# Attack Detection and Mitigation in Industrial IoT : An Optimized Ensemble Approach

**[1]Bibhuti Bhushan Behera, [2]Rajani Kanta Mohanty, [3]Binod Kumar Pattanayak***

1: Department of Computer Science and Engineering. Institute of Technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
2: Department of Computer Science and Engineering-SP, Jain University, Bengaluru, India
3: Department of Computer Science and Engineering. Institute of Technical Education and Research, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India
*: Corresponding Author

**Abstract**: The following four primary steps (a) pre-processing, (b) feature extraction, (c) attack detection, and (d) attack mitigation) are used to create a unique IIOT attack detection and mitigation framework in this research work. The acquired raw data (input) is first treated to a pre-processing step, which includes data normalization activities. subsequently, the features retrieved from the pre-processed data include technical indicators, enhanced higher order statistical features (Skewness, Kurtosis, Variance, and Moments), and improved Mutual Information, Symmetric Uncertainty, Information gain ratio, and Relief based features. A two-stage ensemble of classifiers is used to build the attack detection framework, which comprises the "Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Optimized Deep Belief Network (DBN)". The retrieved features are used to train the Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) that resides within the first stage of the ensemble-classifier. The optimal Deep belief Network (DBN)-in the second layer of ensemble classifier, which is trained with the outcomes obtained from the "Gated recurrent unit (GRU), Recurrent Neural Network (RNN)", and Convolutional neural Network, determines the final detection about the presence/absence of attack in the IIoT network (CNN). The weight functions of the Deep belief Network (DBN) are optimized utilizing the newly projected Migration updated with Supervisor guidance (MUSG) to obtain greater detection accuracy. The proposed hybrid optimization model combines both the Sandpiper Optimization Algorithm (SOA) and the Teamwork Optimization Algorithm (TOA) concepts. The control is handed to the attack mitigation framework whenever an attacker is discovered within the network by the optimised Deep belief Network (DBN). Attack Mitigation Framework: Using the updated BAIT technique, the discovered attacker is mitigated. As a result, the IIoT network is protected to be efficient via comparative analysis.

**Keywords**:IIoT; Security; Attack Detection; Ensemble-of-Classifiers; Attack Mitigation; Weighted BAIT

## 1. Introduction

Industrial Internet of Things (IIoT) refers to industrial-specific augmentation of conventional Internet of Things (IoT) related applications. The IIoT improves an industry's capability to implement more dependability but also performance in its manufacturing applications. The measurement and management powers of an industrialized society have been considerably increased in an intelligent manufacturing system [1, 9, 10, 11, 12, 13, 14], with the integration of various cyber-physical networks as well as current communication technologies [1]. The notion of smart manufacturing has now become critical to grasping the long term goals of the future generation of the industrial revolution, known as Industry 4.0. The industrial sector incorporates a large number of sensors, actuators, and innovative technology. According to a recent assessment, the marketplace for Internet - Connected devices is estimated to reach $75.4 billion by 2025 [1]. Durability, reaction time, and network latency are all critical considerations in today's business world. Data transmission along with decision-making systems should indeed be improved without personal interaction in light of all of these

aspects. In recent times, the Internet of Things (IoT) has emerged as among the most appealing study fields; it has been broadly applied to link an infinite wide range of consumer products in order to give convenience as well as simplicity in customers' everyday lives [1, 4, 5, 15, 16, 17, 18]. The use of the Internet of Things throughout the industrial sector, as per the concept of Industry 4.0, enhances the output, profitability, as well as reliability of manufacturing applications [6]. In a nutshell, the IIoT primarily concerned with the effective application of IoT in industry applications. A four-layered framework may be used to define the IIoT. The middleware layer, which includes cloud services, an application programming interface, and web applications, provides communication between the network and application layers. The application layer in fact represents the topmost layer in the IIoT architecture, and it allows for a number of industrial processes as well as applications, such as smart factories, smart buildings, smart healthcare, smart cars, robots, and so on. The IIoT is a comprehensive infrastructure that can be used by a variety of people as well as businesses. Nevertheless, it introduces a slew of additional security, security, economic, as well as societal factors. Solving these problems necessitates massively scalable technologies. IoT sensor nodes have limited resources, thus security products that can meet the requirements pertaining to minimum possible storage, less possible power along with minimum low cost are required. Standardized communication techniques must be compatible with these solutions. Throughout industry applications, IoT devices create enormous amounts of information, making an IIoT system a tempting target for cyber criminals [7 ,8]. Conventional data processing approaches appear to be not suitable for IoT as well as IIoT applications due to the massive amount of information. As a result, among the most suited computing models for providing embedded intelligence in IoT devices is machine learning (ML). [1, 19, 20, 21, 22, 23].

Appropriate control over the large-scale industrial systems belonging to IIoT is a difficult task. Computing platforms must be able to interpret and analyze large amounts of massive volumes of data quickly and securely [9, 10]. Moreover, the system's capability as well as throughput must've been high in order to ensure low latency and high transmitting data dependability. In regards of dependability and reliability, “machine learning (ML) algorithms” and models have considerably enhanced performance of the industrial sector as a whole. These algorithms offer a lot of promise in terms of addressing security vulnerabilities in IIoT systems [11, 12]. However, they fall short of the requisite level of precision, and their computational cost is larger. Therefore, the optimization algorithms [26, 27, 28, 29, 30] can be deployed into the deep learning model for providing promising solution to attack detection.

The network intrusion detection system (NIDS) is critical in identifying and responding to all World wide web intrusions as a network safety precaution. The IIoT has evolved into a critical component of today's data and information transfer infrastructure, prompting the necessity for global network security [16]. Network intrusion detection systems (NIDS) are frequently used to monitor network communications in order to protect workstation strategies against numerous grid invasions. The network intrusion detection system (NIDS) is critical in identifying and responding to all World wide web intrusions as a network safety precaution. The IIoT has evolved into a critical component of today's data and information transfer infrastructure, prompting the necessity for global network security [16]. Network intrusion detection systems (NIDS) are frequently used to monitor network communications in order to protect workstation strategies against numerous grid invasions.

The major contributions pertaining to this research work are:

- Extracting the most relevant features for precise detection of attacks in the network.
- Introduces a two stage ensemble-of-classifier for accurate network attack detection. it comprises the “Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Optimized Deep Belief Network (DBN)”.
- To optimize the weight of DBN with new Migration updated with Supervisor guidance (MUSG) for enhancing the convergence speed of the solutions as well as to enhance the detection accuracy
- Introduces a new Migration updated with Supervisor guidance (MUSG) by conceptually blending the standard Sandpiper Optimization Algorithm (SOA) and the Teamwork Optimization Algorithm (TOA), respectively.
- Introduces an improved BAIT framework for attack mitigation in IIoT.

The rest of this paper is arranged as follows. Section 2 tells about the recent works in IIoT attack detection. Section 3 depicts about IIOT attack detection and mitigation: an overview, Section 4 covers the pre-processing via data normalization. Section 5, Section 6 and Section 7 describe about multi-feature extraction, attack detection framework and attack mitigation framework, respectively. The acquired outcomes are discussed is Section 8. This paper is concluded in Section 9.

## 2.Litearture review

### 2.1 Related Works

The IIoT connects a wide range of sensors, equipment, industrial applications, databases, services, and workers. Smarter cities, agriculture, and e-healthcare are just a few of the ways the IIoT is enhancing our lives. Although the IIoT and consumer IoT share some traits, the two networks use distinct cybersecurity measures. Threats as well as cyberattacks on IoT infrastructure are continuously increasing in lockstep with the expanding use of IoT infrastructure across all domains. “Attacks and anomalies that might cause an IoT system failure include Denial of Service, Data Type Probing, Malicious Control, Malicious Operation, Scan, Spying, and Wrong Setup”. A complete set of security solutions should be used to secure IIoT infrastructure so that operations, service dependability, and profitability are not harmed. A practical and simple, yet safe solution that IIoT device makers and their customers can simply and broadly embrace is more successful than a "super solution" that fails to acquire momentum.

In 2021, Nayak et al. [1] have projected a deep learning based IIoT attack detection model for detecting intended attacks. The attack detection was carried out using the “Generative Adversarial Network-Classifier (GAN-C)” method that was formulated by blending the GAN as well as Support Vector Machine (SVM), respectively. The projected model has been identified to be the optimal approach for centralized attack detection in IIoT.

In 2021, Moustafa et al. [2] have projected a new “Distributed Anomaly Detection (DAD)” system for exploring attacks in the IIoT edge networks. The Gaussian Mixture-based Correntropy has been utilized within the projected model to effectively monitor as well as recognize the IIoT zero-day attacks. The projected model has been validated with “NSL-KDD and UNSW-NB15 datasets”.

In 2021, Tsogbaatar et al. [3] have projected “deep ensemble learning framework framework for IoT anomaly detection and prediction”- DeL-IoT using SDN. The projected model has been made with the “anomaly detection, intelligent flow management, and device status forecasting”. The handy features has been extracted with

the deep and stacked autoencoders of DeL-IoT. The projected model has exhibited higher reliability in detecting the attacks.

In 2020, Aoudi et al. [4] have developed generalizes PASAD for intrusion detection in IIOT network. The routing protocol has been administered effectively and accurately while maintaining the trustworthiness and security of data routing. For circumstances with a significant proportion of malevolent nodes, a scenario of node collaboration has been provided. By establishing a dynamic Bayesian equilibrium between the attacker and the detection node, anomalous intrusions have been prevented. The experimental findings suggest that playing the mechanisms cooperative game improves the success rate of detection of anomalous monitoring nodes and reduces the amount of faked notifications considerably.

In 2020, Wang et al. [5] projected a cluster-based routing technique. The routing protocol has been administered effectively and accurately while maintaining the trustworthiness and security of data routing. For circumstances with a significant proportion of malevolent nodes, a scenario of node collaboration has been provided. By establishing a “ynamic Bayesian equilibrium” between the attacker and the detection node, anomalous intrusions have been prevented. The experimental findings suggest that playing the mechanisms cooperative game improves the “event detection” “success rate” of anomalous monitoring nodes and reduces the amount of faked notifications considerably.

Li et al. [6] suggested a deep learning based strategy using a “multi-convolutional neural network (multi-CNN)” fusion technique for IIoT intrusion detection in 2019. The feature information has so far been separated into four parts based on the connection, and then the “one-dimensional feature” data has been turned into a “grayscale graph”. CNN has been incorporated into the intrusion detection problems utilising the stream information visualization approach, and indeed the optimum of the four outcomes emerges. Relating to the NSL-KDD dataset, the findings further suggest that the multi-CNN fusion model has been particularly suited for delivering a classification technique with extreme reliability as well as minimal complication.

Sun et al. [7]focued on malware detection research on IIoT in 2021, proposing a framework for a categorised behaviour “graph-based intelligent detection framework” for malware intrusions, which could not only minimize the massive price of graph matching thereby achieving high accuracy in case of malware detection. Investigations with the “malware families Delf, Obfuscated, Small, and Zlob”, each of which has 880 samples, demonstrate that the greatest accuracy TPR may exceed 99.9%.

Althobaiti et al. [8]predicted a unique cognitive computing-based IDS approach for industrialized CPS cybersecurity in 2021. " Data collecting, preprocessing, feature selection, classification, and parameter optimization" are some of the processes included with the suggested model. Preprocessing was used for the suggested model to eliminate any data noise. The algorithm subsequently selects an optimal collection of characteristics using a feature selection strategy based on binary bacterial foraging optimization (BBFO). In addition, in the industrial CPS setting, the “gated recurrent unit (GRU)” model has been used to detect intrusions. Finally, the hyperparameter optimization of the GRU model was evaluated using the “Nesterov-accelerated Adaptive Moment Estimation (NADAM) optimizer”, which increased the detection rate.

Li et al. [6] suggested a deep learning strategy Using a “multi-convolutional neural network (multi-CNN)” fusion technique for IIoT intrusion detection in 2019. The feature information has so far been split into four parts based on the connection, and then the “one-dimensional feature” data has been turned into a “grayscale graph”. CNN has been

incorporated into the intrusion detection problems utilizing the stream information visualization approach, and indeed the optimum of the four outcomes emerges. On the NSL-KDD dataset, the findings further suggest that the multi-CNN fusion model has been particularly suited for delivering a classification technique with extreme reliability as well as minimal complication.

Sun et al. [7]focued on malware detection research on IIoT in 2021, proposing a framework for a categorised behaviour "graph-based intelligent detection framework" for malware intrusions, which could not only minimize the massive price of graph matching but also achieve high malware detection accuracy. Investigations with the "malware families Delf, Obfuscated, Small, and Zlob", each of which has 880 samples, demonstrate that the greatest accuracy TPR may exceed 99.9%.

Althobaiti et al. [8]predicted a unique cognitive computing-based IDS approach for industrialized CPS cybersecurity in 2021. " Data collecting, preprocessing, feature selection, classification, and parameter optimization" are some of the processes included with the suggested model. Preprocessing was used for the suggested model to eliminate any data noise. The algorithm subsequently selects an optimal collection of characteristics using a feature selection strategy based on binary bacterial foraging optimization (BBFO). In addition, in the industrial CPS setting, the "gated recurrent unit (GRU)" model has been used to detect intrusions. Finally, the "hyperparameter optimization" of the GRU model was evaluated using the "Nesterov-accelerated Adaptive Moment Estimation (NADAM) optimizer", which increased the detection rate.

## 2.2 Review

For numerous years, both the scientific community as well as business have been researching DDoS detection and mitigation. Numerous research have been done to provide methods to deal with this problem in a broad sense, according to the linked publications. Another collection of papers focused on giving solutions for both high-volume and low-volume DDoS assaults. Additionally, notwithstanding the Computer Emergency Response Team's (CERT) several suggestions for combating DDoS assaults as well as rules defined through Request for Comments (RFC), these attacks continue to happen on a regular basis.

The incompetency of preventing and analyzing DDoS attempts is inextricably connected to continual setup errors and lost time owing to the lack of technologies that track the characteristics of the network without constant human interaction, according to a research conducted decades previously. As a result, researchers are turning to autonomous solutions that can operate (identify as well as mitigate) traffic depending on its behaviour and features. In this regard, the use of "artificial intelligence (AI)" techniques, particularly ML, has been noted for providing great adaptability inside the categorization process, hence increasing the identification of hostile traffic [18].

**Table.1** Advantages and Drawbacks of existing models

| **Author [ Citation]** | **Methodology** | **Features** | **Challenges** |
|---|---|---|---|
| Latif et al. [9] | random neural network (RaNN) | ✓ precision, recall, and F1 score are higher<br>✓ higher detection accuracy | accuracy and feasibility need to be improved |

| | | | |
|---|---|---|---|
| Li et al. [10] | SRKD | ✓ efficient storage and communication<br>✓ presents a way to defend against node replication attacks<br>✓ potentially discover and revoke copies in a cost-effective manner | overhead is higher<br>Storage and transmission costs are slightly greater. |
| Li et al. [11] | "bidirectional long and short-term memory network with multi-feature layer (BMLSTM)" | ✓ It has a decreased rate of false positives and false negatives. | The training time is higher |
| Antonopoulos and Verikoukis [12] | "Network-Coding-aware Statistical Approach" | ✓ Detects rogue users on the network with ease. | The channel errors are higher |
| Ho et al. [13] | "Sequential Probability Ratio Test (SPRT) with the probabilistic inspection" | ✓ Increase the detection rate of code-reuse packets in both small and big groups.<br>✓ Through simulation and analysis, achieves resiliency in detection with minimal overhead. | The average false positive rate can be improved. |
| Qureshi et al. [14] | "Routing Protocol for Low Power and Lossy Networks (RPL)" | ✓ "HELLO-Flood attack, Version number attack, Sinkhole attack, and Black hole attack" detection capabilities<br>✓ With reduced data loss and delay, the packet delivery rate is higher. | higher False positive rate |
| Naeem et al. [15] | MD-IIOT | ✓ The detection accuracy and redictive time are both improved. | Lower classification accuracy, Precision, and Recall rate |
| Xiong et al. [16] | Rabin cryptosystem | ✓ Provides the desired levels of security and functionality.<br>✓ It strikes a careful balance between security and efficiency, and it works better in real-world circumstances.<br>✓ sturdy enough to withstand all known threats while still achieving more optimal qualities | Higher computational complexity and cost |

## 3.IioT attack detectoin and Mitigation: an Overview

### 3.1 Architectural Description

Cyber terrorist and hacker attacks are increasingly targeting vital smart infrastructure that relies on smart IIoT devices. Although a slew of new studies on the IIoT have been published, the detection accuracy has yet to meet the expected level of satisfaction. As a result, the main goal of this study is to offer a novel two-level classification model for detecting attacks. Furthermore, the attacks that have been found have been neutralised. As a result, the network becomes extremely secure. The newly introduced innovative attack detection as well as mitigation model is introduced with four major phases: **(a) pre-processing, (b) feature extraction, (c) Two-stage Attack detection and (d) attack mitigation**. The steps followed in the projected model are depicted below:

Let the collected input data be $D^{in}$

**Step 1:** **Pre-processing:** Initially, the collected raw data (input $D^{in}$) is subjected to pre-processing phase, wherein the **data normalization** operations are accomplished. The pre-processed data is denoted as $D^{pre}$.

**Step 2:** **Feature extraction**: The features inclusive of Higher Order statistical features $f^{H-stat}$- improved Skewness, kurtosis, variance, moment; Second Order Technical Indicators $f^{SOTI}$-ATR, KST OSCILLATOR, MASS INDEX (MI) and TR; Symmetrical Uncertainity $f^{SU}$; Improved mutual information $f^{IMI}$; Information gain ratio $f^{IIG}$; RelifF features $f^{RF}$ are extracted from the pre-processed data $D^{pre}$.

**Step 3:** The acquired features are fused together as: $f^{H-stat} + f^{SOTI} + f^{IMI} + f^{IIG} + f^{SU} + f^{RF} = F$

**Step 4:** Two-stage Attack detection: The attack detection framework is constructed with two-stage classifiers.

- Stage 1: In the first stage, the classifiers like Gated recurrent unit (GRU), Recurrent Neural Network (RNN), Convolutional neural Network (CNN) is modeled. All these classifiers run in parallel and are independent to each other. These classifiers are trained with the extracted features $F$**.** The outcome acquired from GRU is $out^{GRU}$; RNN is $out^{RNN}$; CNN is $out^{CNN}$.
- Stage 2: In the second stage, the optimized Deep belief Network (DBN) is models, which makes the final decisions regarding the presence/absence of attack in the network. Since, the DBN is the ultimate decision maker, its weight, its weights functions are fine-tuned with the newly projected hybrid optimization model- Migration updated with Supervisor guidance (MUSG). Thereby, the detection accuracy increases. The projected **hybrid optimization model** hybrids the concept of both the standard **Sandpiper Optimization Algorithm (SOA)** [33] and **Teamwork Optimization Algorithm (TOA)** [34], respectively.

**Step 5:** In case, if an attackers is identified within the network by the optimized Deep belief Network (DBN), then the control is transferred to the attack mitigation framework.

**Step 6:** Attack mitigation Framework: the detected attacker is mitigated using the improved BAIT approach. As a consequence, the IIoT network becomes secured.

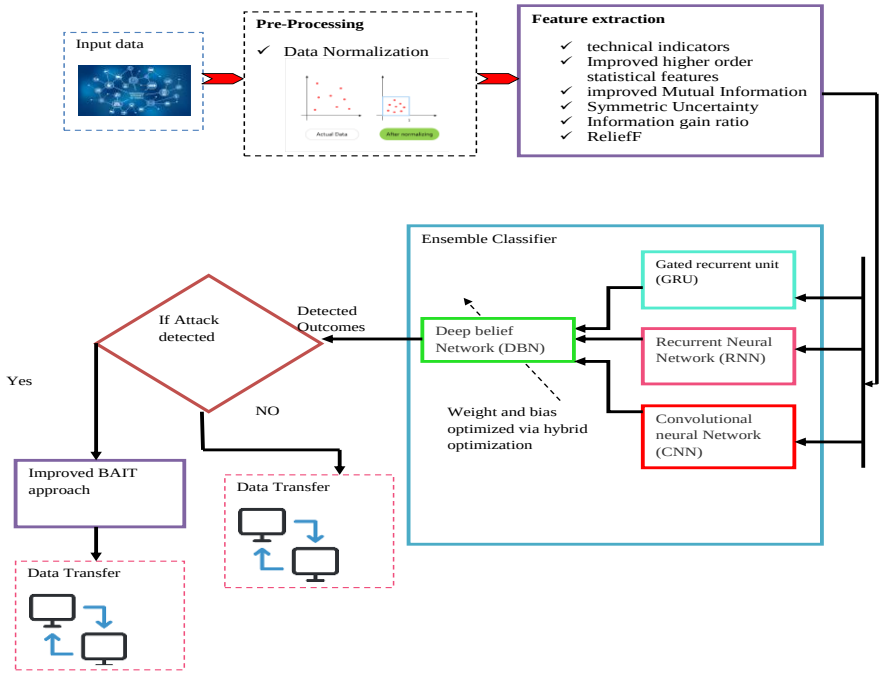



**Fig.1** Architecture of the projected model

## 4.Pre-Processing via data normalization

The collected raw input $D^{in}$ enters into the pre-processing stage.The pre-processing is the basis step that is being utilized for transforming the raw data into an useful as well as understandable format. Therefore, the collected raw data $D^{inp}$ is subjected to pre-processing stage. Among the available pre-processing techniques, the data normalization is the most efficient model. the pre-processing stage is shown in Fig.2.

### 4.1 Data Normalization

The data normalization [44] scales down the data into a regularized range that could be easier to be processed further. In this research work, $D^{inp}$ is normalized within the range 0 to 1. The normalized data acquired as outcome is denoted as $D^{pre}$-preprocessed data. Subsequently, from $D^{pre}$, the most reliable multi-features are extracted.

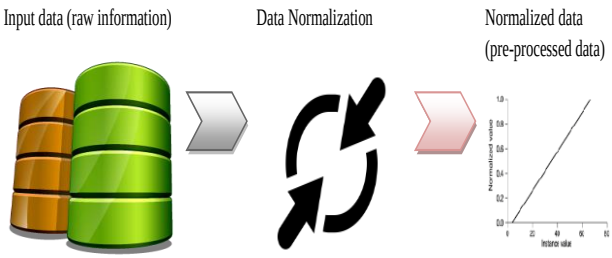



**Fig.2** Pre-processing

## 5.Multi-Feature Extraction

Feature extraction is important in detection models because it helps machine learning models improve their detection capacity. Feature extraction, in general, is the process of converting input into a numerical feature. Advantages like as accuracy improvement, over-fitting relief, speeding up the training process, and improved data visualization may be realized by extracting the most trustworthy features. As a result, the most trustworthy multi-features are extracted in this study. As a result of this, the system's

detecting performance improves. The extracted features are shown diagrammatically in Fig.3.

The following multi-features are extracted in this research work:

1. Higher Order statistical features- improved Skewness, kurtosis, variance, moment
2. Second Order Technical Indicators-ATR, KST OSCILLATOR, MASS INDEX (MI) and TR
3. Symmetrical Uncertainity
4. Improved mutual information
5. Information gain ratio
6. ReliefF features

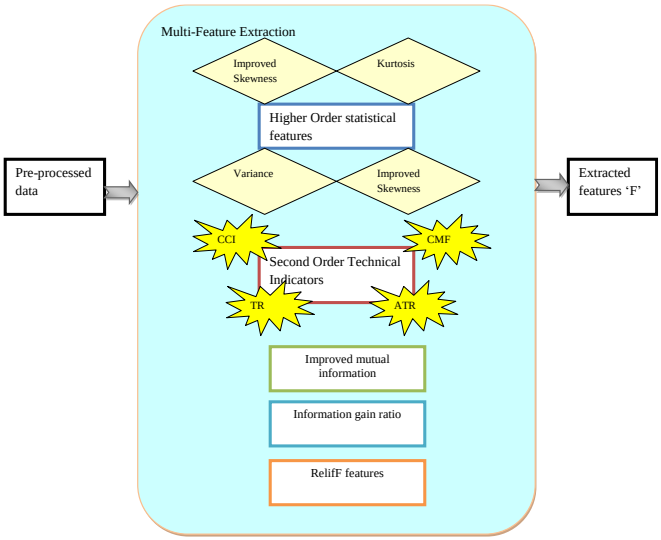



**Fig.3** Feature Extraction

## 5.1 Higher Order Statistical Features

The higher order statistical features like improved Skewness, kurtosis, variance, moment are extracted from $D^{pre}$.

(a) **Improved Skewness:** The skewness of a real-valued random number's probability distribution around its mean is a measure of its asymmetry. The skewness might be positive, negative, or indeterminate, which is one of its biggest advantages. The skewness, on the other hand, fails owing to its unpredictability. As a result, a better skewness metric has been included in this study. The newly proposed skewness metric can be expressed mathematically as Eq (1).

$$\mu_3 = \frac{\sum_i^N \left(D_i^{pre} - \overline{D_i^{pre}}\right)^3}{(N-1)*\sigma^3} \quad (1)$$

Here, $\overline{D_i^{pre}}$ is the weighted harmonic mean.

$$\overline{D_i^{pre}} = \frac{\sum W_i}{\left(\sum W_i / D_i^{pre}\right)} \quad (2)$$

Here, $W_i$ denotes the weight of data points.

(b) Kurtosis: It tells you how "tailed" the probability distribution of the real-valued random variable is. The kurtosis may be calculated mathematically using Eq. (3).

$$\mu_3 = \frac{\sum_i^N \left(D_i^{pre} - \overline{D_i^{pre}}\right)^4}{\sigma^4} \quad (3)$$

(c) Variance: It is the expectation of a random variable's squared deviation from its popular mean or sample mean. The variance may be calculated mathematically using Eq.(4).

$$\text{var}(D^{pre}) = \sum_i^N \left(D_i^{pre} - \overline{D_i^{pre}}\right)^2 \quad (4)$$

(d) Moment: It contains information on the graph function's shape. The first moment is the centre of mass, and the second moment is the rotational inertia, assuming the moment function represents a mass. If the function is a probability distribution, the anticipated value is the first moment, whereas kurtosis is the second central moment, third standardized moment, and fourth standardized moment. The variance may be calculated mathematically using Eq (5).

$$\sigma \equiv \left[\sum_i^N \left(D_i^{pre} - \overline{D_i^{pre}}\right)^2\right]^{1/2} \quad (5)$$

## 5.2 Second Order Technical Indicators

The second order technical indicators like ATR, KST OSCILLATOR, MASS INDEX (MI) and TR are extracted from $D^{pre}$.

(e) ATR: It is a technical indicator that decomposes the complete range of network information to determine the volatility of the information flow. The ATR finds a succession of real values in order to improve network security. ATR may be expressed mathematically as Eq (6).

$$ATR = \frac{1}{T}\sum_{i=1}^{T} R_i \quad (6)$$

Here, $R_i$ points to the particular true range at the time period $T$.

(f) KST OSCILLATOR: The KST oscillator is a momentum indicator that makes interpreting network rate of change simple.

(g) MASS INDEX (MI): The mass index attempts to estimate the range of high and low values for information flow during a certain time period.

(h) TR: It gives information on the information flow's volatility.

## 5.3 Symmetrical Uncertainity

The symmetrical uncertainty (SU) is used to evaluate a non-linear connection between variables and between variables and class labels. Entropy and information gain are used to formulate SU. SU may be stated mathematically as Eq (7).

$$SU(A,B) = 2\left[\frac{IG(A|B)}{H(A)+H(B)}\right] \quad (7)$$

The information gain $IG(A|B)$ can be expressed as per Eq. (8).

$$IG(A|B) = H(A) - H(A|B) \quad (8)$$

The entropy $H(A)$ and $H(A|B)$ is mathematically expressed as per Eq. (9).

$$H(A) = \sum_{i=1} p(A_i).Log(P(A_i)) \quad (9)$$

$$H(A|B) = -\sum_j p(B_j)\sum_i p(A_i|B_j)\log\left[p(A_i|B_j)\right] \quad (10)$$

In which, $P(A_i)$ and $p(A_i|B_j)$ denotes the probability of $A$ and probability of $A$ given $B's$ value, respectively.

The extracted symmetrical uncertainty (SU) feature is pointed as $f^{SU}$

### 5.4 Improved mutual information

MI may be used for nonlinear transformation as well as high-order statistics extraction. Furthermore, improved mutual information (IMI) is taken into account to improve the relevance between two variables. The IMI may be expressed mathematically as Eq. (11).

$$IMI = \frac{1}{\min(D_1, D_2) - 1} \sum_{i=1}^{D_1} \sum_{j=1}^{D_2} \frac{p(A,B) - p(A).p(B)}{p(i).p(j)} \quad (11)$$

Here, $D_1, D_2$ is the size of the $A, B$

### 5.5 Information gain ratio

The ratio of information gain to intrinsic information is known as the information gain-to-intrinsic-information ratio. It's the entropy decrease that comes from splitting set using characteristics $A$, and finding the optimal candidate that produce the highest value.

$$f^{IG}(R, A) = H(R) - H(R|A) \quad (12)$$

Here, $H(R|A)$ is the entropy of $R$ given $A$. Moreover, $R$ is a random value.

### 5.6 ReliefF features

It computes the "proxy statistic for each feature that can be used to estimate feature 'quality' or 'relevance' to the target concept (i.e. predicting endpoint value)". These feature statistics are necessarily referred to as feature weights $(W|A)$-weight of the attribute '$A$'. The features inclusive of Higher Order statistical features $f^{H-stat}$- improved Skewness, kurtosis, variance, moment; Second Order Technical Indicators $f^{SOTI}$-ATR, KST OSCILLATOR, MASS INDEX (MI) and TR; Symmetrical Uncertainity $f^{SU}$; Improved mutual information $f^{IMI}$; Information gain ratio $f^{IIG}$; ReliefF features $f^{RF}$ are extracted from the pre-processed data $D^{pre}$.

The acquired features are fused together as: $f^{H-stat} + f^{SOTI} + f^{IMI} + f^{IIG} + f^{SU} + f^{RF} = F$

## 6.Attack detection Framework

Stage 1: In the first stage, the classifiers like "Gated recurrent unit (GRU), Recurrent Neural Network (RNN), Convolutional neural Network (CNN)" is modeled. All these classifiers run in parallel and are independent to each other. These classifiers are trained with the extracted features $F$ **.** The outcome acquired from GRU is $out^{GRU}$; RNN is $out^{RNN}$; CNN is $out^{CNN}$.

Stage 2: In the second stage, the optimized Deep belief Network (DBN) is models, which makes the final decisions regarding the presence/absence of attack in the network. Since, the DBN is the ultimate decision maker, its weight, its weights functions are fine-tuned with the newly projected hybrid optimization model. Thereby, the detection accuracy increases. The projected **hybrid optimization model-** Migration updated with Supervisor guidance (MUSG) hybrids the concept of both the standard **Sandpiper Optimization Algorithm (SOA)** [33] and **Teamwork Optimization Algorithm (TOA)** [34], respectively.

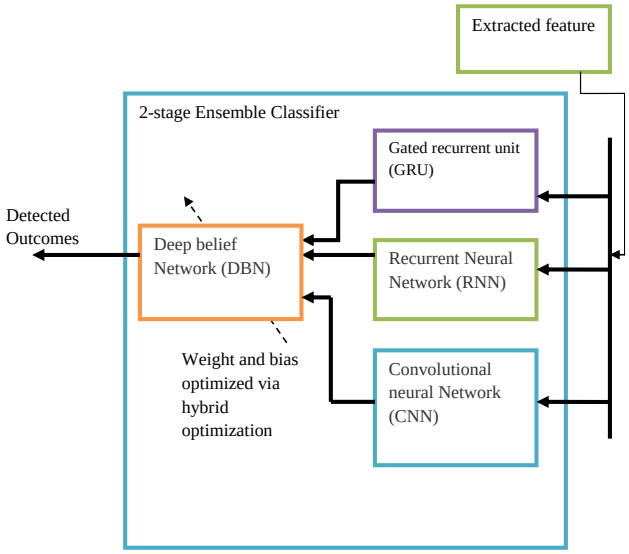



**Fig.4** Attack detection

### 6.1 GRU

The Gated Recurrent Unit, or GRU [45], is indeed an RNN structure that is comparable to LSTM components. Instead of the input, output, and forget gates of the LSTM, the GRU has a reset gate as well as an update gate. The reset gate specifies how well the new input is combined with the previously used memory, while the update gate decides how much of the existing memory is retained. We can go back to our normal RNN model by setting the reset to all 1s and the update gate to all 0s. The hidden state $h^t$ for the GRU may be determined as follows:

$$Z^t = \alpha(F_t.U^z + h^{t-1}.W^z) \quad (13)$$

$$r^t = \alpha(F_t.U^r + h^{t-1}.W^r) \quad (14)$$

$$\bar{h}^t = \tanh\left[F_t.U^h + \left(h^{t-1} * r^t\right)W^h\right] \quad (15)$$

$$h^t = (1 - Z^t)h^{t-1} + Z^t\bar{h}^t \quad (16)$$

Here, $r, g$ point to the rest gate and update gate, respectively. In addition, $U$ and $W$ denote the recurrent connection at the previous hidden and current hidden layers and, weight matrix connecting the inputs to the current hidden layer, respectively.
The outcome acquired from GRU is $out^{GRU}$

### 6.2 RNN

Recurrent neural networks [46] are the type of artificial neural network wherein the nodes' connections create a directed graph which represents a temporal sequence. In recurrent neural networks, in contrast to feed-forward neural networks, process sequences tend to use their internal state memory. Recurrent neural networks are necessarily valuable as well as adaptable to data analysis and other applications because of their dynamic activity. Mathematically, RNN can be given as per Eq. (17) and Eq. (18), respectively.

$$Hid^{time} = fun^H(Wg^{IH}.F^{time} + Wg^{HH}.h^{time-1}) \quad (17)$$

$$out^{RNN} = fun_0(Wg^{HO}.h^{time}) \quad (18)$$

Here, $F$ is the input feature vector and $out^{RNN}$ is the outcome from RNN. In addition, $Wg^{IH}$, $Wg^{HH}$ and $Wg^{HO}$ points to the weight matrices between input and hidden; hidden and hidden; and hidden and output layer, respectively.

### 6.3 CNN

CNN [47 ][48] is a newly developed classifier with three layers: fully connected, pooling, and convolution (multiple hidden layers). To calculate distinct feature maps, the convolution layer encompasses numerous convolution kernels. The entire feature map is obtained by combining multiple distinct kernels. The feature values at position $(x,y)$referred to as $M^{q}_{x,y,z}$ are computed using Eq. (19) in the $q^{th}$ layer corresponding to the $n^{th}$ feature map. The bias term and the weight vector are represented as $bi^{q}_{n}$ and $wi^{q}_{n}$, respectively, in the $n^{th}$ filter of the $q^{th}$ layer.

The input patches linked in the $n^{th}$ layer at position $(x,y)$are represented by $g^{q}_{x,y}$. The activation function introduces non-linearity to CNN, which consequently assists in the prediction of non-linear characteristics in multi-layer networks. The non linear activation function is indicated by the term $act(\bullet)$, and the activation value $\left(act^{q}_{x,y,z}\right)$for the convolutional features $M^{q}_{x,y,z}$ is determined using Eq.(20). The shift-variance is obtained at the pooling layer by lowering the resolution of the feature maps. The pooling function is represented as $pool(\ )$for each feature map, and the local neighbourhood around the location $(x,y)$for each feature map $\left(act^{q}_{x,y,z}\right)$is represented as $Rn_{x,y}$ Eq (21).

$$M^{q}_{x,y,z} = wi^{q^T}_{n} g^{q}_{x,y} + bi^{q}_{n} \quad (19)$$

$$act^{q}_{x,y,z} = act\left(M^{q}_{x,y,z}\right) \quad (20)$$

$$yi^{q}_{x,y,x} = pool\left(act^{q}_{t,v,n}\right), \forall (t,v) \in Rn_{xy} \quad (21)$$

The loss of CNN, denoted by $Loss$ is calculated using Eq. (22), in which all of the CNN parameters $(\theta)$ corresponding to $R$ desirable input-output relations are represented as $\left\{\left(xi^{(n)}, yi^{(n)}\right); n \in [1,\cdots,R]\right\}$; $xi^{(n)}$, $yi^{(n)}$ and $oi^{(n)}$ ,represent the nth input data, the associated target labels, and the CNN output, respectively.

$$Loss = \frac{1}{R}\sum_{n=1}^{R} q\left(\theta; yi^{(n)}, oi^{(n)}\right) \quad (22)$$

The optimized Deep Belief Network (DBN) is used in the second step to make final judgements about the existence or absence of an attack in the network. The DBN's weight and weight functions are fine-tuned with the newly projected hybrid optimization model because it is the final decision maker. As a result, the detection accuracy improves. Migration updated with Supervisor guidance (MUSG) is a proposed hybrid optimization model that combines the concepts of the basic Sandpiper Optimization Algorithm (SOA) [33] and the Teamwork Optimization Algorithm (TOA) [34].

### 6.4 Optimized DBN

In 2018, Smolensky created DBN [49] [50], which consists of many layers. The input layer and the output layer, respectively, are implanted with visible and buried neurons. The symmetry and exclusivity of the connections between these hidden as well as visible neurons are more pronounced. Furthermore, there is no link between the hidden and input neurons. The resultants of the Boltzmann networks are designated $O$. The probability of output $(O)$ is indicated by $pf(\tau)$and is theoretically described by the equation Eq. (23). The probabilityistic of output $(O)$is stated to be a sigmoid-shaped

function defined in Eq. (24). The pseudo-temperature $(T)$ plays an important role here, and when it falls to 0 (zero), the stochastic model becomes deterministic. A stochastic neural network containing stochastic neurons is known as a Boltzmann machine, sometimes known as a recurrent network.

$$pf(\tau)=\frac{1}{1+e^{\frac{-\tau}{T}}} \quad (23)$$

$$O=\begin{Bmatrix}0 & with\ probability\ of\ 1-pf(\tau)\\ 1 & with\ probability\ of \quad pf(\tau)\end{Bmatrix} \quad (24)$$

$$\lim_{T\to 0^+} pf(\tau)=\lim_{T\to 0^+}\frac{1}{1+e^{\frac{-\tau}{T}}}=\begin{cases}0 & for \quad \tau<0\\ \frac{1}{2} & for \quad \tau=0\\ 1 & for \quad \tau>0\end{cases} \quad (25)$$

Eq.(26) calculates the Boltzmann machine's associated energy across the specified neuron state. The weight between the hidden and visible neurons $u$ and $v$ is denoted by $wt_{u,v}$, whereas the biases are denoted by $\psi_x$. The energy difference is computed over the unit state $c_u$ using Eq (27). The gradient descent technique is used to compute the minimum feasible energy throughout the training phase.

$$En(c)=-\sum_{u<v} c_u\, c_v\, wt_{u,v}-\sum_u \psi_u\, c_u \quad (26)$$

$$\Delta En(c_u)=\sum_v c_v wt_{u,v}+\psi_u \quad (27)$$

In the process of calculation of the difference in energy, the RBM remains independent of the visible as well as hidden neurons. Such a calculation is carried out using Eq. (27), Eq. (28) and Eq. (29), respectively.

$V_u$ and $L_v$ represent the binary states of visible unit $u$ and hidden unit $v$, respectively. The visible and hidden neuron biases are denoted by A and B, respectively, while the weight between the neurons is denoted by $wt_{u,v}$. The training of RBM takes done with the aid of unsupervised learning.

$$En(\vec{V},\vec{L})=\sum_{(u,v)} wt_{u,v}V_u L_v-\sum_u u_u A_v-\sum_v V_v B_v \quad (27)$$

$$\Delta En(V_u,\vec{L})=\sum_v wt_{uv}L_v+A_u \quad (28)$$

$$\Delta En(\vec{V},L_v)=\sum_u wt_{uv}V_u+B_v \quad (29)$$

$Z$ denotes the training samples, and RBM is taught to improve the assigned probabilities between these training sets $Z$. In addition, the weight assignment $Wg_q$ is assigned to yield the maximum probability, which is mathematically stated as Eq. (30). Every pair of visible and hidden neurons in RBM is assigned a probability according to Eq. (31). The partition function represents the total sum of the potential energy levels of all the neurons, and it is denoted by $X$, as Per Eq. (32) is the mathematical expression for $X$.

$$Wt_q=\max_{Wg}\prod_{\vec{V}\in Z} pf(\vec{u}) \quad (30)$$

$$pf(\vec{V},\vec{L})=\frac{1}{X}\sum_{\vec{V}} e^{-En(\vec{V},\vec{L})} \quad (31)$$

$$X=\sum_{\vec{V},\vec{L}} e^{-En(\vec{V},\vec{L})} \quad (32)$$

The loss function of SBN is given in MSE. This loss function is the difference between the actual as well as the predicted value. Our major objective is to less the MSE of DBN, as it

makes the final detection regarding the presence/ absence of attacks in IIoT. The objective function of this research work (minimization of detection error) is given in Eq. (33).

$$Obj - Min(MSE) \quad (33)$$

In order to achieve this

Contrastive Divergence (CD) learning is more effective in initializing the observable states of neurons.

**6.5 Hybrid Optimization Model- MUSG**

To achieve a higher detection accuracy, weight function of the Deep belief Network (DBN) is optimized using the newly projected hybrid optimization model (Migration updated with Supervisor guidance (MUSG)) . The projected hybrid optimization model hybrids the concept of both the standard Sandpiper Optimization Algorithm (SOA) [33] and Teamwork Optimization Algorithm (TOA) [34], respectively. The movement and attacking behavior of sandpipers are the major inspirations for SOA. The major purpose of creating the TOA is to imitate cooperation behaviors among team members in order to achieve a common goal. For ease of use in solving optimization issues, the TOA and SOA have been mathematically represented. The solutions can reach the fastest convergence time by optimizing the TOA and SOA models, and they can obtain global solutions without becoming stuck in local optima. The solution fed as input to MUDG is the weight of DBN that has been optimized to achieve the objective defined in Eq. (33). The solution fed as input to MUSC model is shown in Fig.5.

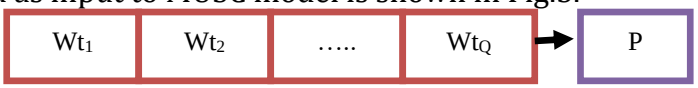



**Fig.5** Solution Encoding

The steps followed in the MUSG model is furnished in the upcoming section

**Step 1:** Initialize the $N$ search agent's population $P$ .

**Step 2:** Initialize the parameters $itr, Max^{itr}$ .]Here, $itr, Max^{itr}$ points to the current iteration and the maximal iteration, respectively.

**Step 3:** While $itr < Max^{itr}$ do

**Step 4:** Using Eq. (33), calculate the search agent's fitness.

**Step 5:** Move on to the SOA model's migration behaviour. This is where we can make a difference. In reality, during the Migration (exploration) phase, three distinct processes take place: collision avoidance, movement in the direction of the best neighbour, and staying near to the best search agent.

✓ A new variable $A$, referred to as the search agent's movement behaviour, is introduced with the goal of avoiding collisions among the search agents.

$$C = A * P(X) \quad (34)$$

Here, $C$ is the search agent's non- colliding position and $P$ is the current position of the search agent. In addition, $A$ can be computed as per Eq. (35), wherein $f_c$ is a variable that has been introduced to control the frequency of the employing $A$ .

$$A = f_c - \left\{ X, \left[ \frac{f_c}{Max^{itr}} \right] \right\} \quad (35)$$

✓ Our contribution was included during the progress along the best neighbor's direction stage. Once the search agents have avoided colliding, they are shifted in the direction of the best neighbour. A novel mathematical model with global optimal solution $G_{best}(X)$ has been presented in this study effort to make this movement more

exact and to prevent solutions from being stranded in local optima. Eq. (36) shows the newly generated mathematical equation.

$$M = B*\left[G_{best}(X) - P(X)\right] \quad (36)$$

This $G_{best}(X)$ is calculated using the TOA model's Supervisor guiding stage. Team members are updated in the first step depending on the supervisor's instructions. At this point, the supervisor shares her or his knowledge and reports with the rest of the team, guiding them toward the goal. Equations (37)–(38) are used to replicate this stage of the update in the TOA.

$$G_{best}(X) = P(X) + r*Z - I*P(X) \quad (37)$$

The index guidance $I$ is mathematically given as per Eq. (38).

$$I = round(1+r)$$

In addition, $Z$ , $P(X)$ and $M$ points to the supervisor, current position of the search agent, position of the search agent, respectively.

✓ Remain near to the best search agent: Finally, according to Eq. (39) of the SOA model, the search agent's location is updated in relation to the best search agent.

$$\vec{D} = \left|\vec{C} + \vec{M}\right| \quad (39)$$

In addition, $\vec{D}$ is the distance in between the best search agent and the current search agent.

**Step 6:** Advance to the Attacking (exploitation) phase: During the attacking phase, the search agents spiral through the air in a spiral motion. Eqs. (40) and (41), respectively, may be used to simulate this spiral movement in the three axes $X, Y, Z$ .

$$x' = r*Cos(K) \quad (40)$$

$$y' = r*Sin(K) \quad (41)$$

$$z' = r*K \quad (42)$$

Here, random variable is $K$

**Step 7:** The updated position of the search agent is modelled as per Eq. (43).

$$P(X) = (D*x'*y'*z') + P_{best}(x) \quad (43)$$

**Step 8:** Return $P(X)$

**Step 9:** Terminate

## 7. Attack Mitigation Framework

### 7.1 Improved BAIT

The aim is to detect the malicious node on path.

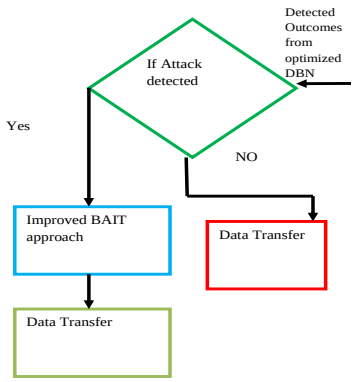



**Fig.6** Attack mitigation

The steps followed in the projected Improved BAIT model are described as follows.

1. First, the source node creates BAIT-RREQ and picks a neighbouring node N at random from its one-hop neighbourhood nodes, cooperating with it and using its address as the destination address in the BAIT-RREQ packet.
2. If the source node broadcast the fake RREQ to the path, then malicious node exist in the network. In addition, if any node sends the RREP for this BAIT other than N, then malicious node exist in the network. Then, the black hole list lists the nodes as malicious and ignored for future transmission. The diagrammatic representation of improved BAIT is shown in Fig.7-Fig.9, respectively.

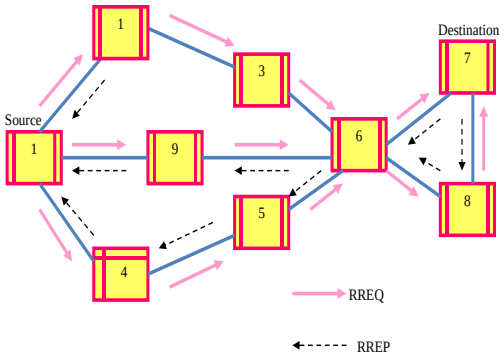



**Fig.7** Network with Node

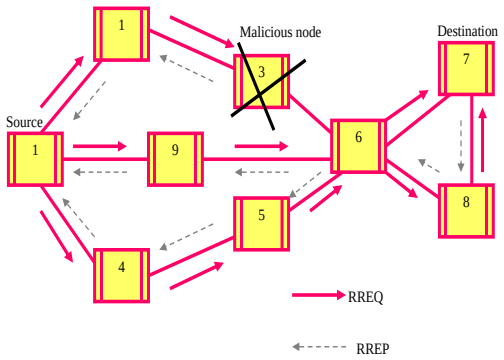



**Fig.8** Bait-RREQ and RREP

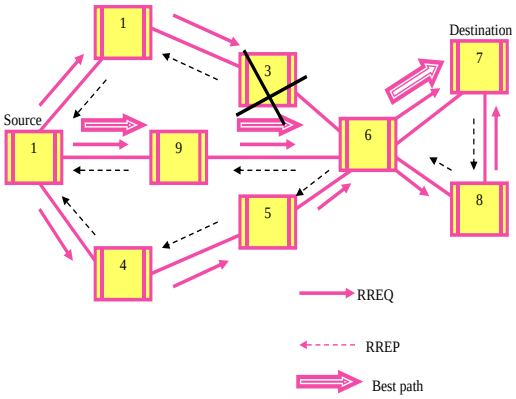



**Fig.9** Best path identifiaction

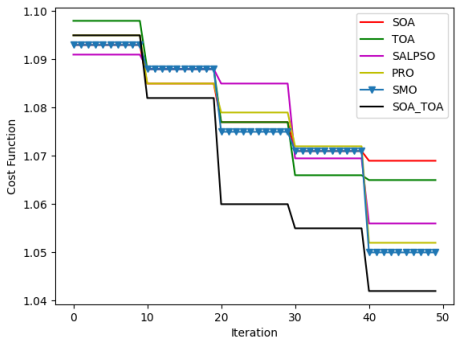
## 8.Result and Discussion

### 8.1 Experimental Setup

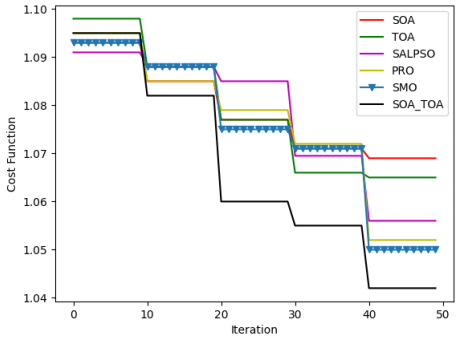
The projected IIoT attack detection and mitigation model is implemented in PYTHON. The data for evaluation has been collected from dataset1[51] and dataset2 [52]. The projected model is validated over the existing models like SOA, TOA, SALPSO, PRO and SMO, respectively. The evaluation has been made in terms of "Accuracy, Sensitivity, Specificity, Precision, Negative Predictive Value (NPV), F1-Score and Mathews correlation coefficient (MCC), False positive rate (FPR), False negative rate (FNR), and False Discovery Rate (FDR)". The evaluation has been made by varying the learning percentage from 60, 70, 80 and 90, respectively.

### 8.2 Convergence Analysis

The convergence analysis is done to validate the efficiency of the projected MUSG (SOA_TOA) over the existing optimization models in solving the defined objective function. This evaluation has been made by varying the count of iterations from 0, 10, 20, 30, 40 and 50, respectively. The outcomes acquired are shown in Fig.10. To show the efficiency of the projected model, a comparative evaluation has been made between MUSG over the existing models like SOA, TOA, SALPSO, PRO and SMO, respectively. This evaluation has been made for both dataset1 and dataset2, respectively. Since, the designed objective function is minimization of error in DBN, [which produces the final detection outcome (presence/ absence of attackers)], the approach with the least cost function is said to be the optimal one in solving the optimization problem. As per the acquired outcomes the projected model shows lowest cost unction for every variation in the learning percentage, and so it's suggested to be the best approach in solving the convergence issues. As a consequence, the projected model is said to be the significant optimization approach for detecting the presence/ absence of attackers. Under dataset1, the projected model has recorded the least cost function as 1.042 at 50$^{th}$ iteration count. Moreover, at this 50$^{th}$ iteration, the existing model has recorded the cost functions as SOA=1.075, TOA=1.07, SALPSO=1.06, PRO=1.05, SMO=1.052. On the other hand, the projected model ha recorded the least cost function under dataset 2 also. This enhancement has been recorded under every variation in the learning percentage. The major reason behind this enhancement owes towards the conceptual blend of two highly convergent optimization techniques. Thu, the projected model has been said to be highly applicable for attack detection in IIoT.




(a)




(b)

**Fig.10** Convergence analysis of the projected model for (a) Dataset 1 and (b) Dataset 2

## 8.3 Performance Analysis

The performance of the projected model (MUSG+2stsge EC) is compared over the existing models like SOA+2stsge EC, SALPSO+2stsge EC, PRO+2stsge EC and SMO+2stsge EC, respectively. The evaluation has been made in terms of "Accuracy, Sensitivity, Specificity, Precision, Negative Predictive Value (NPV), F1-Score and Mathews correlation coefficient (MCC), False positive rate (FPR), False negative rate (FNR), and False Discovery Rate (FDR)". The outcomes acquired are shown in Fig.11-Fig,.19. To manifest that the projected model is better than the existing models, the positive measures like accuracy, sensitivity, specificity as well as precision needs to be as high as possible. On the other hand, the negative measure or error measures like FPR and FNR needs to be maintained at its lowest range. This evaluation has been made by varying the learning percentage. On observing the acquired outcome, the projected model has recorded the highest accuracy under both dataset1 and dataset2, respectively. Under dataset1, the projected model has recorded the highest accuracy as 97.5% at 60$^{th}$ learning percentage, 97.8% at 70$^{th}$ learning percentage at 80$^{th}$ learning percentage, 98% and 99% at 90$^{th}$ learning percentage. Under dataset 2, the projected model has recorded the higher accuracy above 98% under every variation in the learning percentage. In addition, the projected model has recorded the highest specificity, sensitivity as well as precision. Moreover, the error measures like FPR and FNR of the projected model is lower than the existing models for every variation in the learning percentage. Therefore, the projected model is said to be less prone to errors. In addition, the other measures like F1-score, NPV and MCC of the projected model is found to be higher with the projected model, which is the most favorable score. All these improvements are due to the extraction of the most relevant features, as well as hybrid optimization for enhancing the detection accuracy. Thus, the projected model is said to be successful in attack detection.

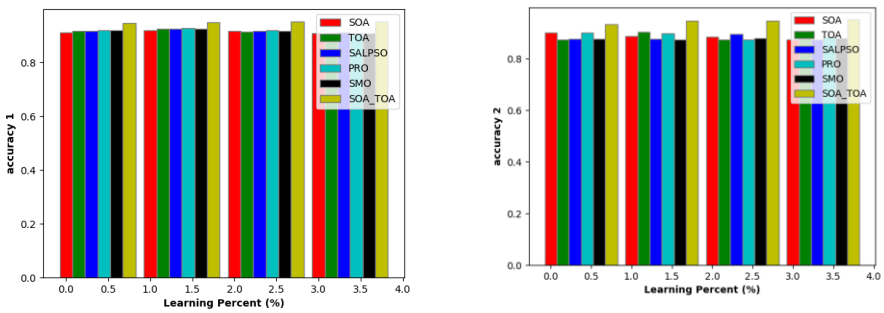


**Fig.11** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of Accuracy

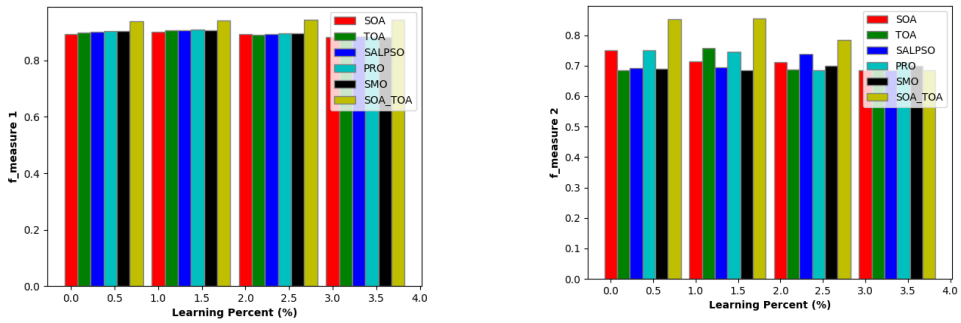



**Fig.12** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of F-measure

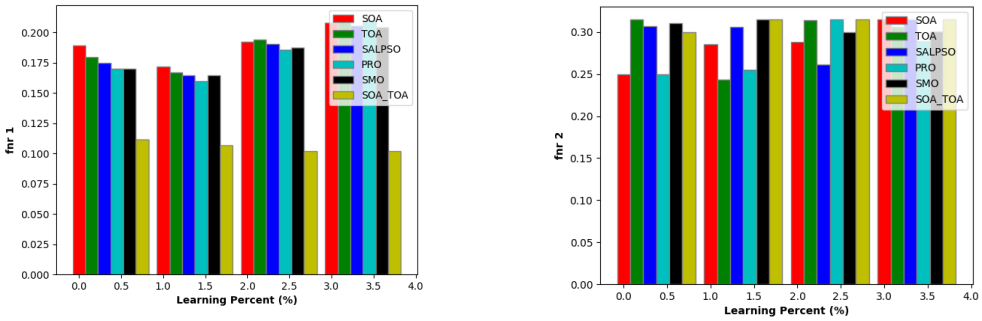



**Fig.13** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of FNR

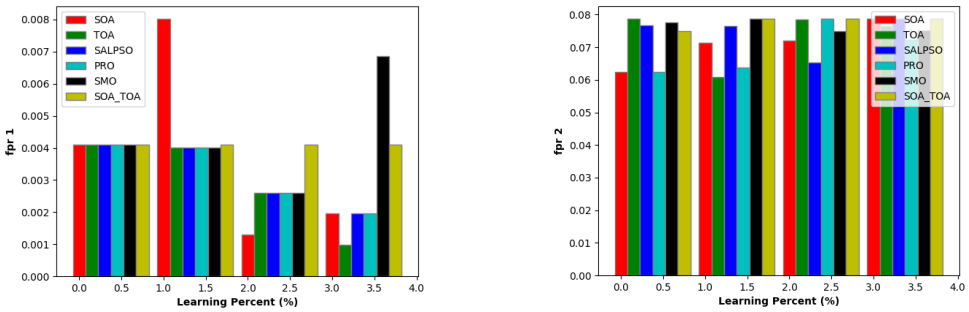



**Fig.14** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of FPR

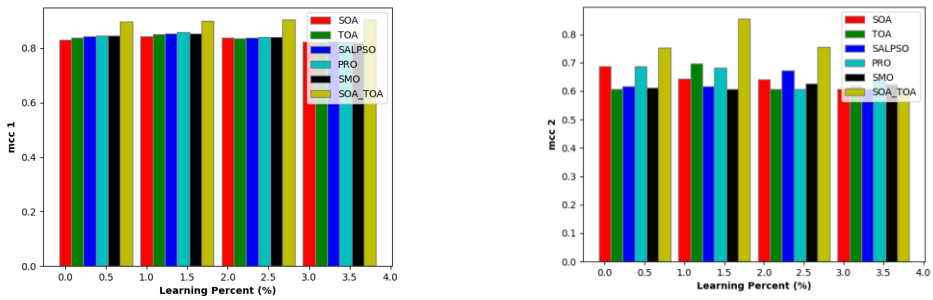



**Fig.15** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of MCC

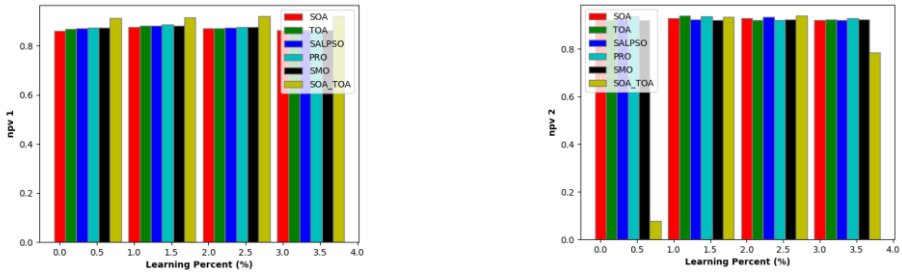



**Fig.16** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of NPV

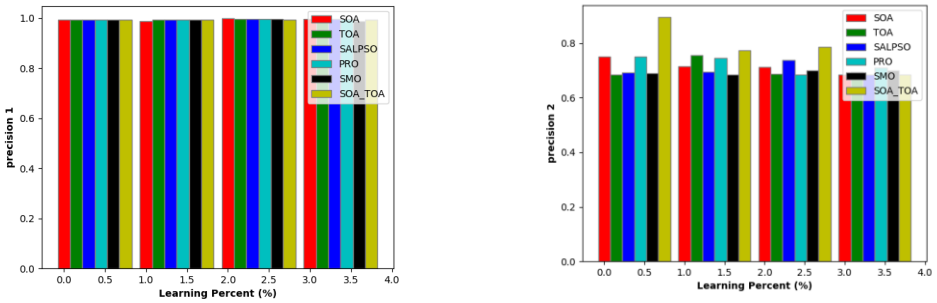



**Fig.17** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of precision

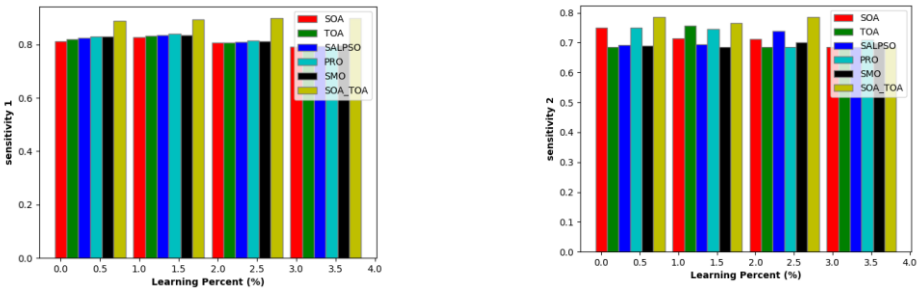



**Fig.18** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of Sensitivity

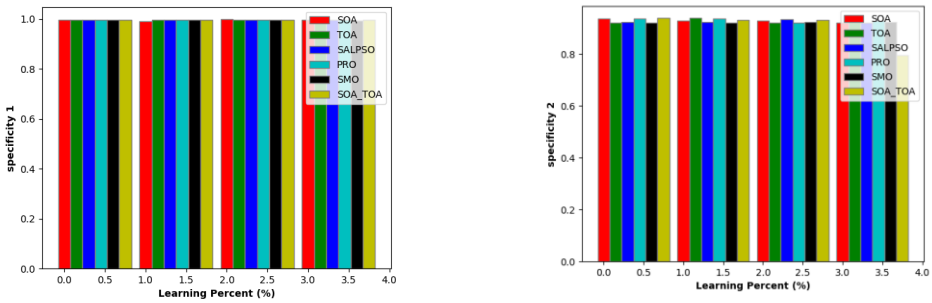



**Fig.19** Performance Analysis of Projected model for (a) Dataset 1 and (b) Dataset 2 in terms of Specificty

## 8.4 Classifier Performance Analysis

The performance of the MUSG+2stsge EC (two-stage ensemble classifier) is compared over the existing models like NB, Bi-LSTM, Bi-GRU, CNN, NN, GAN-SVM and EC+ CMUSMA, respectively. On observing the outcomes, the projected model has recorded the best performance over the existing ones. The accuracy of the projected model under

dataset1 is 3.06%, 40.10%, 1.71%, 6.53%, 21.08%, 1.99% and 1.75% better than the existing models like LSTM, Bi-GRU, CNN [6], NN, GAN-SVM [1] and EC+ CMUSMA, respectively. In addition, the accuracy of the projected model under dataset 2 is 24%, 29%, 15%, 19%, 24%, 12% and 5% improved over the existing models like LSTM, Bi-GRU, CNN, NN, GAN-SVM[1] and EC+ CMUSMA, respectively. The projected model has recorded the highly sensitivity as 99.5% and precision as 99.4% under dataset 1. In addition, the projected model has recorded the least error values. The FNR of the projected model under dataset 1 is 42.47%, 88.89%, 30.85%, 27.17%, 12.24%, 52.4% and 57.76% improved over the existing models like LSTM, Bi-GRU, CNN, NN, GAN-SVM and EC+ CMUSMA, respectively. In addition, the FPR of the projected model under dataset 2 is 55%, 61%, 35%, 34%, 55%, 23% and 23% improved over the existing models like LSTM, Bi-GRU, CNN, NN, GAN-SVM and EC+ CMUSMA, respectively. Thus, the projected model is said to be applicable for attack detection in IIoT.

**Table 2** Table Styles

| Measures | NB | Bi-LSTM | Bi-GRU | CNN[6] | NN | GAN-SVM [1] | EC+ CMUSMA [51] | PRO+2stsgeEC |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.919865 | 0.56835 | 0.93266 | 0.93468 | 0.748822 | 0.929966 | 0.932304 | 0.948889 |
| SENSITIVITY | 0.814353 | 0.0390016 | 0.845554 | 0.853354 | 0.878315 | 0.929966 | 0.951164 | 0.893204 |
| SPECIFICITY | 1 | 0.970379 | 0.998815 | 0.996445 | 0.650474 | 0.929966 | 0.938507 | 0.995902 |
| PRECISION | 1 | 0.5 | 0.998158 | 0.994545 | 0.656177 | 0.929966 | 0.924971 | 0.994595 |
| F-MEASURE | 0.897678 | 0.0723589 | 0.915541 | 0.918556 | 0.751167 | 0.929966 | 0.95316 | 0.941176 |
| MCC | 0.84482 | 0.0257588 | 0.868375 | 0.871625 | 0.53028 | 0.859933 | 0.865636 | 0.900271 |
| NPV | 0.876428 | 0.570732 | 0.894904 | 0.899465 | 0.875598 | 0.929966 | 0.873193 | 0.916981 |
| FPR | 0 | 0.0296209 | 0.00118483 | 0.0035545 | 0.349526 | 0.0700337 | 0.0488363 | 0.0040984 |
| FNR | 0.185647 | 0.960998 | 0.154446 | 0.146646 | 0.121685 | 0.0700337 | 0.0676956 | 0.106796 |

**Table 3** Table Styles

| Measures | NB | Bi-LSTM | Bi-GRU | CNN[6] | NN | GAN-SVM [1] | EC+ CMUSMA [51] | PRO+2stsgeEC |
|---|---|---|---|---|---|---|---|---|
| ACCURACY | 0.721388 | 0.675222 | 0.807264 | 0.809524 | 0.719774 | 0.835997 | 0.894616 | 0.94558 |
| SENSITIVITY | 0.303471 | 0.188055 | 0.51816 | 0.52381 | 0.299435 | 0.589992 | 0.897768 | 0.765814 |
| SPECIFICITY | 0.825868 | 0.797014 | 0.87954 | 0.880952 | 0.824859 | 0.897498 | 0.896342 | 0.932541 |
| PRECISION | 0.303471 | 0.188055 | 0.51816 | 0.52381 | 0.299435 | 0.589992 | 0.82845 | 0.77451 |
| F-MEASURE | 0.303471 | 0.188055 | 0.51816 | 0.52381 | 0.299435 | 0.589992 | 0.886459 | 0.85412 |
| MCC | 0.129338 | -0.0149314 | 0.3977 | 0.404762 | 0.124294 | 0.48749 | 0.791228 | 0.85412 |
| NPV | 0.825868 | 0.797014 | 0.87954 | 0.880952 | 0.824859 | 0.897498 | 0.911623 | 0.93412 |
| FPR | 0.174132 | 0.202986 | 0.12046 | 0.119048 | 0.175141 | 0.102502 | 0.102232 | 0.0786844 |
| FNR | 0.696529 | 0.811945 | 0.48184 | 0.47619 | 0.700565 | 0.410008 | 0.105384 | 0.314738 |

### 8.5 Statistical Performance

Since, the projected optimization model is stochastic in nature, it has been executed 5 times, and the best values acquired in terms of " mean, median, standard deviation, best and worst" measures have been noted for dataset 1 and dataset 2, respectively. The mean of the projected model for dataset 1 is 41.3%, 40.26%, 38.9%, 38.7% and 38.06%

improved over like SOA, TOA, SALPSO, PRO and SMO, respectively. In addition, under dataset 2, the mean of the projected model is 50.4%, 52.12%, 52.6%, 54.01% and 49.1% improved over like SOA, TOA, SALPSO, PRO and SMO, respectively. All these improvement are owing towards the utilization of the 2 stage ensemble classifier for attack detection.

**Table 4** Statistical Analysis of the projected model for dataset1

| Measures | best | worst | mean | median | std |
|---|---|---|---|---|---|
| SOA+2stsge EC | 0.0811111 | 0.0911111 | 0.0862037 | 0.0862963 | 0.00398363 |
| TOA+2stsge EC | 0.0766667 | 0.0922222 | 0.0846296 | 0.0848148 | 0.00550906 |
| SALPSO+2stsge EC | 0.0755556 | 0.09 | 0.0828704 | 0.082963 | 0.00513451 |
| PRO+2stsge EC | 0.0755556 | 0.0922222 | 0.0825 | 0.0811111 | 0.00610479 |
| SMO+2stsge EC | 0.0733333 | 0.0916667 | 0.0816204 | 0.0807407 | 0.00656216 |
| PRO+2stsge EC | 0.0488889 | 0.0533333 | 0.0505556 | 0.05 | 0.00184257 |

**Table 5** Statistical Analysis of the projected model for dataset2

| Measures | best | worst | mean | median | std |
|---|---|---|---|---|---|
| SOA+2stsge EC | 0.0999167 | 0.125895 | 0.113822 | 0.114738 | 0.00923705 |
| TOA+2stsge EC | 0.0972523 | 0.125895 | 0.117818 | 0.124063 | 0.0119445 |
| SALPSO+2stsge EC | 0.10458 | 0.125895 | 0.118984 | 0.122731 | 0.00841707 |
| PRO+2stsge EC | 0.1199 | 0.125895 | 0.122565 | 0.122231 | 0.00256907 |
| SMO+2stsge EC | 0.0999167 | 0.125895 | 0.110824 | 0.108743 | 0.010583 |
| PRO+2stsge EC | 0.04879 | 0.067459 | 0.0563648 | 0.054605 | 0.0068323 |

## 9. Conclusion

In this research work, a novel IIOT attack detection and mitigation framework have been designed by following four major phases (a) pre-processing, (b) feature extraction, (c) attack detection and (d) attack mitigation. The steps followed in the projected model are depicted below: Pre-processing: Initially, the collected raw data (input) have been subjected to pre-processing phase, wherein the data normalization operations have been accomplished. Feature extraction: The features inclusive of technical indicators, Improved higher order statistical features (Skewness, Kurtosis, Variance and Moments) and improved Mutual Information, Symmetric Uncertainty, Information gain ratio and ReliefF based features have been extracted from the segmented data. A two-stage ensemble of classifiers is used to build the attack detection framework, which comprises the Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Optimized Deep Belief Network (DBN). The retrieved features are used to train the Gated Recurrent Unit (GRU), Recurrent Neural Network (RNN), and Convolutional Neural Network (CNN) that resides within the first stage of the ensemble-classifier. The optimal Deep belief Network (DBN)-in the second layer of ensemble classifier, which is trained with the outcomes obtained from the Gated recurrent unit (GRU), Recurrent Neural Network (RNN), and Convolutional neural Network, determines the final detection about the presence/absence of attack in the IIoT network (CNN). The weight functions of the Deep belief Network (DBN) are optimized utilizing the newly projected Migration updated with Supervisor guidance (MUSG) to obtain greater detection accuracy. The proposed hybrid optimization model combines both the Sandpiper Optimization Algorithm (SOA) and the Teamwork Optimization Algorithm (TOA) concepts. The control is handed to the attack mitigation framework whenever an attacker is discovered within the network by the optimized

Deep belief Network (DBN). Attack Mitigation Framework: Using the updated BAIT technique, the discovered attacker is mitigated. As a result, the IIoT network is protected to be efficient via comparative analysis. In case, if an attackers is identified within the network by the optimized Deep belief Network (DBN), then the control have been transferred to the attack mitigation framework. Attack mitigation Framework: the detected attacker have been mitigated using the improved BAIT approach. As a consequence, the IIoT network have been secured.

**References**

[1] Sharmistha Nayak,Nurzaman Ahmed,Sudip Misra,"Deep Learning-Based Reliable Routing Attack Detection Mechanism for Industrial Internet of Things",Ad Hoc Networks, 2021

[2] Nour Moustafa,Marwa Keshk,Monica Whitty,"DAD: A Distributed Anomaly Detection system using ensemble one-class statistical learning in edge networks",Future Generation Computer Systems, 2021

[3] Enkhtur Tsogbaatar,Monowar H. Bhuyan,Youki Kadobayashi,"DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT",Internet of Things, 2021

[4] Wissam Aoudi,Magnus Almgren,"A scalable specification-agnostic multi-sensor anomaly detection system for IIoT environments",International Journal of Critical Infrastructure Protection, 2020

[5] Chao Wang, "IoT anomaly detection method in intelligent manufacturing industry based on trusted evaluation",The International Journal of Advanced Manufacturing Technology, 2020

[6] Yanmiao Li,Yingying Xu,Lizhen Cui,"Robust detection for network intrusion of industrial IoT based on multi-CNN fusion", Measurement, 2019

[7] Yi Sun,Ali Kashif Bashir,Fei Xiao,"Effective malware detection scheme based on classified behavior graph in IIoT",Ad Hoc Networks, 2021

[8] Maha M. Althobaiti,K. Pradeep Mohan Kumar,Romany F. Mansour,"An intelligent cognitive computing based intrusion detection for industrial cyber-physical systems",Measurement, 2021

[9] S. Latif, Z. Zou, Z. Idrees and J. Ahmad, "A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network," in IEEE Access, vol. 8, pp. 89337-89350, 2020

[10] L. Li et al., "A Secure Random Key Distribution Scheme Against Node Replication Attacks in Industrial Wireless Sensor Systems," in IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 2091-2101, March 2020

[11] X. Li, M. Xu, P. Vijayakumar, N. Kumar and X. Liu, "Detection of Low-Frequency and Multi-Stage Attacks in Industrial Internet of Things," in IEEE Transactions on Vehicular Technology, vol. 69, no. 8, pp. 8820-8831, Aug. 2020

[12] A. Antonopoulos and C. Verikoukis, "Misbehavior detection in the Internet of Things: A network-coding-aware statistical approach," Industrial Informatics (INDIN), Poitiers,pp. 1024-1027, 2017

[13] J. Ho, "Efficient and Robust Detection of Code-Reuse Attacks Through Probabilistic Packet Inspection in Industrial IoT Devices," in IEEE Access, vol. 6, pp. 54343-54354, 2018

[14] Kashif Naseer QureshiShahid Saeed RanaAwais AhmedGwanggil Jeon,"A novel and secure attacks detection framework for smart cities industrial internet of things",Sustainable Cities and Society,2020

[15] Hamad NaeemFarhan UllahMuhammad Rashid NaeemShehzad KhalidSaqib Saeed,"Malware detection in industrial internet of things based on hybrid image visualization and deep learning model",Ad Hoc Networks,2020
[16] Mengxia ShuaiLing XiongChanghui WangNenghai Yu,"A secure authentication scheme with forward secrecy for industrial internet of things using Rabin cryptosystem",Computer Communication,2020
[17] N. Koroniotis, N. Moustafa, F. Schiliro, P. Gauravaram and H. Janicke, "The SAir-IIoT Cyber Testbed as a Service: A Novel Cybertwins Architecture in IIoT-Based Smart Airports," in *IEEE Transactions on Intelligent Transportation Systems*. doi: 10.1109/TITS.2021.3106378
[18] A. S. M. S. Hosen, P. K. Sharma, I. -H. Ra and G. H. Cho, "SPTM-EC: A Security and Privacy-Preserving Task Management in Edge Computing for IIoT," in *IEEE Transactions on Industrial Informatics*. doi: 10.1109/TII.2021.3123260
[19] M. Al-Hawawreh and E. Sitnikova, "Developing a Security Testbed for Industrial Internet of Things," in *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5558-5573, 1 April1, 2021. doi: 10.1109/JIOT.2020.3032093
[20] F. Rezaeibagha, Y. Mu, X. Huang, W. Yang and K. Huang, "Fully Secure Lightweight Certificateless Signature Scheme for IIoT," in *IEEE Access*, vol. 7, pp. 144433-144443, 2019.
doi: 10.1109/ACCESS.2019.2944631
[21] Q. Tian *et al*., "New Security Mechanisms of High-Reliability IoT Communication Based on Radio Frequency Fingerprint," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7980-7987, Oct. 2019. doi: 10.1109/JIOT.2019.2913627
[22] J. M. Mcginthy and A. J. Michaels, "Secure Industrial Internet of Things Critical Infrastructure Node Design," in *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8021-8037, Oct. 2019. doi: 10.1109/JIOT.2019.2903242
[23] F. Khan, M. A. Jan, A. u. Rehman, S. Mastorakis, M. Alazab and P. Watters, "A Secured and Intelligent Communication Scheme for IIoT-enabled Pervasive Edge Computing," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 5128-5137, July 2021. doi: 10.1109/TII.2020.3037872
[24] P. Goswami, A. Mukherjee, M. Maiti, S. K. S. Tyagi and L. Yang, "A Neural Network Based Optimal Resource Allocation Method for Secure IIoT Network," in *IEEE Internet of Things Journal*. doi: 10.1109/JIOT.2021.3084636
[25] J. Wan, J. Li, M. Imran, D. Li and Fazal-e-Amin, "A Blockchain-Based Solution for Enhancing Security and Privacy in Smart Factory," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3652-3660, June 2019. doi: 10.1109/TII.2019.2894573
[26] K. Tange, M. De Donno, X. Fafoutis and N. Dragoni, "A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities," in *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2489-2520, Fourthquarter 2020.
doi: 10.1109/COMST.2020.3011208

[27] G. Raja, S. Anbalagan, G. Vijayaraghavan, P. Dhanasekaran, Y. D. Al-Otaibi and A. K. Bashir, "Energy-Efficient End-to-End Security for Software-Defined Vehicular Networks," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5730-5737, Aug. 2021.
doi: 10.1109/TII.2020.3012166
[28] T. Wu, C. Chen, K. Wang and J. M. Wu, "Security Analysis and Enhancement of a Certificateless Searchable Public Key Encryption Scheme for IIoT Environments," in *IEEE Access*, vol. 7, pp. 49232-49239, 2019.
doi: 10.1109/ACCESS.2019.2909040
[29] W. Yang, S. Wang, X. Huang and Y. Mu, "On the Security of an Efficient and Robust Certificateless Signature Scheme for IIoT Environments," in *IEEE Access*, vol. 7, pp. 91074-91079, 2019.
doi: 10.1109/ACCESS.2019.2927597
[30] T. Gebremichael *et al*., "Security and Privacy in the Industrial Internet of Things: Current Standards and Future Challenges," in *IEEE Access*, vol. 8, pp. 152351-152366, 2020.
doi: 10.1109/ACCESS.2020.3016937
[31] G. Rathee, S. Garg, G. Kaddoum and B. J. Choi, "Decision-Making Model for Securing IoT Devices in Smart Industries," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4270-4278, June 2021.
doi: 10.1109/TII.2020.3005252
[32] M. Martinezdelucena and A. A. Frohlich, "Security and Effectiveness Analysis of the Gateway Integrity Checking Protocol," in *IEEE Transactions on Dependable and Secure Computing*.
doi: 10.1109/TDSC.2021.3058057
[33] F. Kohnhäuser, D. Meier, F. Patzer and S. Finster, "On the Security of IIoT Deployments: An Investigation of Secure Provisioning Solutions for OPC UA," in *IEEE Access*, vol. 9, pp. 99299-99311, 2021.
doi: 10.1109/ACCESS.2021.3096062
[34] M. Zolanvari, Z. Yang, K. Khan, R. Jain and N. Meskin, "TRUST XAI: Model-Agnostic Explanations for AI With a Case Study on IIoT Security," in *IEEE Internet of Things Journal*.
doi: 10.1109/JIOT.2021.3122019
[35] H. Xiong, Y. Wu, C. Jin and S. Kumari, "Efficient and Privacy-Preserving Authentication Protocol for Heterogeneous Systems in IIoT," in *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11713-11724, Dec. 2020.
doi: 10.1109/JIOT.2020.2999510
[36] M. M. Hassan, S. Huda, S. Sharmeen, J. Abawajy and G. Fortino, "An Adaptive Trust Boundary Protection for IIoT Networks Using Deep-Learning Feature-Extraction-Based Semisupervised Model," in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2860-2870, April 2021.
doi: 10.1109/TII.2020.3015026
[37] K. A. Abuhasel and M. A. Khan, "A Secure Industrial Internet of Things (IIoT) Framework for Resource Management in Smart Manufacturing," in *IEEE Access*, vol. 8, pp. 117354-117364, 2020.
doi: 10.1109/ACCESS.2020.3004711
[38] M. I. Aziz Zahed, I. Ahmad, D. Habibi and Q. V. Phung, "Content Caching in Industrial IoT: Security and Energy Considerations," in *IEEE Internet of Things Journal*,

vol. 7, no. 1, pp. 491-504, Jan. 2020. doi: 10.1109/JIOT.2019.2948147
[39] W. Zhang, H. Zhang, L. Fang, Z. Liu and C. Ge, "A Secure Revocable Fine-grained Access Control and Data Sharing Scheme for SCADA in IIoT Systems," in *IEEE Internet of Things Journal*. doi: 10.1109/JIOT.2021.3091760
[40] J. Xiong *et al.*, "A Personalized Privacy Protection Framework for Mobile Crowdsensing in IIoT," in *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231-4241, June 2020. doi: 10.1109/TII.2019.2948068
[41] A. Moradbeikie, K. Jamshidi, A. Bohlooli, J. Garcia and X. Masip-Bruin, "An IIoT Based ICS to Improve Safety Through Fast and Accurate Hazard Detection and Differentiation," in *IEEE Access*, vol. 8, pp. 206942-206957, 2020. doi: 10.1109/ACCESS.2020.3037093
[42] Amandeep Kaur, Sushma Jain, Shivani Goel, "Sandpiper optimization algorithm: a novel approach for solving real-life engineering problems", Applied Intelligence, 2019
[43] Mohammad Dehghani and Pavel Trojovský,"Teamwork Optimization Algorithm: A New Optimization Approach for Function Minimization/Maximization", Sensors, 2021
[44] V. Sathya Durga and T. Jeyaprakash, "An Effective Data Normalization Strategy for Academic Datasets using Log Values," 2019 International Conference on Communication and Electronics Systems (ICCES), 2019, pp. 610-612, doi: 10.1109/ICCES45898.2019.9002089.
[45] N. Zhai, P. Yao and X. Zhou, "Multivariate Time Series Forecast in Industrial Process Based on XGBoost and GRU," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 1397-1400, doi: 10.1109/ITAIC49862.2020.9338878.
[46] Yuying Chen, "Crowd Behaviour recognition using Enhanced Butterfly Optimization Algorithm based Recurrent Neural Network", Multimedia Research, Vol 3, No 3, 2020.
[47] G.Gokulkumari, "Classification of Brain tumor using Manta Ray Foraging Optimization-based DeepCNN classifier", Multimedia Research, Vol 3, No 4, 2020.
[48] Snehal S. Shinde, "Enhanced Manta-Ray Foraging Optimization Algorithm based DCNN for Lane Detection", Multimedia Research, Vol. 4, Issue 3, 2021.
[49] Ayesha Hojage, "Race Detection using Mutated Salp Swarm Optimization Algorithm based DBN from Face Shape Features", Multimedia Research, Vol. 4, Issue 2, 2021.
[50] Renjith Thomas and MJS. Rangachar, "Hybrid Optimization based DBN for Face Recognition using Low-Resolution Images", Multimedia Research, Vol.1,No.1, pp.33-43,2018.
[51] Dataset1,WUSTLIIOT2018Dataset(https://www.cse.wustl.edu/~jain/iiot/index.html)
[52] Dataset-2 https://sites.google.com/view/iot-network-intrusion-dataset/home