Ingénierie des Systèmes d'Information
Vol. 27, No. 2, April, 2022, pp. 213-221
Journal homepage: http://iieta.org/journals/isi

An Intelligent Multi-Objective Evolutionary Model for Establishing Security in Cyber-Physical Systems

Jampani Satish Babu*, Gonuguntla Krishna Mohan

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur 522302, A.P., India

Corresponding Author Email: jampanisatishbabu@kluniversity.in

https://doi.org/10.18280/isi.270205

Received: 7 January 2022
Accepted: 14 April 2022

Keywords:
cyber-physical system, element-driven problem, multi-objective evolutionary algorithm, optimization, uncertain, feasibility

ABSTRACT

The modelling of an efficient CPS is emphasis with the origin of novel functionality over the applications with various design elements. These elements are generally uncertain where these elements include sensors, scheduling, resources, and optimization process. Here, an extensive analysis is carried out with the modelling of CPS element-driven problem formulation. The formulated problems are resolved using a multi-objective evolutionary (MOEA) algorithm to show the intelligence of the optimization approach in the CPS design level. The proposed MOEA shows the viability of the CPS element-driven problem in an explicit manner. The feasibility of the uncertain CPS is measured with the suitability measures in diverse perspectives. The efficiency of the MOEA is examined over the MATLAB 2020a simulation environment. The proposed MOEA model gives better trade-off in contrary to prevailing optimization approaches.

1. INTRODUCTION

Throughout Cyber-Physical System (CPS) is a physical facility with integrated sensor devices that can be managed and evaluated remotely by electronic methods, which we imagine are scattered virtual servers agents executed by Virtual Network Functions (VNFs), the majority of which are positioned at the edge nodes [1]. Logical control loops over actual lines of communication are used to manage and control CPS. Between the sensors/actuators and the VNFs, these channels are generated. The channel carries data on the facility's state as well as reaches the audience for changing the operating mode. Power grids, smart buildings, next-generation cellular telecommunication networks, healthcare systems, and accurate agricultural systems are just a few examples of where CPSs are being used [2]. CPSs have become increasingly important in our daily lives in recent years. As a result, the effective gathering and processing of the information generated at each CPS's physical portion become critical. To facilitate the efficient extraction of actionable insight from the generated CPS data, multiple study fields and approaches must be merged in the contemporary effort [3]. SDN and NFV, edge computing, system modeling, and machine learning are some of these technologies. The harvesting of despite appropriate may enable proactive CPS supervision by either a high-level layer or even a cross-layer component responsible for implementing coordinated organizational structures in networked use cases involving several administrative subdomains [4]. Because of the non-centralized nature of a CPS, it is managed by numerous virtualized (i.e. VNF) actors. As a result, each VNF agent is in charge of overseeing and controlling a certain section of the CPS [5]. As a result, a specific VNF agent should make individual judgments depending on certain relevant information from the local system. Nonetheless, each VNF agent has specific contextual information that may differ from what is provided to others for a variety of reasons, namely geography. As a result, any VNF agent could make strategic decisions that were compatible with those of others. Compared to centralized decision-making, the CPS may have sub-optimal performances when mitigating the flexibility of a distributed NFV strategy [6]. This centralized management optimization inefficiency is akin to an operating cost, signifying a drop in CPS performance. The investigators propose the use of a term is used to describe to reward collaboration and enable efficient method to alleviate this productivity decrease.

CPS's robustness refers to its capacity to tolerate a known band of unpredictable disruptions, whereas its privacy refers to its design to sustain and safeguard against unplanned and malevolent occurrences. These two characteristics are preventative: the CPS is built to be durable and secure. It is exceedingly expensive to build a secure and reliable CPS, and endpoint protection and robustness are difficult to achieve [7]. As a result, it is vital to assess the system's endurance (post-event) is defined as the model's capacity to recover from disruptive occurrences. CPS is a complicated system with several operational loops operating at various time and space scales [8]. The dependability of the parts involved may (often) be used to estimate the whole overall system performance. The frequency of a structure with no relay nodes is higher than the false alarm rate of any of the program's independent units. Both the qualities of the components and the interface design influence the attributes of a CPS [9]. Dependable and availability analysis is a topic that generalizes fundamental insights and puts them into useful frameworks. CPS reliability and trustworthiness study is often based on established system reliability analysis methods [10]. Some of the approaches are CPS composite reliability as depicted by author [11]. A comprehensive study on the unreliability of CPS needs to anticipate reliability and develop techniques to optimize it.

This is where typical convergent validity and modeling approaches must be used or enhanced.

Some reliability researches are performed to discover faults in network connectivity and to establish the specific robustness requirements of a network. In the planning stage, viability modeling comes before analysis [12]. Later in the design process, when we know more detailed data associated with the implementation, we do a viability study. The building of a model to anticipate the dependability or vulnerabilities of a system based on existing data is known as reliability modeling. We can determine trustworthiness measures for a system using reliability modeling. Algorithmic frameworks like RBD, FT, and others can be used, as well as state-based stochastic models like MC and SPN [13]. Algorithmic models give closed-form calculations that facilitate system dependability directly. The complexity of these systems grows as more components are added (e.g., state-space explosion). As a result, for increasingly complicated systems, alternative models are required. Probabilistic stochastic models, such as Bayesian Networks (BNs), have lately been used to describe dependability, either immediately or via importing fault trees [14].

When a model has been created, it may be tested using either classic quantitative modeling approaches or simulation software. Formal techniques are increasingly being recognized as a helpful tool for designing and assessing models. Academic systems depend on the complicated system's generalization, simplicity, and unreasonable assumptions. This can render them prone to errors, especially in highly complicated systems. In comparison to standard quantitative and simulated approaches, formal methods provide a more severe manner of assessment [14]. The scope of this study does not include network reliability evaluation, assessment, or modeling. Complete research on dependability analysis may be found in a work [14]. Boolean logic, crisp categorization, causality, and mathematical modeling are all used in traditional modeling and reasoning procedures. CPS is meant to contain all of the necessary information to address the issue. Relevant information is important in the actual world. Soft computing approaches are a collection of adaptable computing tools that can cope with ambiguous data and seek approximations. In cyber-physical and other complicated processes, a variety of evolutionary computation approaches may be employed to increase system faithfulness or model durability. CPS, unlike IoT systems, undertakes physical activities defined by global control loops that get critical input from the respondents. In addition, CPS has a wide range of node counts and network control. A hybrid system emerges from this ecosystem of sophisticated smart systems, which uses fuzzy sets, NNs, and SA in various phases [15].

Physical state and computer operations are linked through CPSs and Smart objects. Approaches such as application programming interfaces can be used to specify this connection (APIs). APIs can offer two-way communications between virtual and real plants in broad: the physical state can be detected and form part of the cyberspace state, and the cyber status can trigger the right things to modify the physical state. The defect and hazard models for the virtual and real subsystems are also linked by this cyber-physical connection.

Changes in the behavior of the main facility, particularly due to faults or malicious intent can affect the status of the computer unit.

The physical state can be affected by changes to input in the cyber sub-systems. This link is significant in and of itself since it expands both the cyber and physical components' fault and danger models. However, several other factors make this new sort of connection particularly riskier. For starters, because actual plants are attentive to time, just altering the timing of data in the cyber subsystem might cause difficulties in the physical facilities. Delays in control actuation in a controlled system, for example, will alter the system's reaction, possibly causing the physical plant to become unstable. Second, our segmentation models in computer systems are idealized [15]. In recent decades, several security issues in both software and hardware have been discovered. Session hijacking attacks, which use timing, power usage, electromagnetic signals, and even disc drive noises to reveal the condition of the computer network, are adequate. Side channels in CPSs and IoT systems also jeopardize safety, data security, and privacy. In a power system, for example, side channels may allow customer data to escape, which is usually unrelated to plant activity; yet, those same side channels might be used to target the core infrastructure. The explanations of information systems partition metaphors are idealized in the same way that effectiveness and performance and zero coupling variables are idealized in complex processes [15]. However, this work concentrates on handling all these issues with the adoption of local and global intrusion detection processes using a deep network model.

The work is organized as: Section 2 provides a detailed analysis with the proposed hierarchical model for local and global intrusion detection. The numerical outcomes acquired from the proposed multi-objective evolutionary (MOEA) algorithm are provided and discussed in section 3 which is followed by the research summary in section 4.

2. METHODOLOGY

This section describes the detailed explanation regarding the design model of CPS to fulfill security and improve the system model. It is composed of three phases: pre-processing, global and local intrusion detection, and a hierarchical model is used for classification. Finally, some metrics like accuracy, F-score, recall, error rate, confusion matrix are evaluated to show the significance of the MOEA model. Figure 1 depicts the block diagram of the anticipated MOEA model.

2.1 NSL-KDD dataset

NSL-KDD is an improved version of KDD'99 dataset to resolve inherent problems. It is a benchmark dataset used to help the investigators to analyze the threats that occur over the generic system model. It has some set of records with reasonable training and testing sets. The preliminary advantage of using this dataset is its competency to run the experiments without selecting any smaller set of the dataset. The evaluation outcomes of these research works are more comparable and consistent. It may not hold any redundancy of records over the available training sets. Therefore, the classifier does not bias with any recurrent records. It may not have any duplicate records and therefore the learner’s performances are biased with better prediction rates. The numbers of chosen records are inversely proportional to the original dataset percentages. As an outcome, the classification rate changes extensively which is proficient for accurate computation. The number of training and testing sets records are sensible.

2.2 System model

Assume, a multi-objective evolutionary model as depicted in Figure 1. The model is composed of multiple edge servers and global servers. Generally, edge servers accomplish pre-processing and local intrusion detection (LID) of various device-based data patterns. The edge server forwards the data and processes it over the cloud and it performs global intrusion detection with higher computational capacity. The intelligent hierarchical model includes three stages:

Stage 1: The server is armed with higher computational capacity than servers and this server offloads the processed data to the server for global edge servers with the correlated data patterns over the data space, i.e. among the IoT devices. Global intrusion detection adopts learning methods that are commonly applied over natural language processing.

Stage 2: Multiple edge servers achieve processing and local intrusion detection among the CPS. The data is normalized and reframed over the heterogeneous devices. The servers adopt PCA for reducing the dimensionality and adopt mapping for diminishing the computational complexity for intrusion detection. The server uses a sequential prediction approach to identify anomalies of every device based on the historical data patterns.

Stage 3: IoT devices produce data over the servers.

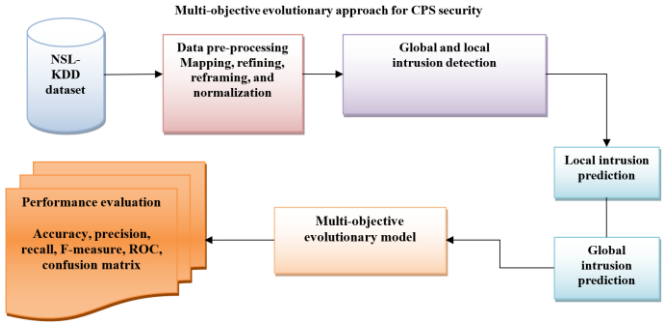

Figure 1. Block diagram of MOEA model

Table 1. Redundant records in KDD testing set

Attacks	250,436	29,378	88.26%
Normal	60,591	47,911	20.92%
Total	311,027	77,289	75.15%

Table 2. Redundant records in KDD training set

Attacks	3,925,650	262,178	93.32%
Normal	972,781	812,814	16.44%
Total	4,898,431	1,074,992	78.05%

Table 3. NSL-KDD dataset attributes

S. No	Label	Attributes	S. No	Label	Attributes	S. No	Label	Attributes	S. No	Label	Attributes
1	B	Duration	10	C	Hot	23	T	Count	32	H	dst_host_count
2	B	Protocol_type	11	C	Num_failed_ Logins	24	T	serror_rate	33	H	dst_host_srv_count
3	B	Service	12	C	Logged_in	25	T	reerror_rate	34	H	dst_host_same_srv_rate
4	B	src_byte	13	C	Num_Compromised	26	T	same_srv_rate	35	H	dst_host_diff_srv_rate
5	B	src_bytes	14	C	Root_shell	27	T	diff_srv_rate	36	H	dst_host_same_src_port_rate
6	B	dat_bytes	15	C	su_attempted	28	T	srv_count	37	H	dst_host_diff_src_port_rate
7	B	Flag	16	C	Num_root	29	T	srv_serror_ rate	38	H	dst_host_serror_rate
8	B	Land	17	C	Num_file_ Creations	30	T	srv_rerror_ rate	39	H	dst_host_srv_serror_
9	B	wrong_fragments	18	C	Num_Shells	31	T	srv_diff_host_ rate	40	H	dst_host_srv_reerror_rate
		Urgent	19	C	Num_access_Files				41	H	dst_host_srv_reerror_rate
			20	C	Num_outbound_Cmds				42	-	Class
			21	C	is_hot_login						
			22	C	is_guest_login						

The Table 1 and 2 describes the dataset attributes with 42 attributes in which 41 attributes are categorized into four diverse classes as shown below:

(1) ***Basic*** (***B***) Features are individual TCP connection attributes.

(2) ***Content*** (***C***) Features are connection recommended using domain knowledge attributes.

(3) ***Traffic*** (***T***) Features are evaluated two-second time window attributes.

(4) ***Host*** (***H***) Features are attacks that lack for two seconds.

2.3 Intrusion detection framework

This section offers an outline of the anticipated framework for intrusion detection (ID) in the context of incoming data. The flow chart is anticipated in Figure 2 which possesses three data processing stages: 1) pre-processing; 2) local intrusion detection (LID) and 3) global intrusion detection (GID). The description of the three-stage model is provided as below:

Stage 1: Pre-processing is the preliminary stage of the proposed hierarchical intelligent model which uses the NSL-KDD dataset as the input to the hierarchical CPS system. The preliminary process is executed over the server-side which is composed of three pre-processing steps: 1) normalization, dimensionality reduction, and mapping. As the CPS is heterogeneous, the processing unit requires data reframing in the same format and the device payloads vary from one another. Some device payloads are qualitative and quantitative where various payloads are normalized to execute the intrusion detection process. Some indicators of the CPS data over the heterogeneous environment remain the same and therefore it is redundant in the intrusion detection process. Therefore, the redundant indicators are discarded to reduce the computational complexity. This work adopts PCA for dimensionality reduction over heterogeneous data. The CPS is a typical appliance with some specific functions and its behavior patterns are static relatively and limited. Therefore, the hierarchical framework is competent to capture the probable benign nature of the IoT devices where benign patterns if every device is computed easily. The reduced data

vectors are categorized with diverse classes in correspondence to the data behaviors. Every class is mapped to a certain data symbol S_t^i where i specifies the device index and t specifies the time slot. Further, symbol mapping diminishes the multi-dimensionality vector computation improves the global and local intrusion detection feasibility.

Stage 2: The data processing is performed over the local intrusion detection through the deep network model which is carried out over the edge server. The data vector classification i, specifies the symbol sequences $S^i = [S_1^i, S_2^i, \dots, S_t^i]$ are recorded over the local server where the symbol sequence specifies the device pattern. Here, the anomalies are predicted based on the probability of the symbol occurrence over the cyber-physical systems as the data devices generally follow certain historical patterns. The deep network model is used because of its time series prediction for tuning the parameters (less) compared to the existing LSTM.

Stage 3: The successive stage is the GID with the conditional random field which shows the identity of its sequential data patterns over the local server and every symbol is provided to the server for global prediction, i.e. every symbol is tested and its likelihood occurrence of every data symbol is analyzed based on the symbol correlation. Therefore, the conditional field is used for prediction purposes in the context of the global symbol space.

The KDD-test set is composed of 25192 instances and the KDD-training set is composed of 22544 instances. The attribute labeled 42 in the dataset is 'class attribute' specifies Table 3 that the given instances are normal or attack.

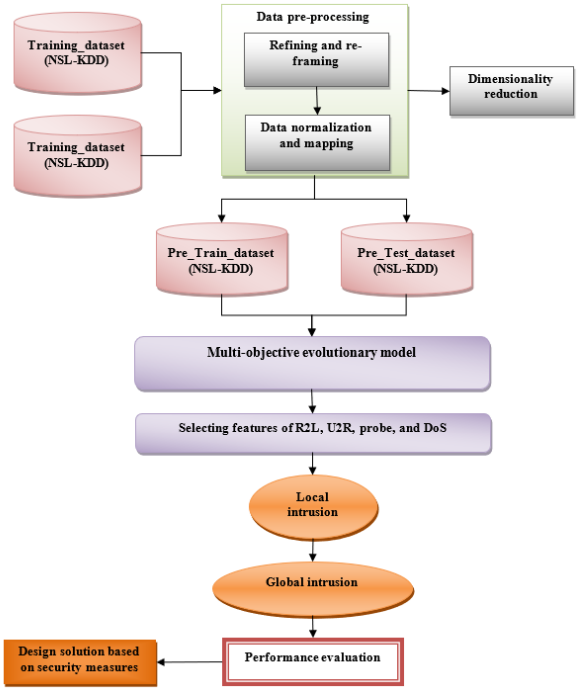

Figure 2. Overall framework of MOEA

2.4 Data model

With the provided raw dataset of CPS $\mathrm{D} = \{D_t^i \in \mathbb{R}^{l_i} | t = 1,2,\dots,T,\ i = 1,2,\dots,N\}$ where N specifies the number of devices and l_i specifies the data vector length of the device i where $\mathbb{D}$ is normalized, re-framed, and mapped to the symbol set $\mathbb{S} = \{S_t^i | t = 1,2,\dots,T,\ i = 1,2,\dots,N\}$ where $S_t^i \in \{1,2,\dots,J_i\}$ and J_i specifies the number of data classes, i.e. number of data symbols of provided i. Thus, the data is an index using the available devices and time.

2.5 Element driven-problem definition

This work intends to evaluate the anomalous score of the CPS to determine the abnormality which is depicted globally and locally using the edge server. The symbol S_t^i is related to the device i at t time determined as the local intrusion where the probability of the occurrence is below the local threshold θ_l is depicted as in Eq. (1):

$$A_{l_t}^i = \begin{cases} 1 & P_{l_t}^i = P(S_t^i | S_{t-1}^i, S_{t-2}^i, \dots, S_{t-k}^i) < \theta\_l \\ 0 & else \end{cases} \quad (1)$$

where, $P_{l_t}^i$ specifies the S_t^i likelihood of historical device patterns and $A_{l_t}^i$ specifies local anomalous. The device symbol S_t^i of the device at t time is specified as the global intrusion when the global probability occurrence P_{gt}^t is below the threshold level θ_g where the computation is mathematically expressed as in Eq. (2):

$$A_{g_t}^i = \begin{cases} 1 & P_{g_t}^i = P\left(S_t^i | S_t^1, S_t^2, \dots, S_t^N\right) < \theta\_g \\ 0 & else \end{cases} \quad (2)$$

where, $A_{g_t}^i$ specifies the S_t^i as a global intrusion. In overall anomaly computation, whether S_t^i is an intrusion or not and it is provided as the Boolean outcome of $A_{l_t}^i$ and $A_{g_t}^i$ is expressed as in Eq. (3):

$$A_t^i = \begin{cases} 1 & A_{g_t}^i || A_{l_t}^i == 1 \\ 0 & else \end{cases} \quad (3)$$

where, $A_t^i = 1$ specifies the S_t^i intrusion.

2.6 Refining data vectors

This section initiates the data refinement process where the data packets (raw) D_t^i over the sequential data packet $D_i^l, D_i^2, \dots, D_i^t$ of i devices are normalized and reframed to C_t^i based on the dataset features with specific characteristics (c_1, c_2, …, c_8). The data is diminished to X_t^i through PCA and mapped as data symbols S_t^i.

2.7 Normalizing and reframing

The reframed data packet normalized is based on the chosen features as depicted. The features are chosen based on the PCA technique where the data values c_1, c_2, …, c_8 are linearly normalized from 0→1 about the maximal and minimal feature values (binary values). Therefore, all the features are normalized among 0 and 1 to attain generic processing over the packets received on the CPS devices. It is expressed as in Eq. (4):

$$[C_1, C_2, \dots, C_N]^{tr} = \begin{bmatrix} C_t^1 & C_{t-1}^1 & \dots & C_{t-k}^l \\ \dots & \dots & \dots & \dots \\ C_t^N & C_{t-1}^N & \dots & C_{t-k}^N \end{bmatrix} \quad (4)$$

where, $C_i = \left[C_t^i, C_{t-1}^i, \dots, C_{t-k}^i\right]^{tr}$ specifies the sequential data of device i with window k time slots and A^{tr} specifies the transposition matric of A.

2.8 Dimensionality reduction

The normalized and reframed data from D_t^i to C_t^i specifies the data refinement by filtering the redundant features through the PCA. Here, C_t^i specifies the value transformed linearly with zero mean and it is expressed as in Eq. (5):

$$\bar{C}_t^i = C_t^i - \frac{1}{8}\sum_{j=1}^{8} C_t^{i,j} \tag{5}$$

where, m=1, 2, …, 8 specifies the entry index C_t^i. Therefore, the square sum of every entry specifies the $\bar{C}_t^i$ where covariance matrix is expressed as in Eq. (6):

$$Covariance^i = \bar{C}^i(\bar{C}^i)^{tr} \tag{6}$$

$$Covariance^i = \begin{bmatrix} \bar{C}_t^i(\bar{C}_t^i)^{tr} & \bar{C}_t^i(\bar{C}_{t-1}^i)^{tr} & \dots & \bar{C}_t^i(\bar{C}_{t-k}^i)^{tr} \\ \dots & \dots. & \dots & \dots \\ \bar{C}_{t-k}^i(\bar{C}_t^i)^{tr} & \bar{C}_{t-k}^i(\bar{C}_{t-1}^i)^{tr} & \dots & \bar{C}_{t-k}^i(\bar{C}_{t-k}^i)^{tr} \end{bmatrix} \tag{7}$$

where, $\bar{C}^i = \left[\bar{C}_t^i, \bar{C}_{t-1}^i, \dots, \bar{C}_{t-k}^i\right]^{tr}$. The cov^i specifies the real-diagonalizable matrix and it is expressed as in Eq. (8):

$$A_i = \begin{bmatrix} \lambda_{i,1} & \dots & \lambda_{i,n} \\ \vdots & \ddots & \vdots \\ \dots & \dots & \lambda_{i,k+1} \end{bmatrix} \tag{8}$$

where, $\lambda_{i,j}$ specifies the j^{th} Eigenvalue of cov^i. The redundant Eigenvectors are related to the small eigenvalues that are discarded from another sequence. Thus, the analysis from $\bar{C}^i$ to X^i with dimensionality reduction and $\bar{C}_t^i$ is converted to X_t^i.

2.9 Mapping symbol

For every row vector X_i specifies X_t^i which is related to the device data at t. The objective is to evaluate data from vectors, i.e. mapping. The model adopts clustering algorithm for data device clustering into J_i classes. Thus, the device set is specified as $\mathbb{S}^i = \{S_t^i | 1,2,\dots,J_i\}$ where $\mathbb{S}^i \subset \mathbb{S}, J_i$ specifies the correlation coefficient to compute the data cluster using various parameters. The mapped data towards the symbol set and the CPS data volume is dimensioned further. The noise data interference is smoothed as the data over the provided cluster is specified using the cluster center and the data variations of the clusters are eliminated. Figure 3 depicts the MOEA framework for security establishment in CPS.

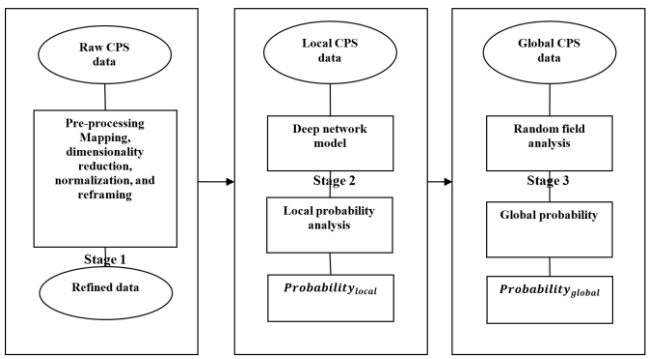

Figure 3. MOEA framework

2.10 Intelligent hierarchical model for global and local intrusion detection

The anticipated model anticipates an approach for anomaly detection in the CPS system by determining the global and local intrusion models. Here, a novel deep network model is used for performing local intrusion detection over the local servers. The existing network model merges the input and forget gate as the simple gate format. The recurrent unit captures the dependencies adaptively over certain time scales. Therefore, the enabling of the deep network model is revealed based on the symbol correlation of various CPS devices with the provided time. The learning ability of the network model is preserved and the provided deep structure diminishes the computational cost. The proposed intelligent hierarchical model is applied where x_t provides the present input state S_t^i and $\hat{y}_t{}^i$ specifies the probability of the occurrence and the weighted coefficient matrices of the deep network model. W_r and W_z specify the trained data samples. In the proposed MOEA framework, the sensing data of CPS is considered as the sentence where t^{th} is used for training the system model. The sentences are processed individually over the CPS server over t time slot where the network unit evaluates the input, output, and prior memory content through the above-mentioned equations. The input and output parameters are expressed as in Eq. (9):

$$\begin{cases} x^i = [S_1^i, S_2^i, \dots, S_t^i \; l \in \mathbb{R}^{1*T} \\ \hat{y}_t^i = [p(1), p(2), \dots, p(J_i)] \in \mathbb{R}^{1*J_i} \\ W_z^i . W_r^i, W^i \in \mathbb{R}^{k*(k+1)} \\ h_t^i, r_t^i, Z_t^i \in \mathbb{R}^{k*1} \\ V^i \in \mathbb{R}^{J_i*k} \\ y_t^i \in \mathbb{R}^{J_i*1} \end{cases} \tag{9}$$

where, k specifies the internal network memory, $\hat{y}_i^t$ specifies the probability vector and y_i^t specifies the ground-truth value at t time. The cross-entropy L_i^t is mathematically expressed as in Eq. (10):

$$L_t^i = -\hat{y}_t{}^i(\log y_t^i) \tag{10}$$

where, the log specifies the logarithm (element-wise) function. The total cross-entropy loss is expressed as in Eq. (11):

$$\Theta^{i*} = \arg\min_{\Theta^i} L^i \tag{11}$$

where, $\Theta^i = \{W_z^i, W_r^i, W^i, V^i\}$. The trained network model is used for computing the probability vector $\hat{y}_i{}^t$ is provided to predict the local intrusion of every device i. The input data probability is evaluated and the threshold probability θ_l is adjusted automatically from 1 to 0 by the server. The detection metrics like TPR, FPR, accuracy, precision, and F-score are evaluated. The threshold model is analyzed based on maximal precision. The threshold and parameters are trained and the network model is used for local intrusion detection. The nature of network traffic is changed and the trained and parameters are re-trained.

2.11 Global intrusion detection

Here, a conditional random field is adopted for measuring global intrusion detection where the conditional distributions

of hidden states are evaluated based on the provided observations. In the proposed MOEA model, the conditional random field model is used for constructing the probability distribution of every state and measures the labeling of every CPS-based data. Assume, the global intrusion state of the CPS system at t tie which is represented as $[A_{gt}^1, A_{gt}^2, \dots, A_{gt}^N]$ which is known as Markov-chain, and the input is provided in a sequential form $x_t = [S_1^t, \dots, S_N^t]$ which is adopted for language processing. The conditional random fields (CRF) are appropriate for categorizing and modeling CPS data as the data is intrinsically considered based on the t time slot. The data produced by the device i at t time is based on the probability distribution of variables modeled through the random field which provides the probability factorized into functional product over intrinsic features. The probability of the variable state intrusion $A_{gt} = [A_{gt}^1, A_{gt}^2, \dots, A_{gt}^N]$ which is provided as in Eq. (12):

$$P(A_{gt} \mid x_t) = \frac{1}{Z(x_t)} \exp\left\{\sum_{i=1}^{N}\sum_{w=1}^{W} \mu_w f_w\,(i, A_{gt}^i, A_{gt}^{i-1}, x_t)\right\} \tag{12}$$

where, f_w specifies the feature function and its weight factor μ_w. $Z(x_t)$. It is expressed as in Eq. (13):

$$Z(x_t) = \sum_{A'_{gt}} \exp\left\{\sum_{i=1}^{N}\sum_{w=1}^{W} \mu_w f_w\,(i, A_{gt}^i, A_{gt}^{i-1}, \quad x_t)\right\} \tag{13}$$

where, A_{gt}^i specifies the probability vector of the state vector $[A_{gt}^i, A_{gt}^2, \dots, A_{gt}^N]$ at t time. The feature function is an indication that demonstrates the adjacent state pair and the random field is completely characterized by the weighted factor/feature set, i.e. (f, μ). It is a supervised model which intends to process the training stage and it evaluates the model parameters based on the labeled data samples. The objective function is expressed as in Eq. (14):

$$\sum_{t=1}^{T} \log P(A_{gt}|x_t) = \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{w=1}^{W} \mu_w f_w\,(i, A_{gt}^i, x_t, A_{gt}^{i-1}) - \sum_{t=1}^{T} \log Z\,(x_t) \tag{14}$$

To train the model parameters, the value of the objective function needs to be maximized as it is due to the convex nature of the objective function. The optimization is guaranteed to attain a global solution. The random field model is trained and it is functional to identify the input data state x_t at time t through the proposed hierarchical model. It is expressed as in Eq. (15):

$$A_{gt}^* = \arg\max_{A_{gt}} P(A_{gt}|x_t) \tag{15}$$

where, A_{gt}^* specifies the intrusion detection mechanism with maximal likelihood. The training process of the local intrusion detection is based on the probability of acquired input, threshold probability which is adjusted automatically from 1 – 0 using the CPS server model. The threshold value is acquired based on the maximal precision value. The network-changing nature relies on the proposed hierarchical model. The algorithm of the anticipated intelligent hierarchical model is depicted in Algorithm 1:

Algorithm 1 MOEA model

Input: $D = \{D_t^i \in R^{l_i} \mid t = 1, 2, \dots, T, i = 1,2, \dots, N\}$
Output: $\{A_t^i \in \{0,1\} \mid t = T+1, T+2, \dots, T', i = 1,2, \dots, N$

1. $D \to C, C \to X, X \to S, \Theta = \{W_z, W_r, W, V\}$; //data normalization, dimensionality reduction, mapping, and parameter initialization,
2. **for** all *i*=1 to *N* do
3. **While** the model does not converge do
4. Evaluate forward and backpropagation;
5. Perform parameter updation;
6. **end while**
7. **end for**
8. **for** all *i*=1 to *N* do
9. **for** all *t*=*T*→*T'* do
10. compute $\hat{y}_t'$ *and* $A_{l_t}^i$;
11. **end for**
12. **end for**
13. Initialize μ parameter;
14. **While** the model does not converge do
15. Compute *U*, gradient value, and μ;
16. **end while**
17. **for** *t*=*T* to *T'* do
18. compute A_{gt}^i;
19. Predict global intrusion;
20. **end for**

3. EXPERIMENTAL OUTCOMES

The proposed MOEA model performance is compared with various existing approaches for predicting anomaly hierarchically over the CPS system. The preliminary stages include pre-processing and local intrusion detection simulated under i7 processor at 2.3 GHz and 8 GB RAM where MATLAB 2020a is used as a simulator. The design-based optimization is used to identify the threat that consumes the cyber system resources, consumes more energy, and causes the error. These factors lead to system performance degradation.

True positive (TP): The proposed classifier needs to determine accurately the class feature to predict where the attack is identified.

True Negative (TN): The classifier needs to determine the class features are negative accurately.

False Positive (FP): The classifier inaccurately determines the normal traffic as an attack pattern.

False Negative (FN): The proposed classifier incorrectly classifies the attack as normal traffic.

With these measures, various metrics are evaluated efficiently. They are ROC, False Positive Rate (FPR), accuracy, precision, F-measure, and recall with error rate computation. Accuracy is defined as the probability of predicting the occurrence of an attack or normal traffic accurately. It is mathematically expressed as in Eq. (16)-(22):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

$$FPR = \frac{FP}{FP + TN} = 1 - specificity \quad (18)$$

$$AUC = \int_0^1 ROC\ (t)dt \quad (19)$$

$$Precision = \frac{TP}{TP + FP} * 100 \quad (20)$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \quad (21)$$

$$RMSE = \sqrt{\left[\sum_{i=1}^{N} (y_i - x_i)^2 / N\right]} \quad (22)$$

Table 4. Accuracy comparison

Iterations MOEA	100 85	300 87	500 92	800 97	1000 99.5
DB	79.45	82.33	90.25	95.6	99
Deep learning RNN [5]	60.35	68.9	72.55	81.96	86.9
federated self-learning [10]	70.11	76.25	81.22	86.99	99.09
federated transfer learning [14]	73.14	79.88	85.2	91.2	99.13
Deep FED [15]	75.55	80.36	87.48	92.33	99.20

Table 5. Precision comparison

Iterations MOEA	100 85.9	300 92.65	500 95.45	800 97.89	1000 100
DB [15]	83.65	88.74	92.44	96.87	100
Deep RNN [5]	71	76	82	90	98
federated self-learning [16]	71.44	76.2	82.51	90.66	98.86
federated transfer learning [17]	76.02	79.2	88.99	93.5	99.34
Deep FED [18]	81.06	83.99	87.25	93.65	98.86

Table 6. Recall comparison

Iterations MOEA	100 85	300 89	500 94	800 85	1000 99.6
DB [15]	81.45	82.33	89.96	91.25	99.5
Deep RNN [5]	71	78	82	89	86
federated self-learning [16]	71.25	78.54	82.69	89.41	96.76
federated transfer learning [17]	73.89	78.88	83.94	93.89	96.82
Deep FED [18]	76.58	80.11	85.62	89.7	97.36

Table 7. F-Measure comparison

Iterations MOEA	100 83	300 92	500 95	800 97	1000 99.8
DB [15]	81.22	87.88	92.15	96.64	99.7
Deep RNN [5]	71	73	81	92	97
federated self-learning [16]	71.10	73.55	81.25	92.85	97.78
federated transfer learning [17]	73.25	78.5	86.9	93.2	98.03
Deep FED [18]	80.05	83.97	91.24	94.55	98.10

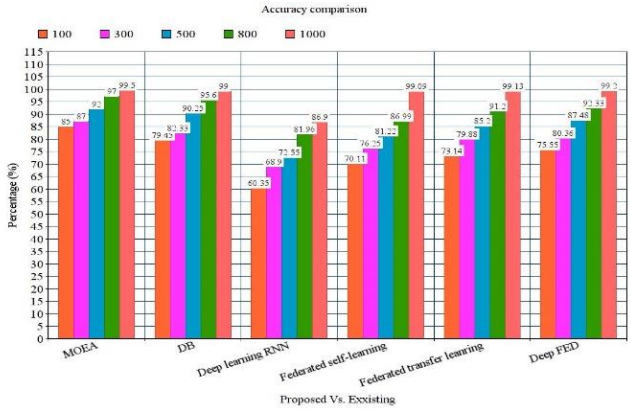

Figure 4. Accuracy comparison

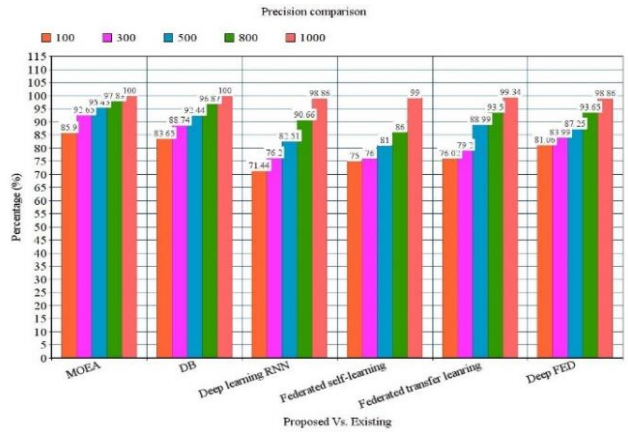

Figure 5. Precision comparison

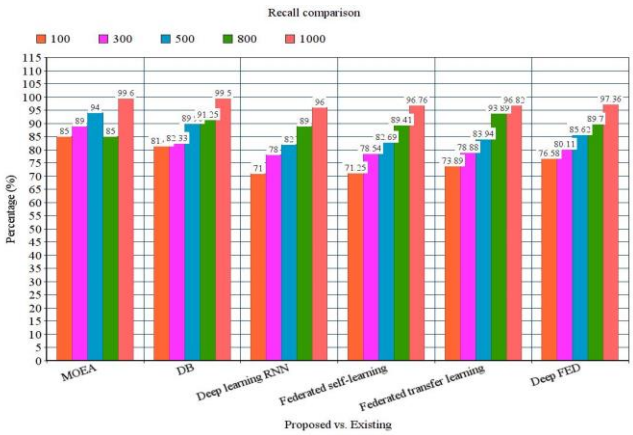

Figure 6. Recall comparison

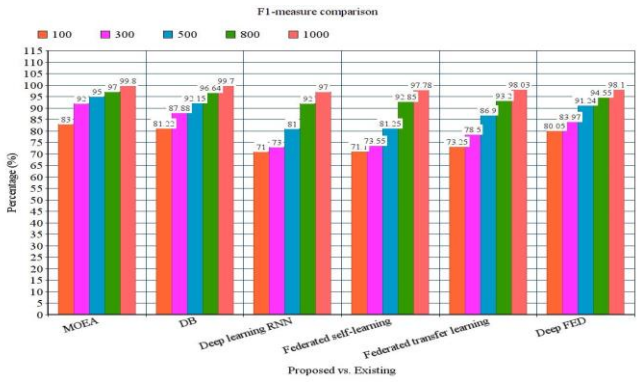

Figure 7. F1-measure comparison

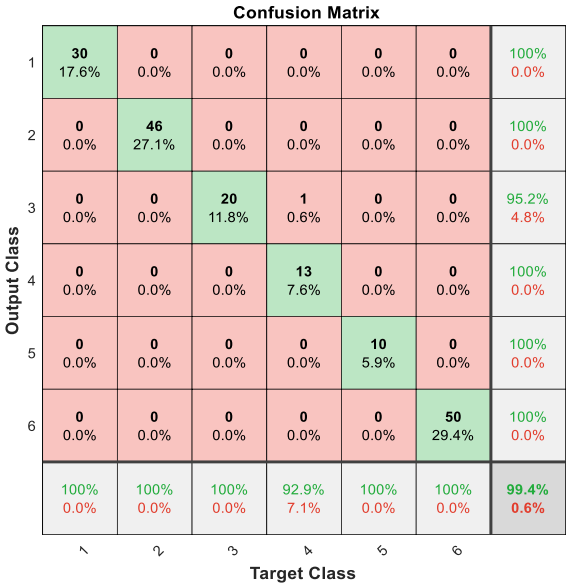

Figure 8. Confusion matrix

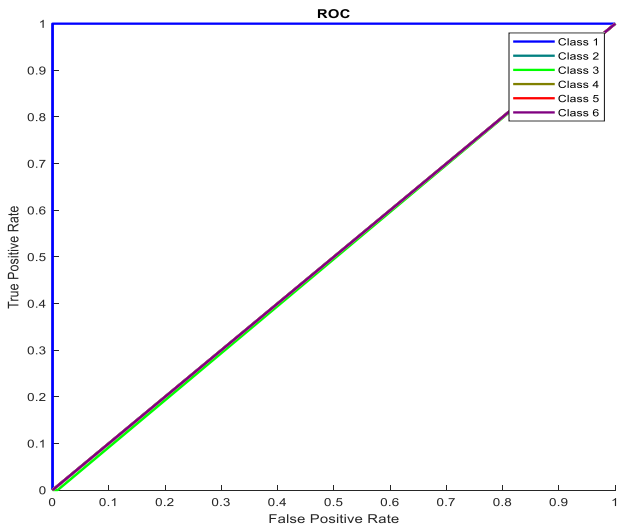

Figure 9. ROC computation

The proposed MOEA model is compared with various existing approaches like DB, Deep RNN, Federated self-learning, federated transfer learning and deep FED model. The major drawbacks rely on the existing approaches are discussed to project its failure in intrusion detection. The deep RNN faces some serious issues like exploding and gradient vanishing problems, complex training and fails to process the long sequential incoming data. The drawbacks with federated self and transfer learning are the complexity towards the understanding of the model and they consume huge time for execution. Similarly, with the deep federated relies on the evaluation with the feature learning and analysis. Without proper feature learning, no further classification is done for predicting the intrusion.

The evaluation is performed for various iterations, i.e. 100, 300, 500, 800, and 1000 respectively. The accuracy of the anticipated MOEA is measured for the 1000th iteration (See Figure 4) where the accuracy is 99% which is 12.6%, 12.5%, 0.31%, 0.27%, and 0.10% higher than DB, DL-RNN, federated self-learning, transfer learning, and Deep FED models as depicted in Table 4. Table 5 depicts the evaluation of the MOEA model over prevailing approaches. The precision of MOEA is 100% which is 1.14%, 0.66%, and 1.14% when compared to federated self-learning, federated transfer learning, and deep FED (See Figure 5). Table 6 demonstrates the recall comparison of MOEA over other models and attains 99.6% which is 2.75%, 2.74%, 2.68%, and 2.14% higher than other approaches (See Figure 6). Table 7 depicts the comparison of F-measure comparison of the MOEA model which is 99.8% which is 1.93%, 1.92%, 1.67%, and 1.6% higher than other approaches (See Figure 7). The error rate of anticipated MOEA is 0.04 and the time consumed for evaluation is 16.85 seconds. Figure 8 shows the confusion matrix generated for actual value and targeted value. Similarly, Figure 9 shows the ROC curve plotting for TPR and FPR rate with class 1-5 respectively.

4. CONCLUSIONS

This research proposes a novel MOEA method used for predicting cyber-attacks over CPS using the deep network model. The prediction process is the combination of both reconstruction and discrimination loss which is needed for data sample mapping. The mapping process is performed for accelerating the loss function and time consumed during the execution process. Some preliminary steps like refining, normalizing, re-framing, and mapping are done to make the model more appropriate for classification purposes. Here, the NSL-KDD dataset is used for computation and various metrics like accuracy, recall, precision, and F-measure is examined. The simulation is done in MATLAB 2020a environment where the performance of the model is compared with existing DB, federated self and transfer learning, and Deep Fed learning model. The metrics comparison with these approaches shows the significance of the proposed MOEA model. However, the major research constraint is the lack of analysis with the prediction latency of cyber-attacks in CPS. However, in the future, this is addressed with the adoption of deep auto-encoders for cyber-attack prediction and intends to reduce the detection latency of the proposed IDS.

REFERENCES

[1] Liu, Y., Peng, Y., Wang, B., Yao, S., Liu, Z. (2017). Review on cyber-physical systems. IEEE/CAA Journal of Automatica Sinica, 4(1): 27-40. http://dx.doi.org/10.1109/JAS.2017.7510349

[2] Oks, S.J., Fritzsche, A., Möslein, K.M. (2017). An application map for industrial cyber-physical systems. In Industrial Internet of Things, pp. 21-46. http://dx.doi.org/10.1007/978-3-319-42559-7_2

[3] García-Valls, M., Dubey, A., Botti, V. (2018). Introducing the new paradigm of social dispersed computing: Applications, technologies and challenges. Journal of Systems Architecture, 91: 83-102. https://doi.org/10.1016/j.sysarc.2018.05.007

[4] Koutsoukos, X., Karsai, G., Laszka, A., et al. (2017). SURE: A modeling and simulation integration platform for evaluation of secure and resilient cyber–physical systems. Proceedings of the IEEE, 106(1): 93-112. https://doi.org/10.1109/JPROC.2017.2731741

[5] Chakraborty, S., Sharma, R.K., Tewari, P. (2017). Application of soft computing techniques over hard computing techniques: A survey. International Journal of

Indestructible Mathematics & Computing, 1(1): 08-17. https://doi.org/10.18510/ijsrtm.2017.542

[6] Khaitan, S.K., McCalley, J.D. (2014). Design techniques and applications of cyberphysical systems: A survey. IEEE Systems Journal, 9(2): 350-365. http://dx.doi.org/10.1109/JSYST.2014.2322503

[7] Zhang, Y., Qiu, M., Tsai, C. W., Hassan, M.M., Alamri, A. (2015). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. IEEE Systems Journal, 11(1): 88-95. http://dx.doi.org/10.1109/JSYST.2015.2460747

[8] Humayed, A., Lin, J., Li, F., Luo, B. (2017). Cyber-physical systems security—A survey. IEEE Internet of Things Journal, 4(6): 1802-1831. http://dx.doi.org/10.1109/JIOT.2017.2703172

[9] Zhu, Q., Basar, T. (2015). Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems. IEEE Control Systems Magazine, 35(1): 46-65. http://dx.doi.org/10.1109/MCS.2014.2364710

[10] Ahmad, W., Hasan, O., Pervez, U., Qadir, J. (2017). Reliability modeling and analysis of communication networks. Journal of Network and Computer Applications, 78: 191-215. http://dx.doi.org/10.1016/j.jnca.2016.11.008

[11] Esmalifalak, M., Liu, L., Nguyen, N., Zheng, R., Han, Z. (2014). Detecting stealthy false data injection using machine learning in smart grid. IEEE Systems Journal, 11(3): 1644-1652. http://dx.doi.org/10.1109/JSYST.2014.2341597

[12] Ahmed, S., Lee, Y., Hyun, S.H., Koo, I. (2018). Feature selection–based detection of covert cyber deception assaults in smart grid communications networks using machine learning. IEEE Access, 6: 27518-27529. http://dx.doi.org/10.1109/ACCESS.2018.2835527

[13] Singh, S.K., Khanna, K., Bose, R., Panigrahi, B.K., Joshi, A. (2017). Joint-transformation-based detection of false data injection attacks in smart grid. IEEE Transactions on Industrial Informatics, 14(1): 89-97. http://dx.doi.org/10.1109/TII.2017.2720726

[14] Lu, C., Saifullah, A., Li, B., et al. (2015). Real-time wireless sensor-actuator networks for industrial cyber-physical systems. Proceedings of the IEEE, 104(5): 1013-1024. http://dx.doi.org/10.1109/JPROC.2015.2497161

[15] Li, B., Lu, R., Wang, W., Choo, K.K.R. (2017). Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system. Journal of Parallel and Distributed Computing, 103: 32-41. http://dx.doi.org/10.1016/j.jpdc.2016.12.012

[16] Goh, J., Adept, S., Tan, M., Lee, Z.S. (2017). Anomaly detection in cyber-physical systems using recurrent neural networks. 2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE), Singapore, pp. 140-145. http://dx.doi.org/10.1109/HASE.2017.36

[17] Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y. (2019). Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Transactions on Industrial Informatics, 16(6): 4177-4186. http://dx.doi.org/10.1109/TII.2019.2942190

[18] Nguyen, T.D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., Sadeghi, A.R. (2019). DÏoT: A federated self-learning anomaly detection system for IoT. In 2019 IEEE 39th International conference on distributed computing systems (ICDCS), Dallas, TX, USA, pp. 756-767. http://dx.doi.org/10.1109/ICDCS.2019.00080