ensemble learning framework for distributed denial of service (DDoS) detection in SCADA systems. The introduced framework identifies the DDoS attacks that are presented in SCADA systems. It is used to protect the system from DDoS attacks. When compared with others, the introduced framework increases the accuracy level of the detection process.

Upadhyay et al. [30] suggested a way to detect intrusions in power grids using SCADA, improving accuracy through smart feature selection and teamwork. The primary goal is to create an efficient intrusion detection system tailored for SCADA-based power grids. The approach uses RFEXGBoost to choose important features and a majority vote ensemble technique to enhance overall detection capabilities. The method performs better than previous intrusion detection methods.

Anwar et al. [31] enhanced anomaly detection in SCADA networks by extending attributes. The aim is to make the method for detecting unusual network activities in vital SCADA systems using support vectors even better. The proposed method achieves this improvement by incorporating behavioral attribute extension for network nodes. The approach demonstrates higher F1 score (from 0.6 to 0.9) and Matthews's correlation coefficient (from 0.3 to 0.8).

Intrusion detection in SCADA systems relies on individual features [17, 13] of the devices or controller operation patterns [18, 28] as discussed above. This encounters a variation feature/pattern-based operation detection where a specific set of intrusions is alone detectable. Different from these methods, cyber-physical system-based intrusion methods [23] rely on graph assessments [22] with semantics. This factor is vulnerable to strong intruders by emphasizing one-point security. The problem is the method's robustness and sustainability due to imperfect controller and device synchronization results in scanning/ device identity exposures. To address such issues, a controller-device synchronized verification-based intrusion detection scheme AIR is introduced in this article.

PROPOSED AGILE INTRUSION RECOGNITION SCHEME

Supervisory Control and Data Acquisition (SCADA) remotely observes the devices in the industrial infrastructure. It is processed in several infrastructures such as power generation, gas refining, transportation, etc. This system operates on both the software and hardware component which remotely collects the data from the industrial devices. It automates industrial automation where it constructs two features such as remote terminal units (RTU) and Programmable logic controller (PLC). It is a control system architecture where the graphical user communicates with each other in high-level supervision machines. In Fig. 1 the proposed scheme is illustrated.

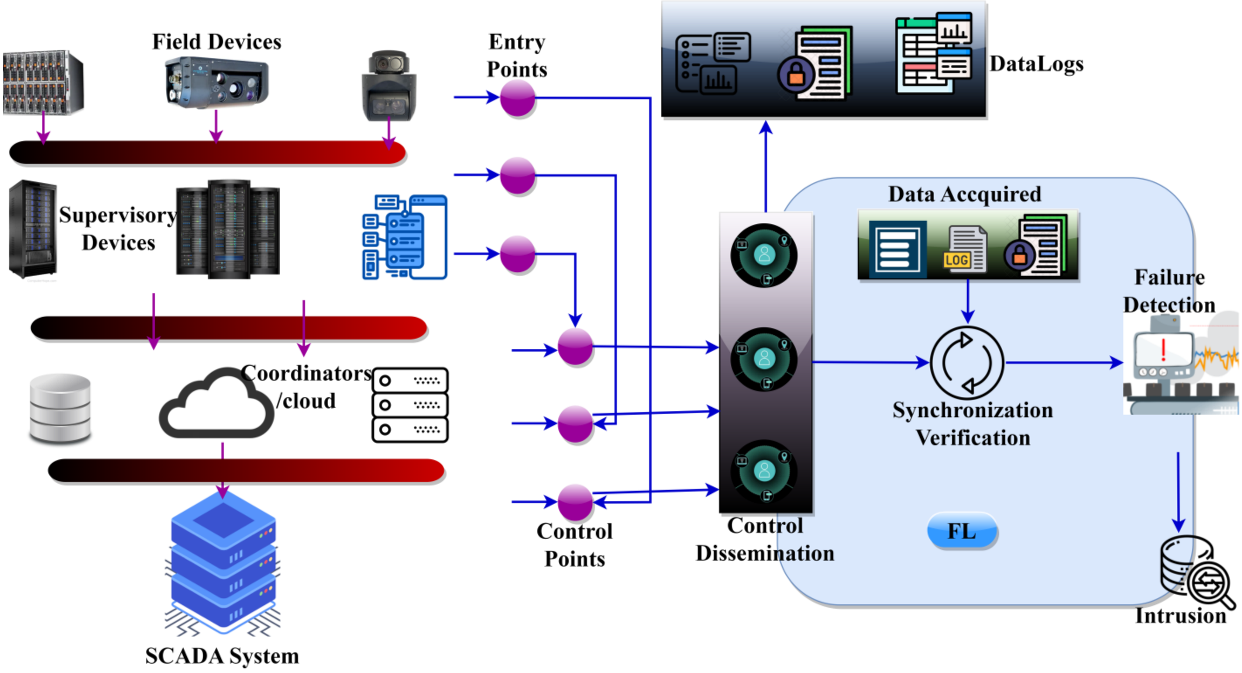

Fig. 1 Proposed AIR Scheme Illustration

This work concentrates on IoT SCADA that includes cloud computing and central controller. Based on this computation this system provides itinerary assimilation of industrial hardware and software for remote handling and control. Intrusions in such systems are vulnerable to seizing the legitimate device control for adversarial purposes. The preliminary step in this SCADA process is to find the remote-terminal units to collect the data from the industrial equipment and it is equated below.

$$M_u = \frac{1}{v_n} * \left[\left(o' + a_0/v_0 \Big/ r_e \Big/ v_n\right)\right] + \textstyle\sum_{o'}[(a_0+..+a_n) * (v_0+..+v_n)] + \left(\frac{\sum r_e + v_0 / o'}{a_n / v_n}\right) * \left[\left(\textstyle\sum_{1/a_n}(o' + v_n) * r_e\right)\right]$$

(1a)

In the above equation, Remote Terminal Units (RTU) functions are validated and represented as M_u, the data is a_0, n-number of data is a_n, the device is described as d_0, whereas, the n-number of devices is labeled as d_n. Here, the collection of data from the industrial devices is symbolized as o', and the remote devices are r_e. From this data are collected from the sensor and the actuators and forwards to the remote devices. Based on this processing, the components relate to the industrial settings in which it monitors the remote device working in industrial units. From this processing, the collection of data is associated with the software and hardware components. In this collection of n-number of data from n-number of devices are expressed as $\sum_{o'}[(a_0 + .. + a_n) * (v_0 + .. + v_n)]$.

From this RTU the necessary data is collected and stored in the system and it controls the remote device accordingly and is represented as $\left(\frac{\sum r_e + {v_0}/_{o'}}{{a_n}/_{v_n}}\right)$. In this step, the industrial devices are interlinked with each other in their operations if one leads to failure then remotely the other falls. To illustrate this issue the RTU is developed among the industrial devices and improves the synchronization. For this verification phase among the devices, Federated learning is introduced in this work, by getting into the learning method the communication infrastructure is overviewed between RTU and central controller and it is equated as follows.

$$U' = \prod_{v_0}^{o'}(a_0 + t_s) + \left(\frac{M_u}{\sum_{t_s}(v_n + r_e)} \Big/ \frac{(T_n + a_0)}{o'}\right) * \left(\frac{\sum_{M_u}(v_0 + o')}{{1}/{d_n}}\right) + \left[\left(\frac{(d_n + t_s)}{[(r_e + a_n) + (o' + T_n)]}\right)\right] *$$
$$(T_n + r_e)$$

(1b)

The communication is developed for the data and the devices and it is labeled as U'. In this process, communication is established between the RTU and central controller and it is formulated as T_n. From this, the collected data are kept in the storage and from which it acquires the information regarding the industrial devices. Based on this computation the data relies on the central controller which holds the interconnection among the filed device, supervisor, and co-coordinators cloud. In this step, it processes the better communication link between the RTU and central controller in the SCADA system. Based on this, it relies on the secure link establishment among the layer which is discussed above. This formulation is used to find the secure link for the central controller in the SCADA system.

From this establishment, the industrial devices are associated with the transmission of the secure data to the remote devices. If there is any fault occurs then verification is examined on that part which acts as the checkpoint. In this format, the entry and the control points are associated with the remote devices whereas the central controllers are associated with the collection of data and it is equated as $\left[\left(\frac{(d_n + t_s)}{[(r_e + a_n) + (o' + T_n)]}\right)\right]$. In this manner, the communication infrastructure is used to state the RTU and the devices on the industrial stand. The structural variants are used to provide a better understanding of data communication in SCADA systems. Keeping this communication establishment into account, the interconnection of the device in the SCADA system is deliberated in the below derivation.

$$E_d, S_u = \begin{cases} \prod_{v_0}(r_e + t_s) * \left(o' + {v_n}/_{r_e} + a_n\right) + \left(\frac{s' + c'}{\sum_{T_n} U'}\right), \mathit{Field\ device} \\ \left[\left((a_0 + o') + \sum_{k_g}(p_0 * v_n)\right)\right] * \left(\frac{o' + U'}{\prod_{c'}(t_s + v_0)}\right), \mathit{Supervisory\ device} \end{cases}$$

(2a)

The above derivation is the combination of field and supervisory devices and they are symbolized as $E_d\ and\ S_u$. From this processing step, the first condition states the collection of data from the sensor or control unit and forwards to the remote device in industrial devices. From this processing, communication is established among the devices and the sensor units. This infrastructure holds the field-oriented devices where the sensing is carried out for the data from the infrastructure and it is represented as $\left(\frac{s'+c'}{\sum_{T_n} U'}\right)$. In this field device, the communication is followed up for the n-number of devices in which it is associated with the sensor and control units.

The second condition is the supervisory device which handles the collection and processing of data and is labeled as p_0. The periodic checking is examined in this case and it is described as k_g. Thus, collecting the data and processing is illustrated in these supervisory devices and it is equated as $\left((a_0 + o') + \sum_{k_g}(p_0 * v_n)\right)$. In this manner, supervisory devices acquire the data from the field data and process where checking is carried out for the collection of data from the industrial devices. Based on this processing step, the entry point and control points are given as the input for the system which is elaborated in co-coordinators and central controller. These two device parameter characteristics are formulated below.

$$C_i, T_r =$$

$$\left\{\overbrace{\left(M_u * \frac{c'+o\prime}{\sum_{r_e}(t_s+k_g)}\right) + \left[\left((k_g + v_0) + \prod_{a_0}(r_e + t_s)\right)\right]}^{Coordinators} \middle| \overbrace{\sum_{r_e}^{a_0}(k_g + c') * \left(M_u + \frac{U' * k_g}{v_0 + t_s}\right)}^{Central\ controller}\right\}$$

(2b)

The coordinator and central controller are processed in SCADA, and it is represented as C_i. The checking is examined here where the remote transmission is carried out for the control unit. Here, the central controller checks for the process in real-time and illustrates the industrial organization. Based on this section the central controller gathers the information of the device and executes them in the real-time environment. In this section, both layers notify the device's computation where the transmission runs through the RTU and central controller in this SCADA. In this section, both the coordinator and central controller lead to the data acquisition which defines the checking periodically for the task assigned to the device. Based on the assignment of the task it delivers the control to the remote device. The controller and field device integration process is illustrated in Fig. 2. This integration is based on the monotonous function perceived for entry and control points.

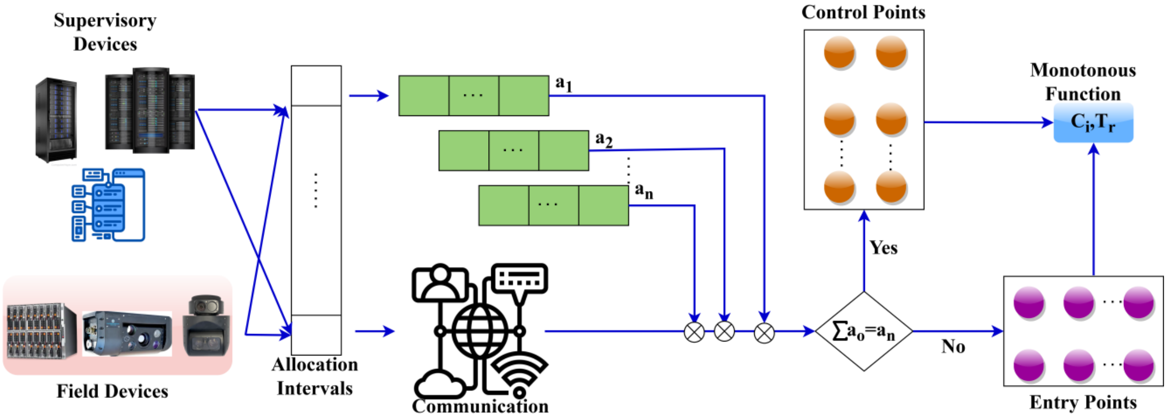

Fig. 2 Integration for Entry and Control Points for Monotonous Function

The prime requirement of the integration for monotonous functions is $\sum a_o = a_n$. The supervisory controllers/ devices generate a_n for either entry/ control points. The field devices perform interval U' for communication with a_o data. If both are the same then control dissemination occurs. Whereas the change deflects the assignment of new entry points across various r_e. Thus the (E_d, S_u) is differentiated based on (C_i, T_r) for monotonous function demands. Therefore M_u is controlled from entry to dissemination by the supervisory devices. This is an ideal case where no intrusion is observed. If this scenario is violated then intrusion detection occurs (Fig. 2). Here, checking is carried out promptly to forward the data to the appropriate remote device represented as $\left(\left(k_g + v_0\right) + \prod_{a_0}(r_e + t_s)\right)$. The central controller is like the supervisory where the collection of data from the sensor is gathered and transmitted to the appropriate device which defines the control units in the industrial organization. Here communication is built between the RTU and central controller regarding the device data processing and it is equated as $\left(M_u + \frac{U'*k_g}{v_0+t_s}\right)$. The contemporary techniques are followed in this SCADA, post to this the central dissemination working is observed in the below equation as follows.

$$\lambda = \left(\frac{\sum_{M_u}(U'+k_g)}{(e_y+c_p)}\right) + \left[(c' + v_0) * \left(M_u + \frac{(C_i+S_u)*E_d}{t_s}\right)\right] \tag{3}$$

The control dissemination is examined and it is formulated as λ, in which the device acquires the data and transmits it to the remote device. The entry point and control point are symbolized as $e_y \; and \; c_p$. Here, the field device, supervisor, and coordinator layers are associated with the SCADA. In this case, the central controller is used to derive the communication between the RTU and the central controller in which the control dissemination is processed in SCADA. In this approach, entry point data runs on the devices from field to coordinate device, and from this central controller is pragmatic with the control points which are given as the input for the control dissemination in the system.

The main concept of this process is to share the data to the control dissemination where it is fetched from the entry point of field, supervisory, and coordinators. Based on this process the communication checking is carried out promptly envelope the RTU and it is formulated as $\left(\frac{\sum_{M_u}(U'+k_g)}{(e_y+c_p)}\right)$. Thus, the control dissemination is examined in this approach by providing the control unit for the data transmission for the remote device. Here, the control dissemination is evaluated for the communication checking between the shared data in SCADA. From this approach, the data logs are illustrated for the control dissemination, and the features of data are formulated as follows.

$$d_a = \left[\left(\lambda + (e_y + c_p)\right) * M_u\right] + (T_r + C_i + S_u + E_d) \tag{4}$$

The data logs are associated with the field, supervisory, coordinators, and central controller for the control dissemination. Here, it deliberates with the entry and control point that accumulates the data logs and it is represented as d_a. The status is observed for the sensor transmission in which the data logs are detected for the control dissemination. In this data logs are the collection of data that holds the device information in the industrial organization. Here, the computation is used to develop the remote device transmission that examines the parameter from the acquired data. The control dissemination and data acquisition process is represented in Fig. 3.

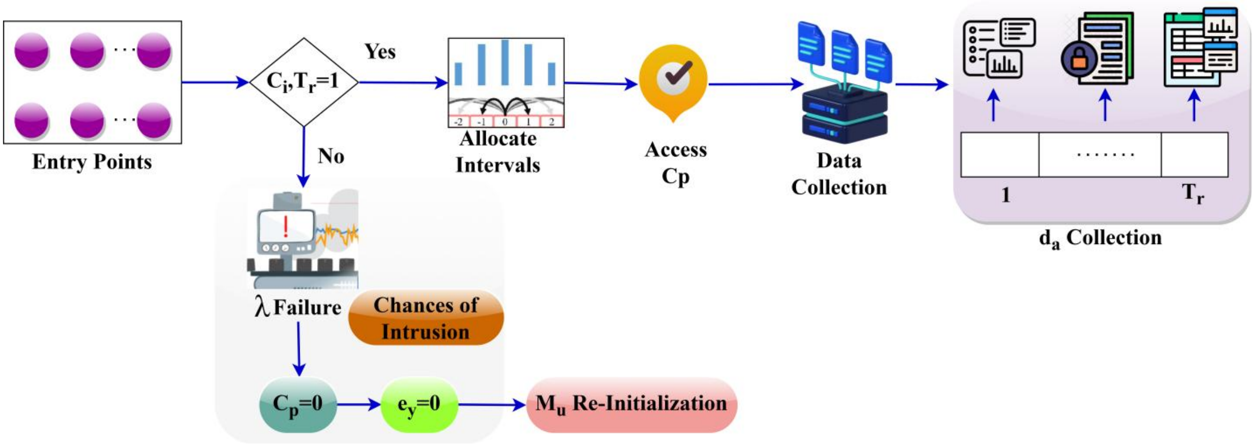

Fig. 3 Control Dissemination and Data Acquisition Process

The λ verification and d_a acquisition is split into two distinct operations. First the $C_i, T_r = 1$ verification passing criteria assigned c_p to pursue device operations, where (c_i, T_r) and (E_d, S_u) are synchronization. As long as these synchronizations are ideal in any interval, the $e_y = c_p$ allocation becomes consistent. The failing (c_i, T_r) results in the closure of c_p and e_y concurrently to prevent further intrusions. Therefore this condition generates the chances for intrusion detection which M_u is re-initiated. Both e_y and c_p connected devices are suspended from the U' and T_n processes for preventing further failures (Fig. 3). The data acquisition is performed for the synchronization of the data in which it holds the RTU and central controller. Here, the data logs hold the collection of information from the entry and control points from the layers of devices. The main purpose of this scheme is to provide the central coordinators' cloud and process the computation step. In this case, the SCADA handles the control dissemination in the system that processes the entry and control point and it is formulated as $\left[\left(\lambda + (e_y + c_p)\right) * M_u\right]$. From this data log the following section discusses the Federated learning.

FEDERATED LEARNING FOR SYNCHRONIZATION VERIFICATION

Federated learning is a decentralized device function-centric process that is associated with the training model. Collaborative learning is deliberated with the privacy data processing that examines the control dissemination. It enables the industrial organization that generate a training methodology that includes three factors such as learning coordinates, data contributors, and user model. Here, it deploys the communication among the layers that provide the training phase in which checking is examined for n-number of parameters. The following equation is used for the training of data from the industrial device.

$$t' = \frac{1}{a_n+v_n} * \sum_{T_r}(d_a + \lambda) + k_g(r_e) * l_r - (e_y + c_p)$$

(5a)

The training is performed in the federated learning that deploys the control dissemination and data logs that hold the collection of data from the industrial devices. This state of training is represented as t', in which it requires the essential form of data transfer in which the checking is performed, the failure is described as l_r. In this checking, both the entry and control points are verified both junction data have the failure and it is isolated from the processing step. In this manner, the training is used to identify the failure data train in the federated learning network and improve the computation step. Here, the entry and control points are associated with this scheme and detect the failure and train them, from this approach, the update model parameter is observed from the trained model from the central controller and it is formulated in the below equation as follows.

$$d_t = \prod_{k_g}(v_0 + o') * \left(\frac{t_s + t'}{q_a/d_a}\right) + U' * T_r + S_u$$

(5b)

The model parameter deliberates with the trained model that deploys the central controller and it is labeled as d_t. This processing step includes the data acquired in the control system and it is represented as q_a. Here, it states the central controller to attain the entry and control point for the data input in the federated learning process. Based on this section of the training model parameter includes the data log in which the status bar is illustrated in the central controller where the control point feeds the input to the control dissemination. Thus, the update is processed in this case for the model parameter from the central controller, and from this computation step, the synchronization verification is performed from the control dissemination and it is equated below.

$$Y = \left[\lambda + \left(e_y + c_p\right) + d_a\right] * \prod_{o'}(v_0 + c') * q_a \qquad (6)$$

The synchronization verification is processed from which it acquires the input from the control dissemination and data-acquiring method and it is described as Y. In this scheme, the entry and control points are associated with the different level of computation. Here, the data points, are deliberated with the synchronization verification where the failure occurs due to the entry and control point in the SCADA system. The verification process using federated learning is illustrated in Fig. 4.

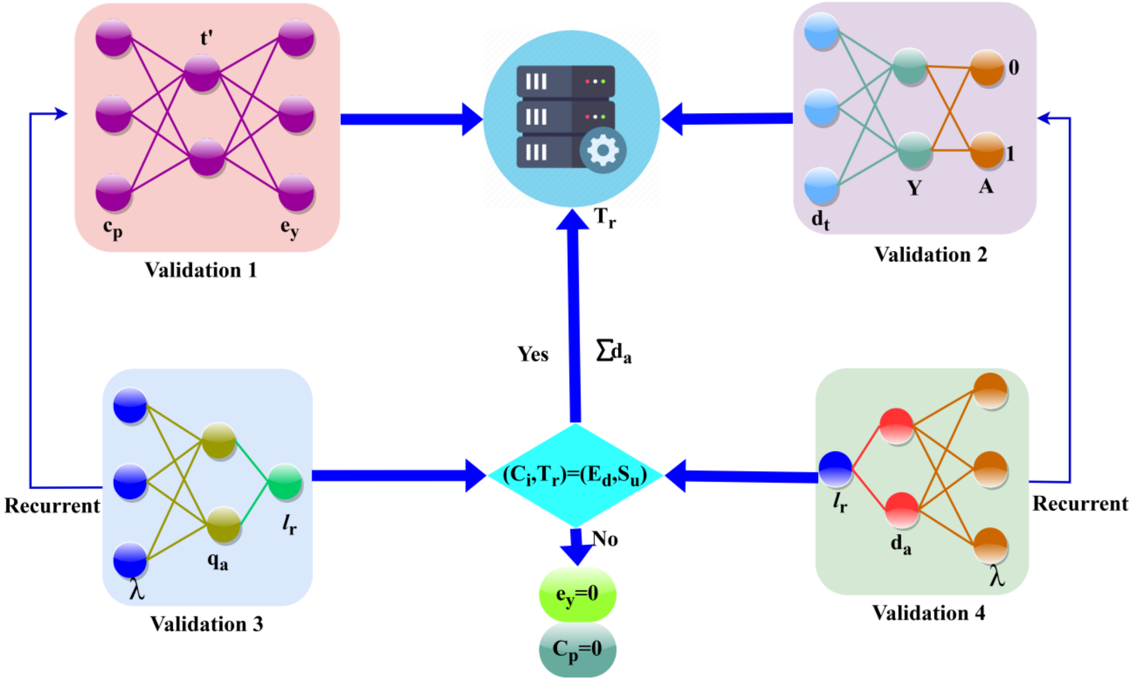

Fig. 4 Verification Process Illustration

The learning process verifies λ and Y under 4 validations represented in Fig. 4 above. Validations 1 and 2 are instigated by the controller device to ensure $c_p = e_y$ and $A \neq 0$ observations. If these observations are true then the λ is forwarded to the field devices. Therefore, (C_i, T_r) and (E_d, S_u) are synchronized together acquiring d_a for assessment. The failing conditions (i.e.) $A = 0$ and $c_p \neq e_y$ halts the current M_u and thus new synchronization is pursued. The validation 3 and 4 are focused on l_r extraction based on λ to d_a and λ to q_a in consistencies. If these validations fail to reduce the synchronization, then $e_y = c_p = 0$ is the halt condition detecting intruders. The passing condition ensures data logs for further assessment. The entry and control points of the intrusions in the system layers are identified using accumulated data logs at the end of disseminated controls. Such logs are analyzed using federated learning at different layer synchronization points. If the synchronization fails then the changes caused entry is marked as an intrusion. Post to this verification phase, data acquiring validation is performed for detecting failure, derived as follows.

$$A = \begin{cases} 1, if\ (q_a + \lambda) + \left(d_a * \frac{T_r + Y}{\Pi(e_y + c_p)}\right) \\ \quad\quad 0,\ Otherwise \end{cases} \tag{7}$$

The data acquiring is processed from the control dissemination and performs the validation to identify a failure, represented as A. From this observation, the central controller is associated with the remote device and provides reliable computation in SCADA. Here, the computation process is reasonable from the entry and control point where validation is executed. If it is error data acquired from the device then it is failure which is defined as 1, or else the otherwise condition is executed. From this condition, the central controller is associated with the validation of data acquired for better results. Post to this re-training on failure and checking synchronization verification is executed based on control dissemination which is formulated below.

$$r(t\,) = (\lambda + d_a) * t'(d_t) + T_r * A \tag{8}$$

The re-training on failure and checking the synchronization is verified from the control dissemination and it is equated as $r(t)$. Here, the central controller is associated with the validation process which failure is detected and synchronization checking is performed. If the synchronization for different entry points is detected then the federated learning is processed with the training phase. The training phase indicates the entry and control point in which it detects the failure of the training given to the particular device. From this re-training is processed to check the synchronization verification from the control dissemination. Thus, the re-training is executed for the synchronized data, and from this federated learning is responsible for validating the synchronous points between control broadcasting and data acquisition intervals which are equated below.

$$N = (\lambda + v_0) * \left(r_e + \frac{t_s + r(t\,)}{q_a + a_0}\right) + A + d_t \tag{9}$$

The analysis is processed for the control dissemination of the device in which the data acquired from the synchronized verification is detected and it is represented as N. In this stage, validation is executed for the update of the data from the transmission point and finds the better device data analysis. This analysis is followed up with the data acquisition which provides the control dissemination from the remote device. The data interval points are detected in this case by finding the central controller in this stage. Thus the validation is performed for the broadcasting and data acquisition intervals are analyzed in the above derivation, the forthcoming process is intrusion detection which is equated below.

$$F = \left.\begin{matrix} \left[\left(a_0 * k_g\right) + t_s\right] + T_s * Y * A, = 0 \\ \sum_{r_e} r(t\ ) * (v_0 + T_r) + \lambda = \emptyset \end{matrix}\right\}$$

(10)

The identification of intrusion is processed and it is described as F. In this stage, the first condition is equal to 0, so the intrusion is detected where the synchronization is carried out here. Whereas, in the second condition no synchronization occurs in this computation so the data is null, in other cases, if there is no dissemination is detected is also termed as the null set. Thus, the identification of intrusion is calculated in this process where the failure is detected in this methodology. The decision on detecting intrusion from the N output is illustrated in Fig. 5.

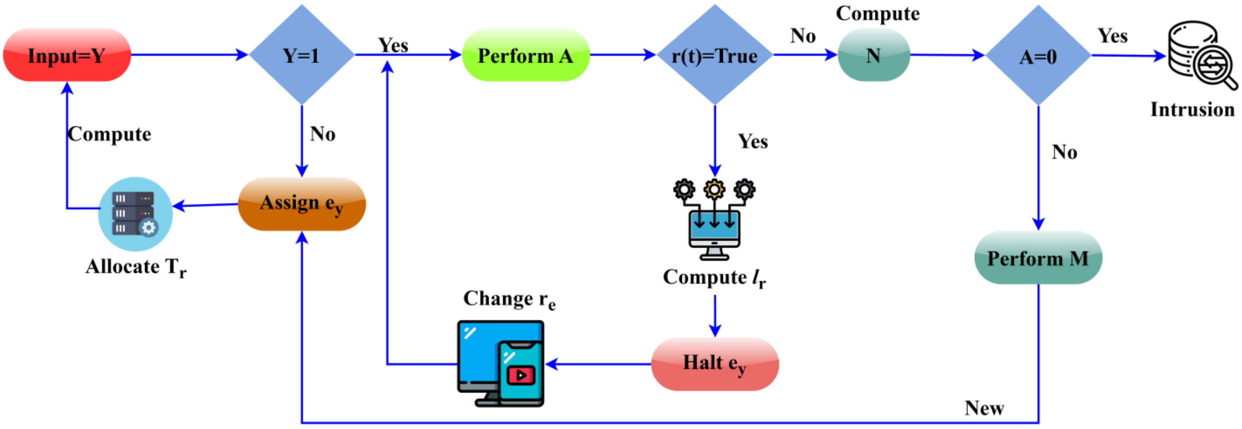

Fig. 5 Decision Process for Detecting Intrusion from *N*

The decision for intrusion detection is presented in the above Fig. 5. The decisions are chained for $Y, r(t)$, and A for their true/ false outputs. The intrusion detection follows $Y = 1, r(t) = false$, and $A = 0$ satisfying conditions. If these conditions fail, then $r(t) = true$ alone validates the chances for intruders whereas $A \neq 0$ insists on performing M. This is repeated from e_y where in c_p is unavailable. Similarly the e_y and c_p imbalance requires new intervals to increase the chances of q_a and d_a to validate the U' for detecting l_r. This is therefore performed using t' other than $r(t)$ to increase detection. This is therefore performed using t' other than $r(t)$ to increase detection. At this point, the SCADA system detects intrusion by estimating N from federated learning. The synchronization failure in the least intervals is reverted with new control and entry points. This process is optimal for detecting random and frequent intrusions in any control interval of the SCADA systems.

RESULTS AND DISCUSSION

This article endorses the "WUSTL-IIOT" dataset [32] based on SCADA cybersecurity research. The following intrusions are considered in this analysis: port scanning, address scanning, device detection, and exploitation. Based on the detected network traffic between r_e and T_r, the data for 25 hours is accumulated; this augmented 70K+

observations for maximum intrusion detection. This dataset is observed from 12 controllers with a random set of controls and therefore we consider 10 broadcasts per interval. With this information, F observed for the 4 different adversaries is presented in Fig. 6.

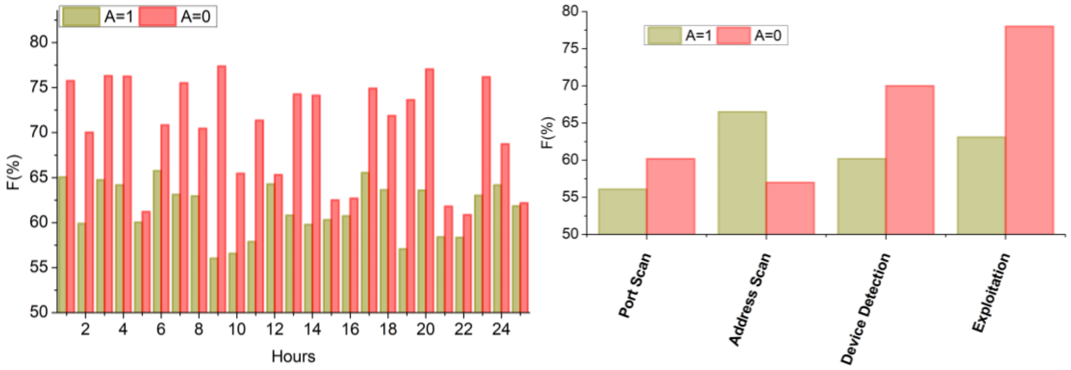

Fig. 6 $F(\%)$ Analyses for Hours and Intrusions

The $F(\%)$ for the observation hours and different intrusions are analyzed in the above Fig. 6. The $r(t)$ and t' instances in the (E_d, S_u) process correlates the device allocation for monotonous M_u. If this fails then the type of adversary pursued is detected in T_r or c_p disseminations. Thus the federated learning differentiates λ and Y f or different intervals maximizing F. In this case, the q_a and $d_a \in N$ are used for verifying the intrusion present. However the l_r due to different intrusions during t' and $r(t)$ is different. This analysis is presented in Fig. 7.

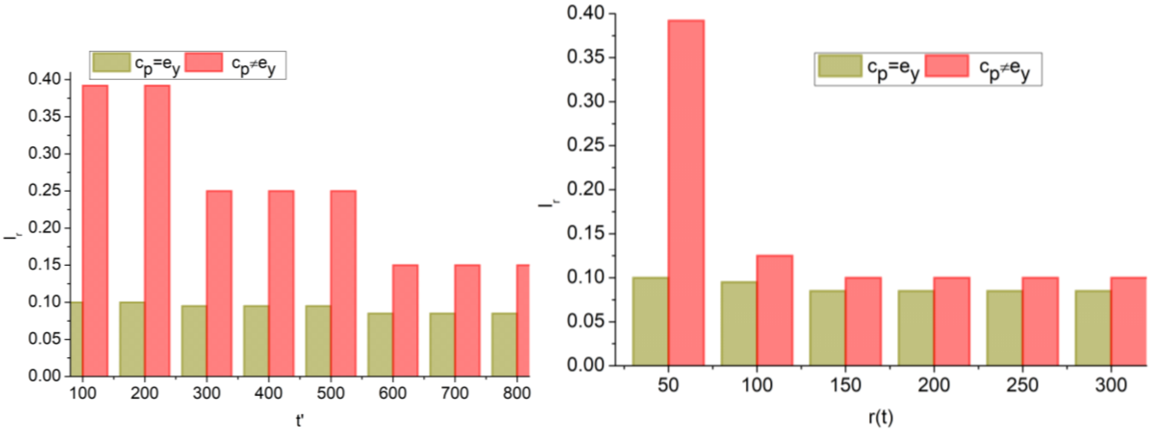

Fig. 7 l_r Analyses for t' and $r(t)$

The l_r is reduced using t' and $r(t)$ iterations as presented above. The chances of $t' > r(t)$ is nominal if $c_p \neq e_y$ and therefore M_u is monotonous. If this is violated then the chances of intrusions are high resulting in high failures. Therefore the control broadcast is confined over different intervals preventing d_t. In such cases, the different assessment nodes of the learning process instigate the need for new e_y allocation between the devices. Therefore the l_r is reduced in t' at a high rate compared to $r(t)$ that is rectified by allocating T_r (Fig. 7).

COMPARATIVE ANALYSIS

In this section, the comparative analysis discussion using detection ratio, control broadcast, synchronization failure, detection time, and data acquisition metrics is presented. The metrics are analyzed under different controls/intervals (1 to 10) and number of controllers (1 to 12). Also, the proposed scheme is compared with TPMAD [28], TPAIDS [18], and ELM-AD [25] methods that were discussed earlier.

DETECTION RATIO

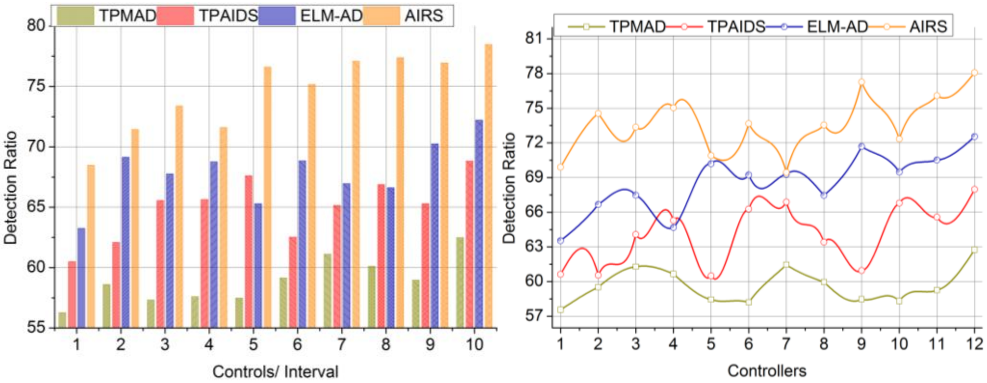

Fig. 8 Detection Ratio

The detection ratio is improved in this proposed work for varying control intervals and controllers (Fig. 8). Here, the intrusion is detected for the device's computation in the industrial organization. In this approach, the transmission is carried out from the field device to the supervisory and then to the coordinators' cloud where the entry and control points are used to forward the input. From the input, the computation is started and finds whether there is any intrusion is detected or not. In this category, the control intervals are associated with the communication infrastructure. Based on this processing the synchronization is followed up for this detection process and enhances the result in this work. The controllers illustrate the coordinator cloud input from which the control point forwards the data to the next level. Here, the control dissemination acquires data from the data log and performs the verification where the intrusion is identified and it is equated as $\left[\left(a_0 * k_g\right) + t_s\right] + T_r$. In this processing step, the data are acquired from the input device and forwarded to the appropriate device.

CONTROL BROADCAST

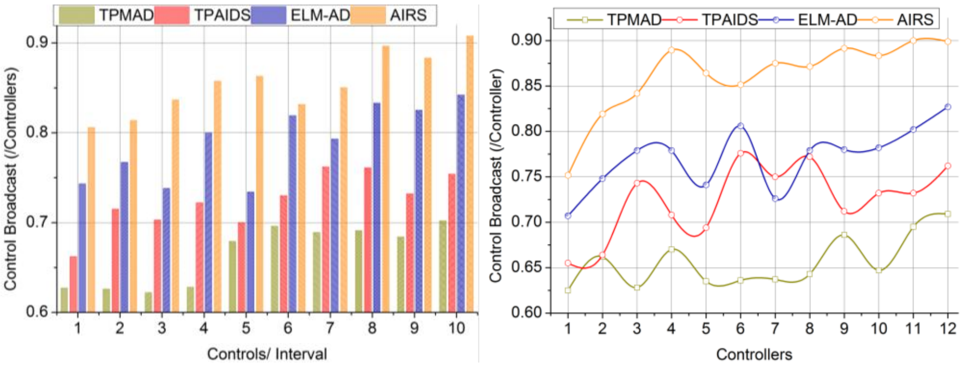

Fig. 9 Control Broadcast

In Fig. 9 the control broadcast improvements are shown considered for the different control intervals ranging from 1 to 10, and controller from 1 to 12. In this scheme, the data is acquired from the coordinators' cloud where it deliberates with the data logs and control dissemination process. The federated learning is introduced to examine the central controller processing among the different control intervals. The data acquired from the cloud environment is used to provide the control point along with the entry point and they generate the data by comparing it with the data logs from which it is linked to the control dissemination process. Based on this strategy, the central controller and RTU communication are built to transfer the data among themselves. The control broadcast is used to provide better data processing which is associated with the SCADA. The broadcasting is developed for the transmission of the data to the appropriate device on a specific interval of time and it is represented as $\left[\left(\lambda + (e_y + c_p)\right) * M_u\right]$. In this manner, the control broadcast shows better processing in this proposed work.

SYNCHRONIZATION FAILURE

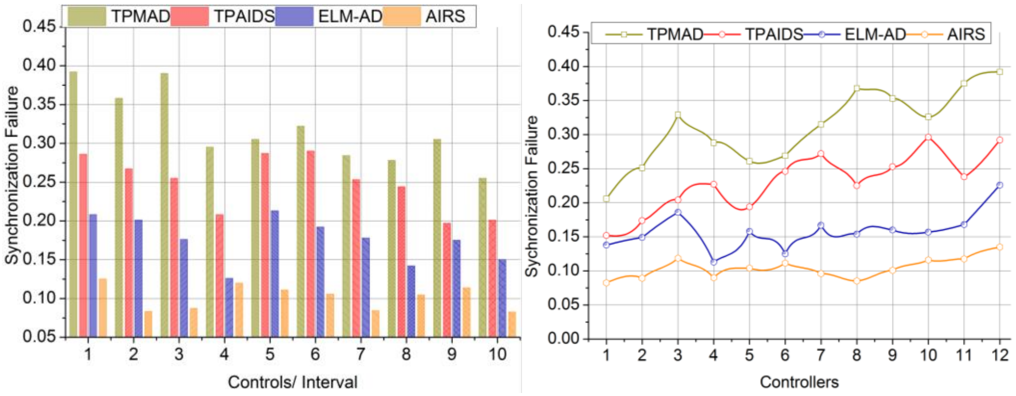

Fig. 10 Synchronization Failure

The synchronization failure is reduced (Fig. 10) for the varying control intervals and controller where it deploys the control dissemination from the control point. Here, the data logs are associated with the status of the data acquired from the field devices and supervisory devices. In this manner, the parameters are observed for the training phase in this federated learning. The training is initiated if there is any failure occurs on the device during the data acquisition and transmission. By using this training the failure is addressed and reduced in this proposed work and it is formulated as$\sum_{T_r}(d_a + \lambda) + k_g(r_e)$. The checking is processed periodically for the data synchronization and finds the error data in the SCADA. The control dissemination is examined for the different intervals of control from the industrial organization. The update is measured for the synchronization between the control dissemination and data logs and shows lesser failure in this work. Thus, the failure is detected in this case, and reduced in this federated learning methodology.

DETECTION TIME

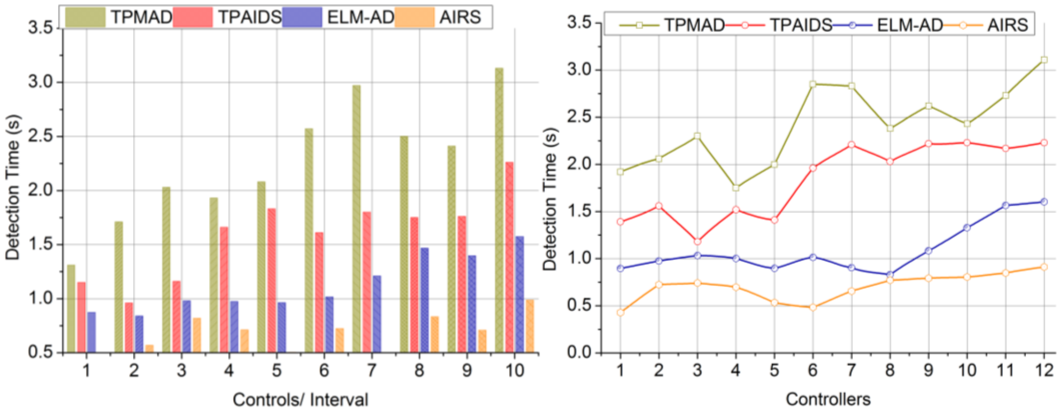

Fig. 11 Detection Time

The detection time decreases differing from control intervals and controllers in this proposed scheme. In this process, the parameters are associated with the control points and entry points from the layers of the device in the industry. Here, the communication is observed between the RTU and central controller and provides better control dissemination among the control point from SCADA. In this category, the field devices are used to forward the input to the supervisory device where the coordinators are used to estimate the better data logs from the control dissemination. In this observation, the intrusion detection time is reduced for the synchronization verification and deploys the data logs. The data are collected from SCADA and processed for better processing in the control dissemination. In this process, the data acquisition is validated to find the failure and it is represented as $\sum_{r_e} r(t\ ) * (v_0 + T_r) + \lambda$. The dissemination data is used to provide better detection of failure and shows less time. The detection time is reduced in this work in which the federated learning is developed for better computation (Fig. 11).

DATA ACQUISITION

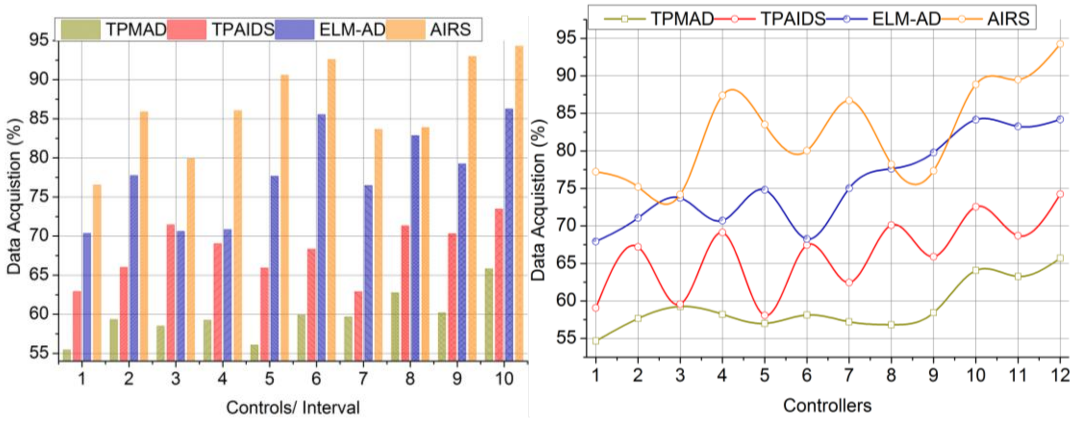

Fig. 12 Data Acquisition

Data acquisition is higher in this proposed work for different controls/ intervals ranging from 1 to 10 and controllers from 1 to 12. Here, the data logs are associated with the SCADA system where the layers are used to provide better training among the entry and control points. The deliberation is used to deploy the parameter of data acquired from the data logs which have the collection of device information. The synchronization verification is examined from the field device and processes the better data acquired from the central controller. Here, the training is developed between the central dissemination and deploys the validation process. In this methodology, the data acquisition is used to transmit the data from the entry point to the next device in SCADA. The validation is observed for the detection of failure from the synchronization verification. From this re-training is estimated for the periodic checking of different entry points in which the verification is examined and it is formulated as $\left(r_e + \frac{t_s+r(t\,)}{q_a+a_0}\right)$. From this data acquisition for the proposed work is found to be high (Fig. 12). The above comparative analysis is briefed in Table 1 and Table 2 for different controls/intervals and controllers.

Table 1 Comparative Analysis Briefing for Controls/ Interval

Metrics	TPMAD	TPAIDS	ELM-AD	AIRS
Detection Ratio	62.48	68.82	72.21	78.479
Control Broadcast (/Controller)	0.702	0.754	0.842	0.9077
Synchronization Failure	0.255	0.201	0.15	0.0827
Detection Time (s)	3.13	2.26	1.574	0.9858
Data Acquisition (%)	65.81	73.45	86.24	94.269

The proposed AIRS increases the detection ratio, control broadcast, and data acquisition by 10.64%, 14.17%, and 9.55% respectively. This scheme reduces the synchronization failure and detection time by 11.93% and 9.59% respectively.

Table 2 Comparative Analysis Briefing for Controllers

Metrics	TPMAD	TPAIDS	ELM-AD	AIRS
Detection Ratio	62.75	67.98	72.55	78.084
Control Broadcast (/Controller)	0.709	0.762	0.827	0.8989
Synchronization Failure	0.392	0.292	0.226	0.1351
Detection Time (s)	3.11	2.23	1.602	0.9122
Data Acquisition (%)	65.69	74.24	84.21	94.265

The proposed AIRS increases the detection ratio, control broadcast, and data acquisition by 10.32%, 13.29%, and 9.78% respectively. This scheme reduces the synchronization failure and detection time by 8.41% and 10.09% respectively.

CONCLUSION

This article introduced an agile intrusion recognition scheme for interoperable SCADA systems. The proposed scheme is designed to mitigate the impact of layered attacks due to cross-functional sequences in SCADA systems. Data acquisition, control log, and operation outputs are periodically verified to identify intrusion entry points to prevent control dissemination. Therefore the field layer and controller synchronization-based verification is performed. This scheme is aided by federated learning to recurrently analyze the verification in inter and intra-operational layers. Based on the recurrent analysis, the changes in logs, data acquired, and control dissemination, the intrusion is identified. The synchronization for control broadcast and data acquisition is verified by the learning process for any new entry and control dissemination points within SCADA. This included the remotely operated devices and the centralized controllers conjoined in the system. From the comparative analysis, the proposed AIR scheme increases the detection ratio, control broadcast, and data acquisition by 10.64%, 14.17%, and 9.55% respectively. This scheme reduces the synchronization failure and detection time by 11.93% and 9.59% respectively.

REFERENCES

[1] Nazir, S., Patel, S., & Patel, D. (2021). Autoencoder based anomaly detection for SCADA networks. *International Journal of Artificial Intelligence and Machine Learning (IJAIML)*, *11*(2), 83-99.

[2] Babu, J. R. (2021). *Design, implementation, and field-testing of distributed intrusion detection system for smart grid SCADA network* (Master's thesis, Iowa State University).

[3] Sen, O., Hassan, T., Ulbig, A., & Henze, M. (2024). Enhancing SCADA Security: Developing a Host-Based Intrusion Detection System to Safeguard Against Cyberattacks. *arXiv preprint arXiv:2402.14599*.

[4] Osman, F. A., Hashem, M. Y., & Eltokhy, M. A. (2022). Secured cloud SCADA system implementation for industrial applications. *Multimedia Tools and Applications*, *81*(7), 9989-10005.

[5] Wai, E., & Lee, C. K. M. (2024). Depth in Defense: A Multi-layered Approach to Cybersecurity for SCADA Systems in Industry 4.0.

[6] Amer, E. Malicious Behavioural Detection in Scada Networks Based on Analyzing Modbus/Tcp Functions Sequences. *Tcp Functions Sequences*.

[7] Gao, J., Gan, L., Buschendorf, F., Zhang, L., Liu, H., Li, P., ... & Lu, T. (2020). Omni SCADA intrusion detection using deep learning algorithms. *IEEE Internet of Things Journal*, *8*(2), 951-961.

[8] Tama, B. A., Lee, S. Y., & Lee, S. (2022). A systematic mapping study and empirical comparison of data-driven intrusion detection techniques in industrial control networks. *Archives of Computational Methods in Engineering*, *29*(7), 5353-5380.

[9] Lin, C. Y., & Nadjm-Tehrani, S. (2023). Protocol study and anomaly detection for server-driven traffic in SCADA networks. *International Journal of Critical Infrastructure Protection*, 100612.

[10] Abou el Kalam, A. (2021). Securing SCADA and critical industrial systems: From needs to security mechanisms. *International Journal of Critical Infrastructure Protection*, *32*, 100394.

[11] Justindhas, Y., & Jeyanthi, P. (2022). Attack detection and prevention in IoT-SCADA networks using NK-classifier. *Soft Computing*, *26*(14), 6811-6823.

[12] Rajesh, L., & Satyanarayana, P. (2022). Evaluation of machine learning algorithms for detection of malicious traffic in scada network. *Journal of Electrical Engineering & Technology*, *17*(2), 913-928.

[13] Balla, A., Habaebi, M. H., Elsheikh, E. A., Islam, M. R., & Suliman, F. M. (2023). The effect of dataset imbalance on the performance of scada intrusion detection systems. *Sensors*, *23*(2), 758.

[14] Mboweni, I. (2024). Hydraulic Data Preprocessing for Anomaly Based Intrusion Detection on SCADA Level of Water Treatment Systems.

[15] Balla, A., Habaebi, M. H., Elsheikh, E. A., Islam, M. R., Suliman, F. E. M., & Mubarak, S. (2024). Enhanced CNN-LSTM Deep Learning for SCADA IDS Featuring Hurst Parameter Self-Similarity. *IEEE Access*.

[16] Zaman, M., Upadhyay, D., & Lung, C. H. (2023). Validation of a Machine Learning-Based IDS Design Framework Using ORNL Datasets for Power System With SCADA. *IEEE Access*, *11*, 118414-118426.

[17] Ahakonye, L. A. C., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2023). Agnostic CH-DT technique for SCADA network high-dimensional data-aware intrusion detection system. *IEEE Internet of Things Journal*.

[18] Ndonda, G. K., & Sadre, R. (2022). Exploiting the Temporal Behavior of State Transitions for Intrusion Detection in ICS/SCADA. *IEEE Access*, *10*, 111171-111187.

[19] Öztürk, T., Turgut, Z., Akgün, G., & Köse, C. (2022). Machine learning-based intrusion detection for SCADA systems in healthcare. *Network Modeling Analysis in Health Informatics and Bioinformatics*, *11*(1), 47.

[20] Ahakonye, L. A. C., Nwakanma, C. I., Lee, J. M., & Kim, D. S. (2023). SCADA intrusion detection scheme exploiting the fusion of modified decision tree and Chi-square feature selection. *Internet of Things*, *21*, 100676.

[21] Al Ghazo, A. T., & Kumar, R. (2023). Critical Attacks Set Identification in Attack Graphs for Computer and SCADA/ICS Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

[22] Nguyen, D. D., Le, M. T., & Cung, T. L. (2022). Improving intrusion detection in SCADA systems using stacking ensemble of tree-based models. *Bulletin of Electrical Engineering and Informatics*, *11*(1), 119-127.

[23] Rabie, O. B. J., Selvarajan, S., Alghazzawi, D., Kumar, A., Hasan, S., & Asghar, M. Z. (2023). A security model for smart grid SCADA systems using stochastic neural network. *IET Generation, Transmission & Distribution*, *17*(20), 4541-4553.

[24] Barsha, N. K., & Hubballi, N. (2024). Anomaly Detection in SCADA Systems: A State Transition Modeling. *IEEE Transactions on Network and Service Management*.

[25] Saheed, Y. K., Abdulganiyu, O. H., & Tchakoucht, T. A. (2023). A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures. *Journal of King Saud University-Computer and Information Sciences*, *35*(5), 101532.

[26] Diaba, S. Y., Anafo, T., Tetteh, L. A., Oyibo, M. A., Alola, A. A., Shafie-Khah, M., & Elmusrati, M. (2023). SCADA securing system using deep learning to prevent cyber infiltration. *Neural Networks*.

[27] Zheng, M., Man, J., Wang, D., Chen, Y., Li, Q., & Liu, Y. (2023). Semi-supervised multivariate time series anomaly detection for wind turbines using generator SCADA data. *Reliability Engineering & System Safety*, *235*, 109235.

[28] Shlomo, A., Kalech, M., & Moskovitch, R. (2021). Temporal pattern-based malicious activity detection in SCADA systems. *Computers & Security*, *102*, 102153.

[29] Oyucu, S., Polat, O., Türkoğlu, M., Polat, H., Aksöz, A., & Ağdaş, M. T. (2023). Ensemble learning framework for DDoS detection in SDN-based SCADA systems. *Sensors*, *24*(1), 155.

[30] Upadhyay, D., Manero, J., Zaman, M., & Sampalli, S. (2021). Intrusion detection in SCADA based power grids: Recursive feature elimination model with majority vote ensemble algorithm. *IEEE Transactions on Network Science and Engineering*, *8*(3), 2559-2574.

[31] Anwar, M., Lundberg, L., & Borg, A. (2022). Improving anomaly detection in SCADA network communication with attribute extension. *Energy Informatics*, *5*(1), 69.

[32] https://www.cse.wustl.edu/~jain/iiot/index.html