Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07060-4

S.I.: LSNC & OUAI

Variational restricted Boltzmann machines to automated anomaly detection

Konstantinos Demertzis[1] · Lazaros Iliadis[2] · Elias Pimenidis[3] · Panagiotis Kikiras[1,2,3]

Received: 11 September 2021 / Accepted: 5 February 2022

Abstract

Data-driven methods are implemented using particularly complex scenarios that reflect in-depth perennial knowledge and research. Hence, the available intelligent algorithms are completely dependent on the quality of the available data. This is not possible for real-time applications, due to the nature of the data and the computational cost that is required. This work introduces an Automatic Differentiation Variational Inference (ADVI) Restricted Boltzmann Machine (RBM) to perform real-time anomaly detection of industrial infrastructure. Using the ADVI methodology, local variables are automatically transformed into real coordinate space. This is an innovative algorithm that optimizes its parameters with mathematical methods by choosing an approach that is a function of the transformed variables. The ADVI RBM approach proposed herein identifies anomalies without the need for prior training and without the need to find a detailed solution, thus making the whole task computationally feasible.

Keywords Automatic differentiation variational inference · Restricted Boltzmann machines · Condition monitoring · Anomalies detection · Predictive maintenance · Industry 4.0

1 Introduction

Finding the solutions for continuous monitoring the operational status of active equipment using an Industrial Internet of Things (IIoT) network is a key priority of Industry 4.0 [1]. Thus, solutions for prognostic analysis, machine learning and rationalization of preventive maintenance are utilized. Functional control and communication between humans and machines in general becomes clear and well defined with the Condition Monitoring (CoMo) process. CoMo technologies include non-destructive test methods for collecting, through active sensor networks, data directly related to the operational status of the equipment [2].

A typical case is that of signals generated by the vibrations of machines, audible or thermal imaging signals or the analysis of liquids such as oil, fuel. Data are analyzed in real time by the most advanced applications of intelligent algorithms, in order to detect hidden knowledge [3]. This analysis provides important information about the operational condition of the equipment, its possible malfunctions, and its vulnerabilities [4].

Anomaly detection [5] can significantly contribute toward the discovery of hidden knowledge in industrial data. Anomaly detection is the process of identifying data, objects, observations, events, or behaviors that do not conform to the expected pattern of a particular group. Such abnormalities are rare and they are very likely to be associated with significant threats such as [6]:

(a) Malfunctions or misuse of equipment
(b) Adverse situations of their operating environment

✉ Konstantinos Demertzis
kdemertz@fmenr.duth.gr

Lazaros Iliadis
liliadis@civil.duth.gr

Elias Pimenidis
Elias.Pimenidis@uwe.ac.uk

Panagiotis Kikiras
kikirasp@uth.gr

[1] School of Science and Technology, Informatics Studies, Hellenic Open University, Thermi, Greece

[2] Computer Science and Creative Technologies, University of the West of England, Bristol, UK

[3] Department of Computer Science, University of Thessaly, 35100 Lamia, Greece

Published online: 01 March 2022

(c) Intentional malicious interference from external agents, such as digital attacks, malware, information interception [7].

Detection of anomalies is considered one of the biggest and most complex challenges in managing large-scale applications [6, 8–10]. The success of these methods is relaying on comparing the current operating condition of the equipment, with patterns that describe normal cases, such as the average number of concurrent services, the average length of the operating cycle, the rate of return. These methods prove to be very sensitive to small changes in the input vectors. Also, the usual approach employed to identify anomalies is the approximation technique for calculating the expected values under the distribution of the dataset by a continuous sampling for a long time until equilibrium was reached. This is not possible for real-time applications due to the nature of the data and the computational cost that is required.

This fact, under certain conditions, may lead to the modification of the behavior of the learning algorithms and to the questioning of the reliability and safety of the approaches in question. Also, in real-world problems, the data contain noise, errors, or are partially identified, which affect the process of developing robust intelligent models [3, 11].

Estimating uncertainty and simulating highly dynamic processes is a difficult task in general. The data used in the development of these models solely reflect difficult situations, so that the employed methods often fail to generalize [12].

Each learning algorithm has specific biases, which may be related to the chosen values of the related hyperparameters, or to the applied classification methodology, or to the chosen information representation method. This fact makes algorithms vulnerable to specialized attacks. It is a fact that the finite training data do not always reflect reality. The data selection process and the assumption that data have the same distribution as all the unknown cases, introduces another level of bias, which makes intelligent systems vulnerable to *Adversarial Attacks* [13].

This paper proposes a new and innovative anomaly detection system based on RBMs methodology. RBMs are discovering several layers of increasingly complex representations of the input, it comes with an efficient layer-by-layer pre-training procedure, it can be trained on unlabeled data and can be fine-tuned for a specific task using the (possibly limited) labeled data. Second, the approximate inference procedure for RBM's incorporates top-down feedback in addition to the usual bottom-up pass, allowing to better incorporate uncertainty about ambiguous inputs. Third, the parameters of all layers can be optimized jointly by following the approximate gradient of a variational lower bound on the likelihood function. This greatly facilitates learning better generative models.

In this spirit, the proposed system is an innovative RBM that uses the advanced ADVI technique, which operates in an automatic way without training to accurately determine a posterior distribution. Using Variational Distribution as a normal distribution of multiple variables, it is possible to correlate the parameters to solve dynamic problems in real time at an affordable computational cost. This is achieved without the need for prior training of the system and without the need to find a detailed solution, which makes it computationally accessible. This methodology is capable of performing well, even when the nature of the anomaly is new and therefore unknown.

2 Literature review

There is an increasing interest in research related to the anomaly detection systems [8, 9, 14] and specifically in real-time anomaly detection systems [15–17]. For example, the authors of [18] reported inefficiencies in most anomaly-based network intrusion detection systems employing supervised algorithms and suggested an unsupervised outlier detection scheme as a measure to overcome these inefficiencies. The method uses a database containing logs of each cloud service. This technique can detect some types of attacks and faulty services in a cloud environment with high accuracy.

Also, the proposed detection system [19] uses an unsupervised stochastic Restricted Boltzmann Machine algorithm to self-learn the reliable network metrics. This algorithm detects and classifies the type of DDoS attacks in a dynamic network environment by framing a new context. Three modules have been implemented in Mininet, namely DDoS attack generation, Flow collection and attack detection. First, the network topology is created with 5 hosts, 1 controller and 1 switch. Each node in the Mininet runs a virtual machine with a real GNU/Linux kernel. The results prove that the RBM-based DDoS detection system achieves higher accuracy than the existing static methods but only as a batch method that needs a lot of training data.

On the other hand, Variational Autoencoders have been used to identify anomalies [20–22]. For example, the authors of the [23] propose a novel botnet detection method, built upon Recurrent Variational Autoencoder, that effectively captures sequential characteristics of botnet anomalies. The experimental results with large-scale intrusion data (NSL-KDD dataset) show that the proposed method can detect previously unseen botnets by utilizing sequential patterns of network traffic by 85.51 accuracy. Also, can detect botnets in the streaming mode, which is the essential requirement to perform real time, online detection. But the method is time-consuming and needs high computational resources.

Additionally, the authors of the [24] present a robust and unsupervised federated learning system in which the central server employs a conditional variational autoencoder to detect and eliminate malicious model updates. Since the reconstruction error of malicious updates is much larger than that of benign ones, it can be used as an anomaly score. The authors formulate a dynamic threshold of reconstruction error to differentiate malicious updates from normal ones, based on this idea. The proposed model is tested in 4 competitive datasets, namely Vwhicle, MNIST, FEMNIST and a Synthetic dataset. The experiments have shown a competitive performance over existing aggregation methods under Byzantine attack and targeted model poisoning attack. A critical disadvantage of the method is that needs very extensive and sophisticated parameter tuning in order to be effective.

A major problem related to the reliability and security of computational intelligence methods is the design of the appropriate input in a specific straight forward way [25]. Failure to do so, leads learning algorithms to wrong results (Adversarial Attacks) [26]. The problem arises from the fact that learning techniques are designed for stable environments, in which training and test data are considered to be generated from the same (possibly unknown) distribution. For example, a trained neural network represents a large decision limit, which corresponds to a common class. A properly designed and implemented attack, which corresponds to a modified input that may come from a slightly differentiated data set (LFW face dataset), may lead the algorithm to make a wrong decision (wrong class) [27]. This yields several interesting observations: (1) Within a specific range, as the perturbation size increases, while the recognition rate for dodging attacks drops, the recognition rate for impersonation attacks increases. (2) Dodging attacks and Impersonation attacks have different sensitivity to the perturbation. A small perturbation (0.001) can significantly help the impersonation attack but could not support the dodging attack. A significant perturbation (0.1) can help the dodging attack, but not the impersonation attack. It could be presumed that dodging attack needs less control on the perturbation. As long as the perturbation makes the original image not to be recognizable, while the impersonation attack needs more control to the perturbation, there is still a chance for it not to be recognized for the victim images. (3)The iteration does affect the performance. More iterations lead to higher recognition rate. It is good for impersonation attacks but not good for the dodging attacks. (4) The granularity of perturbation is also critical to the perturbation generation. The recommendation is to start with a small scale, such as 0.001.

On the other hand, anomaly detection systems are an essential cog of the network security suite that can defend the network from malicious intrusions and anomalous traffic. Many anomaly detection methods have been proposed in the literature for the detection of anomalies [7, 16, 17, 28]. However, recent works have shown that models are vulnerable to adversarial perturbations through which an adversary can cause IDSs to malfunction by introducing a small impracticable perturbation in the network traffic. There are several defense methods to adversarial attacks, which are deliberately created to fool learning models [29–31].

For example, to improve the attack performance against the variational autoencoder, which is robust to tiny perturbations through uncertainty modeling, the authors of [32] design a mechanism to weaken its robustness by introducing a variance regularization to the optimization. Simulation results show that the adversarial attacks generated by a universal adversarial sample generator can effectively degrade the performance of the autoencoder-based systems. The system was tested with 3 vast datasets (MNIST, FEMNIST, and CIFAR-10). However, state-of-the-art attack methods can generate attack images by adding small perturbations to the source image [13]. These attack images can fool the classifier but have little impact to humans. Therefore, such attack instances are difficult to generate by searching the feature space. How to design an effective and robust generating method has become a key requirement. Inspired by adversarial examples, we propose two novel generative models to produce adaptive attack instances, in which a conditional generative adversarial network is adopted and a distinctive strategy is designed for training. The authors of [30] propose two models that can reduce the generating cost and improve robustness, resulting in about one-fifth of the running time for producing attack instance. The system was tested in MNIST and FEMNIST datasets with high accuracy. Nonetheless, the model is not adequate for real-time applications.

3 Proposed anomaly detection model

In this work, a RBM is proposed which combined with the ADVI methodology, creates a framework for dealing with Adversarial Attacks. It can automatically detect anomalies in dynamic systems based on the posterior distribution of the data it uses. The proposed RBM can be defined based on random Markovian fields, as a two-dimensional graph with non-directed edges.

It comprises of m visible neurons $V = V_1, V_2, \ldots, V_m$ and n hidden ones $H = H_1, H_2, \ldots, H_n$. Since the work studies the binary problem of the anomalies' existence, the model takes binary values [33]. Thus, the random variables

(V, H) take values $(u, h) \in \{0, 1\}^{m+n}$. The common probability distribution of the model is the Boltzmann distribution [34, 35]:

$$p(\boldsymbol{u}, \boldsymbol{h}) = \frac{1}{Z} e^{-E(\boldsymbol{u}, \boldsymbol{h})} \tag{1}$$

The model's Energy function $E(u, h)$ is defined as follows [36]:

$$E(\boldsymbol{u}, \boldsymbol{h}) = -\sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} h_i u_j - \sum_{j=1}^{m} b_j u_j - \sum_{i=1}^{n} c_i h_i \tag{2}$$

The parameters of the model for $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$ are the weights w_{ij} and the biases b_j and c_i for the jth visible and ith hidden neuron. All change parameters get real values. RBMs do not allow connections between neurons belonging to the same level [37].

This implies a conditional independence for elements at a certain level so that [35, 38]:

$$p(\boldsymbol{h}|\boldsymbol{u}) = \prod_{i=1}^{n} p(h_i|u) \tag{3}$$

$$p(\boldsymbol{u}|\boldsymbol{h})\& = \prod_{j=1}^{m} \left[p\left(u_j|\boldsymbol{h}\right) \right] \tag{4}$$

The marginal distribution is determined as follows [22, 39–41]:

$$\begin{aligned} p(u) &= \sum_h p(u, h) = \frac{1}{Z} \sum_h e^{-E(u,h)} \\ &= \frac{1}{Z} \sum_{h_1} \sum_{h_2} \cdots \sum_{h_n} e^{\sum_{j=1}^m b_j u_j} \prod_{i=1}^{n} e^{h_i \left(c_i + \sum_{j=1}^m w_{ij} u_j \right)} \\ &= \frac{1}{Z} e^{\sum_{j=1}^m b_j u_j} \sum_{h_1} e^{h_1 \left(c_1 + \sum_{j=1}^m w_{1j} u_j \right)} \cdots \\ &\sum_{h_n} e^{h_n \left(c_n + \sum_{j=1}^m w_{nj} u_j \right)} \\ &= \frac{1}{Z} e^{\sum_{j=1}^m b_j u_j} \prod_{i=1}^{n} \sum_{h_i} e^{h_i \left(c_i + \sum_{j=1}^m w_{ij} u_j \right)} \\ &= \frac{1}{Z} \prod\nolimits_{j=1}^m e^{b_j u_j} \prod_{i=1}^{n} \left(1 + e^{c_i + \sum_{j=1}^m w_{ij} u_j} \right) \end{aligned} \tag{5}$$

To calculate the conditional probabilities of a given hidden or visible neuron, the set of visible variables can be defined as u_{-l} without considering the variable l (α and β are vectors for the visible and hidden units) [38, 42, 43]:

$$\begin{aligned} a_l(h) &= -\sum_{i=1}^{n} w_{il} h_i - b_l \\ \beta(u_{-l}, h) &= -\sum_{i=1}^{n} \sum_{j=1, j \neq l}^{m} w_{ij} h_i u_j - \sum_{j=1, j \neq l}^{m} b_j u_j - \sum_{i=1}^{n} c_i h_i \end{aligned} \tag{6}$$

The Energy function $E(u, h)$ is given by the following relation [44, 45]:

$$E(\boldsymbol{u}, \boldsymbol{h}) = \beta(\boldsymbol{u}_{-l}, \boldsymbol{h}) + u_l a_l(\boldsymbol{h}) \tag{7}$$

The conditional probability of V_l is equal to [38, 46, 47]:

$$\begin{aligned} p(V_l = 1|\boldsymbol{h}) = p(V_l = 1|\boldsymbol{u}_{-l}, \boldsymbol{h}) &= \frac{p(V_l = 1, \boldsymbol{u}_{-l}, \boldsymbol{h})}{p(\boldsymbol{u}_{-l}, \boldsymbol{h})} \\ &= \frac{e^{-E(u_{l=1}, \boldsymbol{u}_{-l}, \boldsymbol{h})}}{e^{-E(u_{l=1}, \boldsymbol{u}_{-l}, \boldsymbol{h})} + e^{-E(u_{l=0}, \boldsymbol{u}_{-l}, \boldsymbol{h})}} \\ &= \frac{e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h}) - 1 a_l(\boldsymbol{h})}}{e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h}) - 1 a_l(\boldsymbol{h})} + e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h}) - 0 a_l(\boldsymbol{h})}} \\ &= \frac{e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h})} \cdot e^{-a_l(\boldsymbol{h})}}{e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h})} \cdot e^{-a_l(\boldsymbol{h})} + e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h})}} \\ &= \frac{e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h})} e^{-a_l(\boldsymbol{h})}}{e^{-\beta(\boldsymbol{u}_{-l}, \boldsymbol{h})} \cdot \left(e^{-a_l(\boldsymbol{h})} + 1 \right)} \\ &\frac{e^{-a_l(\boldsymbol{h})}}{e^{-a_l(\boldsymbol{h})} + 1} \frac{\frac{1}{e^{a_l(\boldsymbol{h})}}}{\frac{1}{e^{a_l(\boldsymbol{h})}} + 1} \frac{1}{1 + e^{a_l(\boldsymbol{h})}} \\ &= \sigma[-a_l(\boldsymbol{h})] \\ &= \sigma\left(\sum_{i=1}^{n} w_{il} h_i + b_l \right) \end{aligned} \tag{8}$$

where σ is the following Sigmoid function 9 [48, 49].

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

The independence condition for a hidden neuron with a given visible level is calculated as follows [50, 51]:

$$p(H_i = 1|\boldsymbol{u}) = \sigma\left(\sum_{j=1}^{m} w_{ij} u_j + c_i \right) \tag{10}$$

$$p\left(V_j = 1|\boldsymbol{h}\right) = \sigma\left(\sum_{i=1}^{n} w_{ij} h_i + b_j \right) \tag{11}$$

The derivative of the logarithmic probability is equal to [35, 38, 42]:

$$\frac{\partial \ln \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{u})}{\partial \boldsymbol{\theta}} = -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) \frac{\partial E(\boldsymbol{u}, \boldsymbol{h})}{\partial \theta} + \sum_{\boldsymbol{u}, \boldsymbol{h}} p(\boldsymbol{u}, \boldsymbol{h}) \frac{\partial E(\boldsymbol{u}, \boldsymbol{h})}{\partial \theta} \tag{12}$$

Setting the parameter θ equal to the weights w_{ij} [52, 53]:

$$\begin{aligned}\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) \frac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial w_{ij}} &= \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) h_i u_j \\ &= \sum_{\boldsymbol{h}} \prod_{k=1}^{n} p(h_k|\boldsymbol{u}) h_i u_j \\ &= \sum_{h_i} \sum_{h_{-i}} p(h_i|\boldsymbol{u}) p(\boldsymbol{h}_{-\boldsymbol{i}}|\boldsymbol{u}) h_i u_j \\ &= \sum_{h_i} p(h_i|\boldsymbol{u}) h_i u_j \underbrace{\sum_{\boldsymbol{h}_{-i}} p(\boldsymbol{h}_{-\boldsymbol{i}}|\boldsymbol{u})}_{=1} \\ &= p(H_i = 1|\boldsymbol{u}) u_j \\ &= \sigma\left(\sum_{j=1}^{m} w_{ij} u_j + c_i\right) u_j \end{aligned} \tag{13}$$

The second term can be expressed as follows:

$$\begin{aligned}\sum_{u,h} \frac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial \theta} &= \sum_{\boldsymbol{u}} p(\boldsymbol{u}) \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) \frac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial \theta} \\ &= \sum_{\boldsymbol{h}} p(\boldsymbol{h}) \sum_{\boldsymbol{u}} p(\boldsymbol{u}|\boldsymbol{h}) \frac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial \theta}\end{aligned} \tag{14}$$

From the above relations, it is a sum over 2^N situations. The derivative of the logarithmic probability can be estimated as follows [49, 54]:

$$\begin{aligned}\frac{\partial \ln L(\theta|\boldsymbol{u})}{\partial w_{ij}} &= -\sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) \frac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial w_{ij}} + \sum_{\boldsymbol{u},\boldsymbol{h}} p(\boldsymbol{u},\boldsymbol{h}) \frac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial w_{ij}} \\ &= \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) h_i u_j - \sum_{\boldsymbol{u}} p(\boldsymbol{u}) \sum_{\boldsymbol{h}} p(\boldsymbol{h}|\boldsymbol{u}) h_i u_j \\ &= p(H_i = 1|\boldsymbol{u}) u_j - \sum_{\boldsymbol{u}} p(\boldsymbol{u}) p(H_i = 1|\boldsymbol{u}) u_j\end{aligned} \tag{15}$$

For a given set $S = \{u_1, \ldots, u_l\}$ [38, 47]:

$$\begin{aligned}\frac{1}{l}\sum_{\boldsymbol{u}\in S} \frac{\partial \ln L(\boldsymbol{\theta}|\boldsymbol{u})}{\partial w_{ij}} &= \frac{1}{l}\sum_{\boldsymbol{u}\in S} \left[\begin{array}{c} -\mathrm{E}_{p(\boldsymbol{h}|\boldsymbol{u})}\left[\dfrac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial w_{ij}}\right] \\ +\mathrm{E}_{p(\boldsymbol{h},\boldsymbol{u})}\left[\dfrac{\partial E(\boldsymbol{u},\boldsymbol{h})}{\partial w_{ij}}\right] \end{array}\right] \\ &= \frac{1}{l}\sum_{\boldsymbol{u}\in S} \left[\mathrm{E}_{p(\boldsymbol{h}|\boldsymbol{u})}\left[u_i h_j\right] - \mathrm{E}_{p(\boldsymbol{h},\boldsymbol{u})}\left[u_i h_j\right]\right] \\ &= \left\langle u_i h_j \right\rangle_{p(\boldsymbol{h}|\boldsymbol{u})q(\boldsymbol{u})} - \left\langle u_i h_j \right\rangle_{p(\boldsymbol{h},\boldsymbol{u})}\end{aligned} \tag{16}$$

The distribution q is the distribution represented by the data set and the above result can be written as:

$$\sum_{u\in S} \frac{\partial \ln L(\theta|u)}{\partial w_{ij}} \propto \left\langle u_i h_j \right\rangle_{\text{data}} - \left\langle u_i h_j \right\rangle_{\text{model}} \tag{17}$$

Setting parameter θ equal to the total of the remaining change parameters which are the weights b_j and c_i, the following expressions will be considered [36, 55]:

$$\frac{\partial \ln L(\boldsymbol{\theta}|\boldsymbol{u})}{\partial b_j} = u_j - \sum_{\boldsymbol{u}} p(\boldsymbol{u}) u_j \tag{18}$$

$$\frac{\partial \ln L(\boldsymbol{\theta}|\boldsymbol{u})}{\partial c_i} = p(H_{i=1}|\boldsymbol{u}) - \sum_{\boldsymbol{u}} p(\boldsymbol{u}) p(H_i = 1|\boldsymbol{u}) \tag{19}$$

It is possible to assess the above terms with Monte Carlo Markov chains. However, the representative subset of model distribution samples would require continuous sampling of the Markov chain for a long time until equilibrium was reached. This is not possible due to the computational cost and an additional approach is required. The usual approach employed when using RBMs is the Contrastive Divergence (CODI) method [56, 57]. CODI is the most common approximation technique for calculating the expected values in the derivative of the logarithmic probability, under the model's distribution [44].

In particular, instead of applying sequential Gibbs sampling steps to balance the neural network [58], CODI introduces a training example $u^{(0)}$ to the visible neurons. It then executes a Gibbs chain for k steps, acquiring a reconstruction $u^{(k)}$. For a large number of problems, even one step k is enough. The resulting approach for the derivative of the logarithmic probability to a change parameter θ, is:

$$\begin{aligned}\mathrm{CODI}_k\left(\boldsymbol{\theta}, \boldsymbol{u}^{(0)}\right) &= -\sum_{\boldsymbol{h}} p\left(\boldsymbol{h}|\boldsymbol{u}^{(0)}\right) \frac{\partial E\left(\boldsymbol{u}^{(0)}, h\right)}{\partial \boldsymbol{\theta}} \\ &\quad + \sum_{\boldsymbol{h}} p\left(\boldsymbol{h}|\boldsymbol{u}^{(k)}\right) \frac{\partial E\left(\boldsymbol{u}^{(k)}, \boldsymbol{h}\right)}{\partial \boldsymbol{\theta}}\end{aligned} \tag{20}$$

A proper approach requires the implementation of CODI across the entire data set S, for every step of the procedure. However, the best way is to apply CODI to a subset of data $\widehat{S} \in S$, which could be a representative and extended mini batch, especially in cases of big data sets. In any case, CODI is an approximate technique. The resulting sample may not be related to the equilibrium distribution of the model, so the approach is biased. Accordingly, the mixing rate of the Markov chain is a measure of how quickly it leads to equilibrium distribution.

It is described by the transition probabilities and is one of the factors, in conjunction with the individual execution steps, that influences the approach error. Also, the order of magnitude of the change parameters affects the mixing rate. This is evident by the expressions of the conditional probabilities in terms of the sigmoid function. High values of change parameters correspond to values close to zero for the conditional probabilities, and the Markovian chain is evolving very slowly over time.

Variational Inference (VAI) [59] is an alternative for cases where Markovian models cannot perform effectively. VAI models are interested in finding the parameter distribution of the model, using faster and scalable methods, which are suitable also for big datasets. Specifically, the models in question, are searching the exact and detailed distribution of a strict approach to the actual posterior distribution, following the following relation [22, 60]:

$$\arg\min_{\phi\in\Phi} \mathrm{KL}(q(\theta;\phi) \parallel p(\theta|\mathrm{data})) \tag{21}$$

where $q(\theta;\phi)$ is the variational density. It is a parametric density which is parametrized by ϕ, which should be easy to sample and evaluate. The proposed anomaly detector refers to the change between two points which is compared to a specified threshold [52]:

$$\Delta_t = y_t - y_{t-1} \tag{22}$$

The threshold is determined based on the Mahalanobis distance and specifically [49, 61]:

$$d(\mathrm{Mahalanobis}; \boldsymbol{x}, \boldsymbol{\mu}) = \left[(x-\boldsymbol{\mu})^{T*}\Sigma^{-1} * (x-\boldsymbol{\mu})\right]^{0.5} \tag{23}$$

As soon as the Mahalanobis distance is estimated, it becomes feasible to estimate the probability $P(X)$ of a sample's appearance as follows:

$$\begin{gathered} P(X;\mu,\Sigma) = \left\{1/\left((2\pi)^{n/2} * |\Sigma|^{1/2}\right)\right\}^{*} \\ \exp(-d(\mathrm{Mahalanobis}; X, \mu)) \end{gathered} \tag{24}$$

where $|\Sigma|$ is the covariance matrix Σ [43, 49].

Using the threshold value ε and for all values $P(X) < \varepsilon$ we have an anomaly in the dataset, which is estimated dynamically as follows:

$$\varepsilon = \frac{\mathrm{X} - \mathrm{X}_{\min}}{\mathrm{X}_{\max} - \mathrm{X}_{\min}} \tag{25}$$

Defining the threshold ε, the relation 26 changes as follows:

$$j_t = \begin{cases} 1 & \Delta_t \geq \varepsilon \\ -1 & \Delta_t < \varepsilon \end{cases} \tag{26}$$

Supposing that the Δ_s are following a Normal Distribution, the output $p_{j,t}$ is related to the cumulative distribution function of the normal distribution as follows:

$$p_{j,t} = \Phi(\Delta_t) \tag{27}$$

The Kullback–Leibler (KL) [56, 62] deviation is the method of measuring the distance between two densities. Therefore, in the Variational Inference methods an attempt is made to minimize the dissimilarity between the classification approach and the actual posterior distribution. The exact form of KL between two densities is calculated by the following formula [62]:

$$\mathrm{KL}(q(\theta) \parallel p(\theta)) = E_q[\log q(\theta) - \log p(\theta)] \tag{28}$$

where the expectation of solutions is directly related to the variational density which is related to the variable θ. The immediate minimization of the KL deviation is a difficult task.

Therefore, the solutions are related to the maximization of the Evidence Lower Bound (ELBO), which is equivalent to minimizing the KL deviation, so that [40, 63]:

$$\mathrm{ELBO}(\phi) = E_{q(\theta;\phi)}[\log p(\mathrm{data}\backslash\theta) + \log p(\theta) - \log q(\theta;\phi)] \tag{29}$$

$$\mathrm{ELBO}(\phi) = E_{q(\theta;\phi)}[\log p(data,\theta) - \log q(\theta;\phi)] \tag{30}$$

Therefore, the goal becomes to find solutions for

$$\phi^* = \arg\max_{\phi} \mathrm{ELBO}(\phi) \tag{31}$$

Solutions that have been proposed for this process, concern the attempt to update the parameter ϕ in variational distribution successively, until certain convergence criteria are met. This requires detailed sources of updates, which can be time consuming at best and impossible in some cases.

ADVI [64] is an alternative approach for the maximization of ELBO. It is a gradient-based procedure, and it requires iterative optimization to detect $\phi*$. Nevertheless, it uses a stochastic gradient descent method that requires the calculation of ELBO derivatives in relation to the parameters. Assuming that all model parameters are continuous, in ADVI, ELBO is rewritten as follows [39, 59, 64]:

$$\begin{aligned} \mathrm{ELBO}(\phi) = E_{q(\zeta;\phi)}\big[&\log p\big(\mathrm{data}, T^{-1}(\zeta)\big) \\ &+ \log|\det J_{T^{-1}}(\zeta)| - \log q(\zeta;\phi)\big] \end{aligned} \tag{32}$$

where, the $T : \mathrm{support}(\theta) \rightarrow \mathbb{R}^{\dim(\theta)}$ is a function that converts the θ to ζ, and $\zeta \in \mathbb{R}^{\dim(\theta)}$. ELBO estimation requires sampling of values from the variables of the normal distributions and the evaluation of the expression through the above expectation. It should be noted that authors of ADVI demonstrate in practice that one sample is sufficient for this evaluation. ELBO maximization requires the ELBO gradient with respect to variational parameters as follows [27, 39]:

$$\begin{aligned} \nabla_\phi \mathrm{ELBO}(\phi) = \nabla_\phi E_{q(\zeta;\phi)}\big[&\log p\big(\mathrm{data}, T^{-1}(\zeta)\big) \\ &+ \log\left|\det J_{T^{-1}(\zeta)}\right| - \log q(\zeta;\phi)\big] \end{aligned} \tag{33}$$

In order to push the gradient inside the expectation, we must first design a standard Normal random variate, and then multiply the random variate by the variational standard deviation and variational mean as in Eq. 34 as follows [39, 43]:

$$\begin{aligned} \nabla_\phi \mathrm{ELBO}(\phi) \approx \log p\left(\mathrm{data}, T^{-1}\left(\tilde{\zeta}\right)\right) + \log\left|\det J_{T^{-1}}\left(\tilde{\zeta}\right)\right| \\ - \log q\left(\tilde{\zeta};\phi\right) \end{aligned} \tag{34}$$

where $\tilde{\zeta} = \mu + z\sigma$, and z is a draw from a standard Normal and (μ, σ) are the variational mean and standard deviations which can be vectors when z is multivariate standard Normal. All that remains is to perform some kind of inclination to obtain solutions for ϕ which is legitimate and computationally feasible. It should be noted that this modeling does not require the calculation or use of Jacobian, nor does it record a correlation between parameters. It also does not record a correlation between parameters, which makes it very affordable and efficient without high computational cost.

In conclusion, the proposed methodology automatically transforms the hidden variables into a real coordinate space. In this space, it chooses an approach which is a function of the transformed variables and optimizes its parameters with stochastic gradient ascent methods. Thus, it can be applied to solve a wide range of models without having to find a detailed solution for each of them. This makes it computationally accessible.

4 Dataset

The *HIL-based Augmented Industrial Control System* (ICS) (HAI) [65] security dataset was used to test the proposed system. This is a fully realistic data set, collected through the Hardware-In-the-Loop (HIL) simulation process. Specifically, it simulated *steam-turbine power generation* and *pumped-storage hydropower generation* mechanisms [66].

HAI is considered one of the most valid and realistic data sets. It includes both normal and abnormal behaviors of the systems in question. Its sole purpose is the detection of abnormalities in ICS. The normal data set was collected from the actual operation of the ICS systems. The abnormal set was collected based on various attack scenarios in the six control loops and in the three different types of industrial equipment devices [67].

Each control loop refers to a system that includes all software functions required to measure and adjust the variable that controls a process. The simulation considers the boiler, the turbine, the water treatment component and the HIL simulator. The boiler process is actually, the transfer of heat from water to water, under conditions of low pressure and moderate temperature. The turbine process includes a rotor test kit aiming to simulate the behavior of a real rotating machine.

In this research, the boiler and turbine processes were interconnected with the HIL simulator to ensure synchronization with the rotation speed of the steam generator. The water treatment process involved pumping water into the upper tank and subsequent release into the lower tank, based on a hydroelectric generating model that uses a storage pump. The three procedures were controlled by three different types of controllers, as shown in Fig. 1 [28].

Data collection was based on the use of SCADA systems which typically collect data elements called points (or tags). Each point represents an individual variable measured or controlled by each system. All scenarios are configured based on the four variables of the feedback loop, namely Setpoints (SP), Process Variables (PV), Control Variables (CV), and Control Parameters (CP).

During normal conditions, it is assumed that the control module is operated normally via the HMI and that the simulator variables associated with the output of the HIL change. The operator monitors the values given by the current sensor displayed on the HMI and adjusts the SPs of the various control devices related to the operation of the system.

An HMI task scheduler was used to periodically initialize the SPs and HIL simulator variables at random or predefined values, which were within the normal range, in order to simulate a *Benign* scenario. The normal SP value limits at which the whole procedure was constant were determined by performing an experimental change of each SP value [67].

The four controls (P1-PC, P1-LC, P1-FC and P1-TC) and two simulation models (steam turbine power generator and pump hydroelectric generator) operated automatically several times on a daily basis. They started with a random delay and reached a random value or default value, within the normal operating range. All SP values were recorded to learn the capabilities of the system. Respectively, based on the four variables of the feedback control loop presented above, the *attack scenarios* are simulated as shown in Fig. 2.

Abnormal behavior occurs either when some of the parameters are not within the normal range, or when they are facing unexpected situations due to attacks,

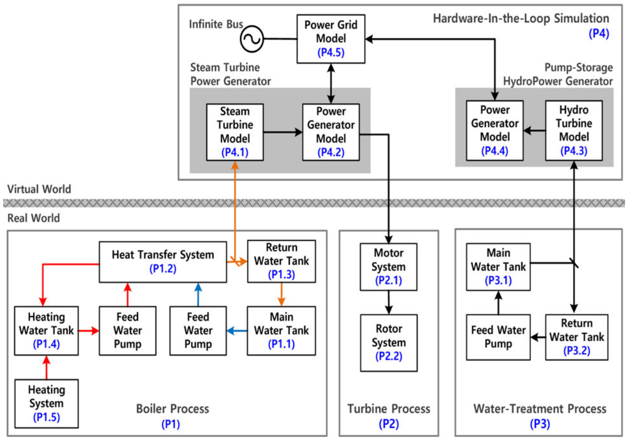

Fig. 1 Process flow diagram

malfunctions and failures. It should be noted that the attack scenarios have been implemented taking into account the attack target, the attack time and the method for each feedback control loop. Overall, the HAI 2.0 dataset that was used, has 79 measurement channels for receiving data from sensors, actuators and control devices, representing the current state of the system, taking one measurement every second.

Totally, 964,804 s of cyber-physical systems are considered, including measurements from 5 cyber-attacks [66]. The aim of the proposed system is to detect the structure of the input data and to discover their hidden patterns, in order to detect the anomalies that are expressed as cyber-attacks. This should be done without providing any experience to the learning algorithm, and without the need to find an analytical solution.

5 Results and discussion

Thorough preprocessing of the dataset was performed to demonstrate its proper use. This process is necessary as the original data often suffer from various kinds of problems, such as conflicting information, coding inconsistencies, noise and extremes. Also, it is often necessary to address specific problems that require data transformation, such as the discretization, the normalization, the reduction of the dimensionality or the selection of the most appropriate characteristics.

Initially, an indicative statistical analysis of the dataset was performed. The main objective of the above statistical procedure was the analysis and interpretation of the used data, with the ultimate goal of drawing safe conclusions that can lead to correct decisions. Specifically, Table 1 shows the probability for each sample to belong to a specific subset, when the sample space consists of discrete random variables, where the distribution can be determined by a cumulative probability function.

Also, for the clear and distinct determination of the variance, the graphs of the statistical frequencies of each feature's values are presented in histograms. The height of each region is equal to the ratio of the frequency to the range of values represented by the rectangle. An illustrative presentation of the features is shown in Fig. 3.

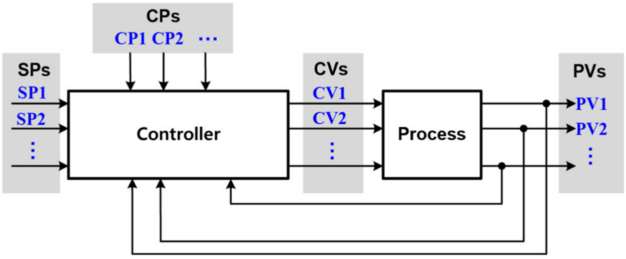

Fig. 2 Attack model based on a process control loop

Correlation analysis has been performed. It can facilitate the comparison of several quantitative variables in order to identify patterns, similarities, complexes, as well as positive, negative or neutral relationships between data. Correlations are useful because they can indicate a predictive relationship that can be exploited in practice, although statistical dependence is not sufficient to prove the presence of a causal relationship (that is, the correlation does not imply causality). A Principal Component Analysis (PCA) test was then performed to detect data covariance and to decide if parameter reduction is required.

As can be seen from the scree plot of Fig. 4 the first two principal components retain slightly less than 41% of the statistical data from the original data, so no parameter reduction is required.

Respectively, the following Fig. 5 identifies the contribution of each variable to the predictive capacity of the set.

Data pre-processing ensures the quality of the used data, in order to avoid any potential bias regarding the experimental procedure. Unsupervised learning was employed for the determination of the clusters in order to reach a reliable optimal model. Moreover, extensive comparisons were made between the introduced algorithm and the following competing clustering methods (SOM, *K*-Means, DBSCAN, Gaussian Mixtures (GaMix) and Spectral Clustering (SpClu)). Two to ten cluster centers were used during experimentations.

For example, in the case of the *k*-means algorithm, which is sensitive to the initial positions of the centers of the clusters, ten initial configurations were created and then the Sums of Squares were averaged. The optimal scenario is when the minimum Sum of Squares is observed within a cluster (how tight each cluster is).

The solutions were evaluated in terms of their homogeneity, according to the *Coefficient of Variation* (CV) [38, 49]. CV is the index of dispersion, and it expresses the homogeneity of a set of measurements of a random quantitative variable and the accuracy of an experimental design. It is estimated by the following relation:

$$\mathrm{CV} = 100 * {}^{S}\!/_{\overline{Y}} \tag{35}$$

where S standard deviation and $\overline{Y}$ arithmetic mean of the samples. Values between $0.00 < \mathrm{CV} \leq 0.25$ declare a high level of homogeneity, $0.25 < \mathrm{CV} \leq 0.40$ median level and $\mathrm{CV} > 0.40$ low level. The results obtained for each method are presented in Table 1.

To confirm the optimal number of clusters, the "Elbow" method was used. This approach calculates the sum of squares for each proposed number of clusters [49]. The

Table 1 Statistical analysis of the data set

	Proposed	SOM	*k*-Means	DBSCAN	GaMix	SpClu
Clusters *k*	2	2	2	2	3	4
Within_ss	0.11	0.22	0.17	0.20	0.19	0.16
Between_ss	0.89	0.78	0.87	0.75	0.77	0.64
CV	0.12	0.27	0.19	0.29	0.31	0.42

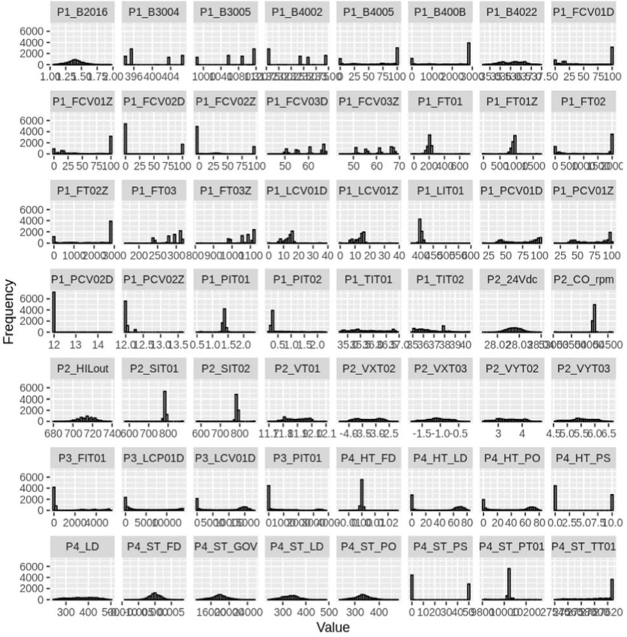

Fig. 3 Historgram for each channel

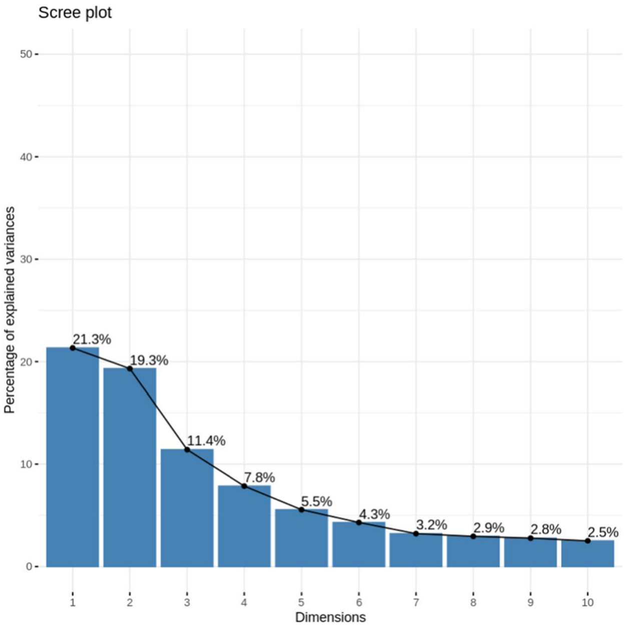

Fig. 4 Scree plot

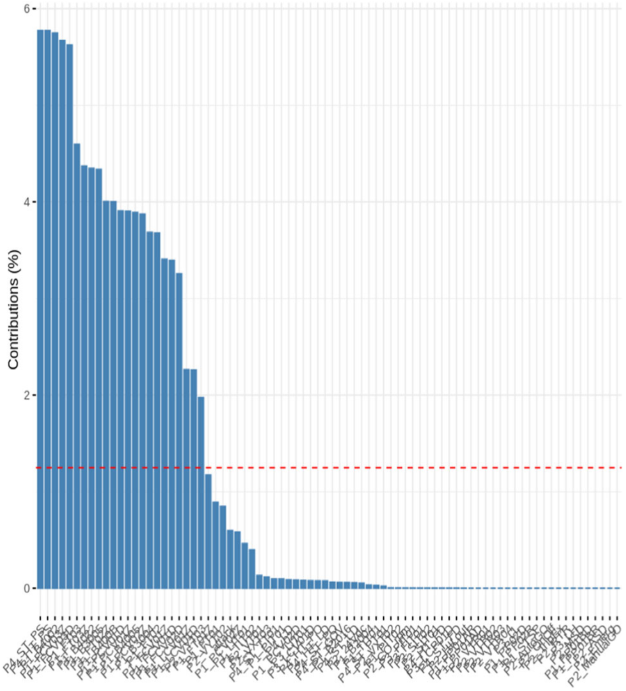

Fig. 5 Contribution of each variable

optimal number is obtained in the case of the abrupt change of inclination as shown in Fig. 6.

The "Silhouette" method was also used which calculates the average silhouette of the observations for different number of clusters. It then calculates the variance within the clusters for each case and compares them with their expected values, under a zero-reference distribution of the data [49]. The estimation of the optimal number of clusters is the value that maximizes the mean silhouette in a range of possible values, which means that the clustering structure is far from the random uniform distribution of points. The "Silhouette" is presented in Fig. 7.

The following Figs. 8, 9 and 10 present in detail the various versions of the instability indices of the clusters related to the proposed model. The size of each node corresponds to the number of samples in each cluster and the arrows are colored according to the number of samples received by each cluster. Transparent arrows are called inbound node ratios and they show how samples from one group end up in another. Also, this index shows the quality of the cluster analysis [49, 68].

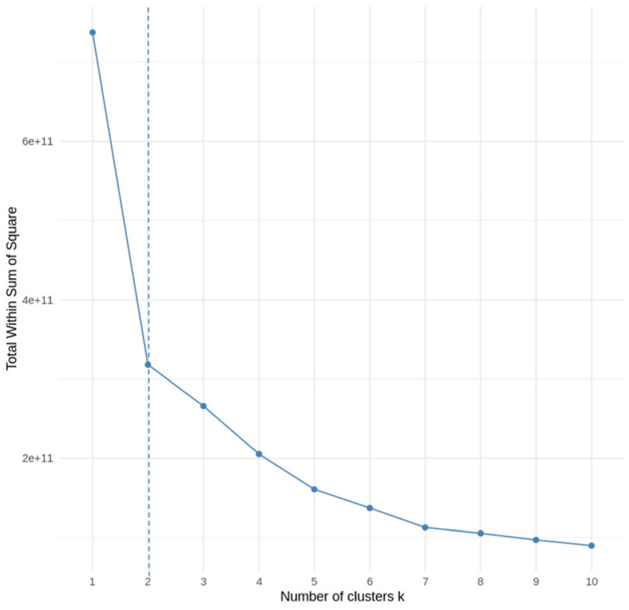

Fig. 6 Elbow method of the proposed algorithm

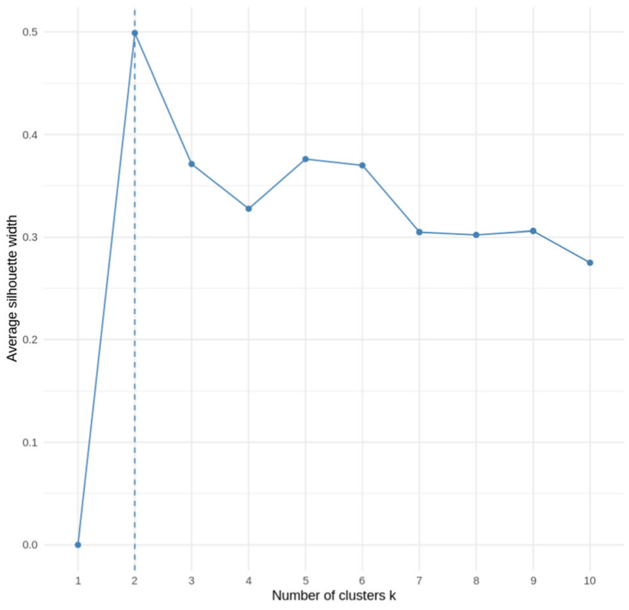

Fig. 7 The silhouette-plot

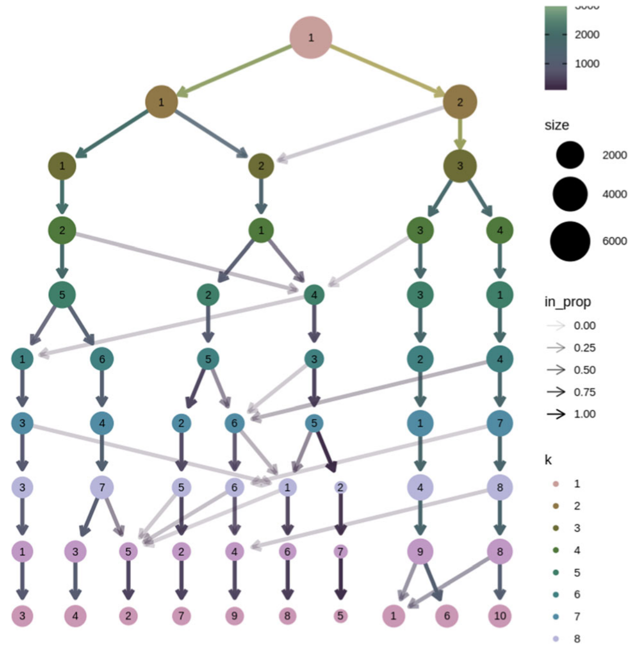

Fig. 8 Instability plot

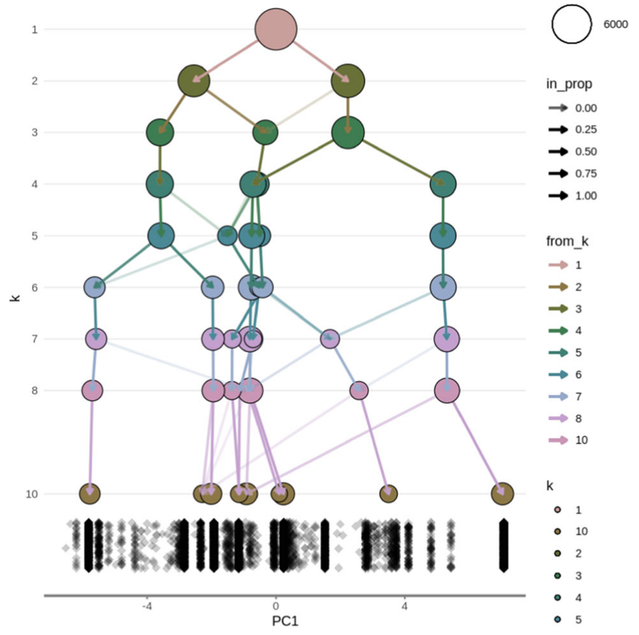

Fig. 9 Instability plot A

The overall results of the anomaly detection process, take into account the most important performance metrics for the investigation of binary cases and they are presented in Table 2 [68]. It should be noted that the comparison does not include the *GaMix* and *SpClu* algorithms, as their results were completely disappointing. The data set management for the above algorithms took into account 3 and 4 clusters, respectively, which did not offer a reliable model for the problem under consideration.

Table 2 clearly shows the superiority of the proposed ADVI RBM algorithm, which excels in all metrics, while the performance error remains very low compared to the other approaches. Specifically, the accuracy of the ADVI RBM, exceeds by 2.7% the second-best method, while the recorded error is significantly smaller. Even in terms of the time it took to cluster the data, with the exception of the k-means algorithm, which is clearly 170 s faster, the other algorithms took about the same amount of time to run. In general, this system of *fully automated learning without training* is very promising and it can significantly improve

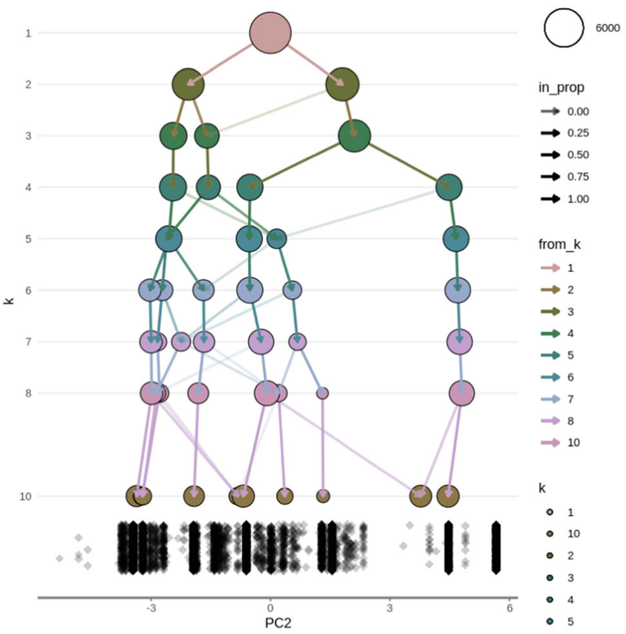

Fig. 10 Instability plot B

the safety of industrial infrastructure. The following Table 3 presents the initial parameters of compared algorithms used.

It is very important that it works very effectively on unlabeled data, without having to define the number of clusters in the system's parameters (as it happens for the k-means) and also without the need to set a threshold value. These features strongly reinforce the belief that the proposed system can effectively model real-world data. Another important advantage is the fact that the RBM, as a logarithmic approach, decides for itself which features are relevant and how to best combine them to form patterns. This feature is clearly demonstrated by the very high-performance results it has achieved, as well as its ability to generalize to new unknown situations.

Also, in the proposed RBM, the hidden modules are conditionally independent, so we can quickly get an unbiased sample from the rear distribution when given a data vector. This process has been significantly enhanced with the use of ADVI, so that the final system has the ability to detect anomalies without the need for prior system training and without the need to find a detailed solution. This makes it computationally accessible and relatively fast (given that times with other algorithms are almost similar). Obviously, the convergence response time of the algorithm is something that needs to be improved, but in any case, this process is recorded as positive.

6 Conclusions and future work

With the installation of sensors and the use of smart applications, CoMo offers accurate, valid and timely data, aiming at the proper management and maintenance of industrial equipment. CoMo security event analysis for real-time anomaly detection is an active approach to security event management. Nonetheless, given that no tool can accurately predict the future state of industrial equipment, intelligent anomaly detection systems prove to be particularly useful and reliable, as they are able to give a clear picture of how these systems work.

This paper presents the innovative ADVI RBM system that performs intelligent and unattended automatic learning, for the detection of abnormalities in industrial control. It is an improved form of RBM, whose functionality is enhanced by the advanced ADVI technique for the precise determination of a posterior distribution without the need of prior training, and without the need to find a detailed solution. Specifically, to calculate the conditional probabilities of a given hidden or visible neuron, the set of visible variables has exponential complexity since it is a sum over 2^N situations. Consequently, the calculated quantity cannot be solved even if the internal sum is factorized in any existing way. The proposed approach does not require the calculation or use of Jacobian matrix, nor does it record a correlation between parameters. The latter

Table 2 Performance comparison

	ACC	PRE	REC	F-SC	ROC	RMSE	Time/s
Proposed ADVI RBM	0.989	0.990	0.990	0.990	0.995	0.1477	548
RBM	0.953	0.955	0.950	0.955	0.955	0.1901	984
VAE	0.971	0.970	0.970	0.970	0.975	0.1742	611
LSTM	0.959	0.960	0.960	0.960	0.960	0.1874	706
SOM	0.948	0.950	0.950	0.950	0.970	0.2468	583
k-Means	0.962	0.960	0.965	0.965	0.965	0.1993	374
DBSCAN	0.937	0.935	0.935	0.940	0.960	0.2815	537

ACC accuracy, *PRE* precision, *REC* recall, *F-SC* F-score, *RO* Receiver operating characteristic curve, *RMSE* root mean squared error, *VAE* variational autoencoder, *LSTM* Long short-term memory network *SOM* self organizing map neural network

Table 3 Parameters of compared algorithms

Algorithm	Parameters
RBM	training_algorithm = Stochastic_Gradient_Descent, k = 0.05, batch_size = 100, learning_rate = 0.1, θ_G = 0.005, θ_A = 0.3, regularization = L2
VAE	encoder_input = 28 × 28 × 1, conv2d_1 = 14 × 14 × 8, batch_normaliz_1 = 14 × 14 × 8, conv2d_2 = 7 × 7 × 16, batch_normaliz_2 = 7 × 7 × 16, dense_1 = 20, batch_normaliz_3 = 20, latent_mu = 2, latent_sigma = 2, z = 2
LSTM	activation = tanh, recurrent_activation = sigmoid, use_bias = True, kernel_initializer = glorot_uniform, bias_initializer = zeros, recurrent_initializer = orthogonal, unit_forget_bias = True, dropout = 0.1, recurrent_dropout = 0.1
SOM	n_neurons = 10 × 10, learning_rate = 0.9, radius = 20, distance_metric = Euclidean, normalization = True, Initialization = random_sample, n_iterations = 1000
k-Means	n_clusters = 8, init = k-means + + , n_init = 10, max_iter = 300, tol = 0.0001, algorithm = elkan
DBSCAN	eps = 0.5, min_samples = 5, metric between instances in a feature array = Euclidean, algorithm = NearestNeighbors, leaf_size = 30, distance between points = Minkowski

makes it very affordable and efficient due to its low computational cost. Thus, the proposed methodology automatically transforms the hidden variables into real coordinate space. In this space, it chooses an approach that is a function of the transformed variables and optimizes its parameters with stochastic gradient ascent methods. Thus, it can be applied to solve a wide range of models without having to find a detailed solution for each of them. This makes it computationally accessible. This process significantly evolves the way RBMs operate, as computational complexity and corresponding computational resource requirements are marginalized.

The usual approach employed when using RBMs is the CODI method. CODI is the most common approximation technique for calculating the expected values in the derivative of the logarithmic probability, under the model's distribution. The representative subset of model distribution samples would require continuous sampling of the Markov chain for a long time until equilibrium was reached. This is not possible for real-time applications due to the computational cost, and an alternative approach is required. As shown in Sect. 2, using Variational Distribution as a normal distribution of multiple variables, it is possible to correlate the parameters to solve dynamic problems at an affordable computational cost.

The high generalizability, as well as the convergence stability of the proposed methodology, prove that it is capable of performing even when the nature of the anomaly is new and therefore unknown. Generalization refers to the model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model. The performance of the proposed algorithm is based on the performance metrics (Accuracy, Precision, Recall, F-Score, and ROC curves) that show values of estimates of the generalization error through the learning process. Table 2 clearly shows the superiority of the proposed ADVI RBM algorithm, which excels in all metrics, while the generalization error remains very low compared to the other approaches.

This work proves that using Variational Distribution as a normal distribution of multiple variables, it is possible to correlate the parameters of a simple model such as the one proposed, to solve dynamic problems at an affordable computational cost.

Future research toward the improvement of the proposed system, will include the process of finding convergence solutions for in shorter times. Another optimization approach will be used to enhance the anomaly detection system with more advanced techniques, capable of estimating hyperparameter values and of using meta-learning methods. The structure of the algorithm should be extensively studied and completed with data transformation techniques, so that it can locate the optimal representations of the data and extract only useful information.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Boubekeur M (2017) Industrial applications for cyber-physical systems. In: 2017 first international conference on embedded distributed systems (EDiS), pp 59–59. https://doi.org/10.1109/EDIS.2017.8284020
2. Banafa A (2018) 2 The Industrial Internet of Things (IIoT): challenges, requirements and benefits. In: Secure and smart internet of things (IoT): using blockchain and AI, river publishers, pp 7–12. [Online]. https://ieeexplore.ieee.org/document/9226906. Accessed 19 Jan 2021
3. Radanliev P et al (2020) Cyber risk at the edge: current and future trends on cyber risk analytics and artificial intelligence in the

industrial internet of things and industry 4.0 supply chains. [Online]. https://www.preprints.org/manuscript/201903.0123/v2. Accessed 19 Jan 2021

4. Demertzis K, Iliadis L, Tziritas N, Kikiras P (2020) Anomaly detection via blockchained deep learning smart contracts in industry 4.0. Neural Comput Appl 32(23):17361–17378. https://doi.org/10.1007/s00521-020-05189-8
5. Chalapathy R, Chawla S (2019) Deep learning for anomaly detection: a survey. [Online]. Preprint http://arxiv.org/abs/1901.03407. Accessed 08 Sep 2021
6. Tsiknas K, Taketzis D, Demertzis K, Skianis C (2021) Cyber threats to industrial IoT: a survey on attacks and countermeasures. IoT. https://doi.org/10.3390/iot2010009
7. Deorankar AV, Thakare SS (2020) Survey on anomaly detection of (IoT)- Internet of Things cyberattacks using machine learning. In: 2020 fourth international conference on computing methodologies and communication (ICCMC), pp 115–117. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00023
8. Pang G, Shen C, Cao L, van den Hengel A (2021) Deep learning for anomaly detection: a review. ACM Comput Surv 54(2):1–38. https://doi.org/10.1145/3439950
9. Elmrabit N, Zhou F, Li F, Zhou H (2020) Evaluation of machine learning algorithms for anomaly detection. In: 2020 international conference on cyber security and protection of digital services (cyber security), pp 1–8. https://doi.org/10.1109/CyberSecurity49315.2020.9138871
10. Al Jallad K, Aljnidi M, Desouki MS (2020) Anomaly detection optimization using big data and deep learning to reduce false-positive. J Big Data 7(1):68. https://doi.org/10.1186/s40537-020-00346-1
11. Falco G, Caldera C, Shrobe H (2018) IIoT cybersecurity risk modeling for SCADA systems. IEEE Internet Things J 5(6):4486–4495. https://doi.org/10.1109/JIOT.2018.2822842
12. Gawlikowski J, et al. (2021) A Survey of uncertainty in deep neural Networks. [Online]. Preprint http://arxiv.org/abs/2107.03342. Accessed 06 Nov 2021
13. Demertzis K, Iliadis L, Kikiras P (2021) A lipschitz-shapley explainable defense methodology against adversarial attacks. In: Artificial intelligence applications and innovations. AIAI 2021 IFIP WG 12.5 international workshops, Cham, pp 211–227. https://doi.org/10.1007/978-3-030-79157-5_18
14. Samrin R, Vasumathi D (2017) Review on anomaly based network intrusion detection system. In: 2017 international conference on electrical, electronics, communication, computer, and optimization techniques (ICEEECCOT), pp 141–147. https://doi.org/10.1109/ICEEECCOT.2017.8284655
15. Dias R, Alexandre L, Mauricio F, Poggi M (2020) Toward an efficient real-time anomaly detection system for cloud datacenters. In: 2020 IFIP networking conference (networking), pp 529–533
16. Feng T, Du Z, Sun Y, Wei J, Bi J, Liu J (2017) Real-time anomaly detection of short-time-scale GWAC survey light curves. In: 2017 IEEE international congress on big data (bigdata congress), pp 224–231. https://doi.org/10.1109/BigDataCongress.2017.38
17. Demertzis K, Iliadis L, Spartalis S (2017) A spiking one-class anomaly detection framework for cyber-security on industrial control systems. In: Engineering applications of neural networks, Cham, pp 122–134. https://doi.org/10.1007/978-3-319-65172-9_11
18. Kumar M, Mathur R (2014) Unsupervised outlier detection technique for intrusion detection in cloud computing. In: International conference for convergence for technology-2014, pp 1–4. https://doi.org/10.1109/I2CT.2014.7092027
19. MohanaPriya P, Shalinie SM (2017) Restricted Boltzmann machine based detection system for ddos attack in software defined networks. In: 2017 fourth international conference on signal processing, communication and networking (ICSCN), pp 1–6. https://doi.org/10.1109/ICSCN.2017.8085731
20. Xu X, Li J, Yang Y, Shen F (2021) Toward effective intrusion detection using log-cosh conditional variational autoencoder. IEEE Internet Things J 8(8):6187–6196. https://doi.org/10.1109/JIOT.2020.3034621
21. Zavrak S, İskefiyeli M (2020) Anomaly-based intrusion detection from network flow features using variational autoencoder. IEEE Access 8:108346–108358. https://doi.org/10.1109/ACCESS.2020.3001350
22. Doersch C (2021) Tutorial on variational autoencoders, [Online]. Preprint http://arxiv.org/abs/1606.05908. Accessed 09 Sep 2021
23. Kim J, Sim A, Kim J, Wu K (2020) Botnet detection using recurrent variational autoencoder. In: GLOBECOM 2020-2020 IEEE global communications conference, pp 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9348169
24. Gu Z, Yang Y (2021) Detecting malicious model updates from federated learning on conditional variational autoencoder. In: 2021 IEEE international parallel and distributed processing symposium (IPDPS), pp 671–680. https://doi.org/10.1109/IPDPS49936.2021.00075
25. Farooq MJ, Zhu Q (2019) IoT supply chain security: overview, challenges, and the road ahead, [Online]. Preprint http://arxiv.org/abs/1908.07828. Accessed 19 Jan 2021
26. Li H, Zhou S, Yuan W, Li J, Leung H (2020) Adversarial-Example attacks toward android malware detection system. IEEE Syst J 14(1):653–656. https://doi.org/10.1109/JSYST.2019.2906120
27. Liu Y, Mao S, Mei X, Yang T, Zhao X (2019) Sensitivity of adversarial perturbation in fast gradient sign method. In: 2019 IEEE symposium series on computational intelligence (SSCI), pp 433–436. https://doi.org/10.1109/SSCI44817.2019.9002856
28. Hwang W-S, Yun J-H, Kim J, Kim HC (2019) Time-series aware precision and recall for anomaly detection: considering variety of detection result and addressing ambiguous labeling. In: Proceedings of the 28th ACM international conference on information and knowledge management, New York, pp 2241–2244. https://doi.org/10.1145/3357384.3358118
29. Tang P, Wang W, Lou J, Xiong L (2021) Generating adversarial examples with distance constrained adversarial imitation networks. IEEE Trans Dependable Secure Comput. https://doi.org/10.1109/TDSC.2021.3123586
30. Yu P, Song K, Lu J (2018) Generating adversarial examples with conditional generative adversarial net. In: 2018 24th international conference on pattern recognition (ICPR), pp 676–681. https://doi.org/10.1109/ICPR.2018.8545152
31. Han K, Li Y, Xia B (2021) A cascade model-aware generative adversarial example detection method. Tsinghua Sci Technol 26(6):800–812. https://doi.org/10.26599/TST.2020.9010038
32. Hwang U, Park J, Jang H, Yoon S, Cho NI (2019) PuVAE: a variational autoencoder to purify adversarial examples. IEEE Access 7:126582–126593. https://doi.org/10.1109/ACCESS.2019.2939352
33. Austin J, Kennedy J, Lees K (1995) A neural architecture for fast rule matching. In: Proceedings 1995 second new zealand international two-stream conference on artificial neural networks and expert systems, pp 255–260. https://doi.org/10.1109/ANNES.1995.499484
34. Xing L, Demertzis K, Yang J (2020) Identifying data streams anomalies by evolving spiking restricted Boltzmann machines. Neural Comput Appl 32(11):6699–6713. https://doi.org/10.1007/s00521-019-04288-5
35. Mrad AB, Delcroix V, Piechowiak S, Leicester P, Abid M (2015) An explication of uncertain evidence in bayesian networks:

likelihood evidence and probabilistic evidence. Appl Intell 43(4):802–824. https://doi.org/10.1007/s10489-015-0678-6

36. De La Rosa E, Yu W (2015) Restricted Boltzmann machine for nonlinear system modeling. In: 2015 IEEE 14th International conference on machine learning and applications (ICMLA), pp 443–446. https://doi.org/10.1109/ICMLA.2015.24
37. Marlin B, Swersky K, Chen B, Freitas N (2010) Inductive principles for restricted Boltzmann machine learning. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 509–516. [Online]. https://proceedings.mlr.press/v9/marlin10a.html. Accessed 10 Sep 2021
38. van de Schoot R et al (2021) Bayesian statistics and modelling. Nat Rev Methods Primer. https://doi.org/10.1038/s43586-020-00001-2
39. Kingma DP, Welling M (2014) Auto-encoding variational bayes. [Online]. Preprint http://arxiv.org/abs/1312.6114. Accessed 09 Sep 2021
40. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828. https://doi.org/10.1109/TPAMI.2013.50
41. Ahmadlou M, Adeli H (2010) Enhanced probabilistic neural network with local decision circles: a robust classifier. Integr Comput Aided Eng 17(3):197–210
42. Tversky A, Kahneman D (1973) Availability: a heuristic for judging frequency and probability. Cognit Psychol 5(2):207–232. https://doi.org/10.1016/0010-0285(73)90033-9
43. Hastie T, Tibshirani R, Friedman J (2016) The elements of statistical learning: data mining, inference, and prediction, second edition, 2nd edn. Springer, New York
44. Barshan E, Fieguth P (2014) Scalable learning for restricted Boltzmann machines. In: 2014 IEEE international conference on image processing (ICIP), pp 2754–2758. https://doi.org/10.1109/ICIP.2014.7025557
45. Chen L, Zou W (2018) Improvement of restricted Boltzmann machine by sparse representation based on lorentz function. In: 2018 7th international congress on advanced applied informatics (IIAI-AAI), pp 968–969. https://doi.org/10.1109/IIAI-AAI.2018.00205
46. Yu J, Gwak J, Lee S, Jeon M (2015) An incremental learning approach for restricted Boltzmann machines. In 2015 International conference on control, automation and information sciences (ICCAIS), pp 113–117. https://doi.org/10.1109/ICCAIS.2015.7338643
47. He T, Luo X, Liu Z (2017) A probabilistic indoor localization algorithm based on Restricted Boltzmann Machine. In 2017 IEEE 2nd advanced information technology, electronic and automation control conference (IAEAC), pp 1364–1368. https://doi.org/10.1109/IAEAC.2017.8054237
48. Alam KMdR, Siddique N, Adeli H (2020) A dynamic ensemble learning algorithm for neural networks. Neural Comput Appl 32(12):8675–8690. https://doi.org/10.1007/s00521-019-04359-7
49. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
50. Fortunato S (2010) Community detection in graphs. Phys Rep 486(3–5):75–174. https://doi.org/10.1016/j.physrep.2009.11.002
51. Lacasa L, Luque B, Ballesteros F, Luque J, Nuño JC (2008) From time series to complex networks: the visibility graph. Proc Natl Acad Sci 105(13):4972–4975. https://doi.org/10.1073/pnas.0709247105
52. Pollock JL (2007) Reasoning and probability. Law Probab Risk 6(1–4):43–58. https://doi.org/10.1093/lpr/mgm014
53. Hamer D (2012) Probability, anti-resilience, and the weight of expectation. Law Probab Risk 11(2–3):135–158. https://doi.org/10.1093/lpr/mgs004
54. Bengio Y, Courville A, Vincent P (2014) Representation Learning: a review and new perspectives. [Online]. Preprint http://arxiv.org/abs/1206.5538. Accessed 09 Sep 2021
55. Kerschke P, Hoos HH, Neumann F, Trautmann H (2019) Automated algorithm selection: survey and perspectives. Evol Comput 27(1):3–45. https://doi.org/10.1162/evco_a_00242
56. Calvayrac F (2015) Kullback-Leibler divergence as an estimate of reproducibility of numerical results. In: 2015 7th international conference on new technologies, mobility and security (NTMS), pp 1–5. https://doi.org/10.1109/NTMS.2015.7266501
57. Liu J-W, Chi G-H, Luo X-L (2013) Contrastive divergence learning for the restricted Boltzmann machine. In: 2013 Ninth international conference on natural computation (ICNC), pp 18–22. https://doi.org/10.1109/ICNC.2013.6817936
58. Yildirim I, Bayesian inference: gibbs sampling, p 6
59. Zhang C, Butepage J, Kjellstrom H, Mandt S (2018) Advances in variational inference [Online]. Preprint http://arxiv.org/abs/1711.05597. Accessed 08 Sep 2021
60. Gregor K, Papamakarios G, Besse F, Buesing L, Weber T (2019) Temporal difference variational auto-encoder. [Online]. Preprint http://arxiv.org/abs/1806.03107. Accessed 08 Sep 2021
61. Shekhar S, Xiong H, Zhou X (eds) (2017) Euclidean distance. In: Encyclopedia of GIS, Springer, Cham, p 556. https://doi.org/10.1007/978-3-319-17885-1_100372
62. Xue Y. Zhang L, Wang B, Li F (2019) feature selection based on the kullback-leibler distance and its application on fault diagnosis. In: 2019 seventh international conference on advanced cloud and big data (CBD), pp 246–251. https://doi.org/10.1109/CBD.2019.00052
63. Kim DH, Baddar WJ, Jang J, Ro YM (2019) Multi-objective based spatio-temporal feature representation learning robust to expression intensity variations for facial expression recognition. IEEE Trans Affect Comput 10(2):223–236. https://doi.org/10.1109/TAFFC.2017.2695999
64. Kucukelbir A, Tran D, Ranganath R, Gelman A, Blei DM (2016) Automatic differentiation variational inference. [Online]. Preprint http://arxiv.org/abs/1603.00788. Accessed 10 Sep 2021
65. Shin H-K, Lee W, Yun J-H, Kim H (2020) {HAI} 1.0: HIL-based augmented {ICS} security dataset, presented at the 13th {USENIX} workshop on cyber security experimentation and test ({CSET} 20), [Online]. https://www.usenix.org/conference/cset20/presentation/shin. Accessed 10 Sep 2021
66. Shin H-K, Lee W, Yun J-H, Min B-G (2021) Two ICS security datasets and anomaly detection contest on the HIL-based augmented ICS testbed. In: Cyber security experimentation and test workshop, Virtual CA USA, pp 36–40. https://doi.org/10.1145/3474718.3474719
67. Choi S, Yun J-H, Kim S-K (2019) A comparison of ICS datasets for security research based on attack paths. In: Critical information infrastructures security, Springer, Cham, pp 154–166. https://doi.org/10.1007/978-3-030-05849-4_12
68. Wood SN (2015) Core statistics. In: Cambridge core, https://www.cambridge.org/core/books/core-statistics/F303F4463E162C6534641616AE38C0A6 Accessed 29 Jun 2021

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.