A Review of Unsupervised Machine Learning Frameworks for Anomaly Detection in Industrial Applications

Usmani Usman Ahmad, Happonen Ari, Watada Junzo

DOI: 10.1007/978-3-031-10464-0_11

A Review of Unsupervised Machine Learning Frameworks for Anomaly Detection in Industrial Applications

Usman Ahmad Usmani[1], Ari Happonen[2(✉)], and Junzo Watada[3]

[1] Universiti Technologi Petronas, Perak, Malaysia
[2] LUT University, Lappeenranta, Finland
ari.happonen@lut.fi
[3] Waseda University, 1 Chome-104 Totsukamachi, Shinjuku City, Tokyo 169-8050, Japan

Abstract. Unsupervised learning, also known as unsupervised machine learning, analyzes and clusters unlabeled data utlizing machine learning techniques. Without human input, these algorithms discover patterns or groupings in the data. In the domain of abuse and network intrusion detection, interesting objects are often short bursts of activity rather than rare objects. Anomaly detection is a difficult task that requires familiarity and a good understanding of the data and the pattern does not correspond to the common statistical definition of an outlier as an odd item. The traditional algorithms need data preparations while unsupervised algorithms can be prepared so that they can handle the data in war format. Anomaly detection, sometimes referred to as outlier analysis is a data mining procedure that detects events, data points, and observations that deviates from the expected behaviour of a dataset. The unsupervised machine learning approaches have shown potential in static data modeling applications such as computer vision, and their use in anomaly detection is gaining attention. A typical data might reveal critical flaws, such as a software defect, or prospective possibilities, such as a shift in consumer behavior. Currently, academic literature does not really cover the topic of unsupervised machine learning techniques for anomaly detection. This paper provides an overview of the current deep learning and unsupervised machine learning techniques for anomaly detection and discusses the fundamental challenges in anomaly detection.

Keywords: Anomaly detection · Unsupervised machine learning · Outliers · Feature representation · Deep learning · Neural network · Machine learning · Real-time video · Pattern matching · Time series · Classifiers · Boltzmann machine · Metric analysis · Sampling · Digitalization · Industry 4.0

1 Introduction

Data representation by using the machine learning algorithms is an important concern in the current literature. Despite this, a considerable portion of the actual effort necessary to run machine learning algorithms is spent setting up in the feature selection and data

transformations. Feature engineering is useful, but it takes time and highlights a flaw in current learning algorithms: their inability to extract the data's information. Human intellect and prior knowledge can compensate for this flaw in the application design, but humans are usually good on noticing patterns, they can imagine the data could contain, which is not the case, for unexpected patterns or groupings. Making learning algorithms less dependent on features extraction will dramatically increase the machine learning's breadth and simplicity of use, enabling speedier development of new applications and, more importantly, advancement toward Artificial Intelligence (AI) utilization in multiple industries, and specially in traditional industries like (references) [1]. An AI must have a deep understanding of the environment around it, which can be accomplished if the machine can recognize and extract the underlying explanatory components inherent in low-level sensory data. Design development can be combined with feature learning to produce cutting-edge results on real-world problems.

The basic technique is to learn higher-level features along-side hand-crafted ones. Feature learning is data transformation and learning representation that makes the extraction of useful information from data, such as immunization records, tracking the patient's health history, understanding customer responses to new products, segmenting unfamiliar markets, differentiating a company brand from its competition, and repositioning a product after its market image has gone stale. The distribution of underlying explanatory variables for the observed data is presented in an appropriate probabilistic model form [2]. This research focuses on reviewing the deep learning techniques such as Convolutional Neural Networks (CNNs) [90], Re-current Neural Networks (RNNs) and Generative Adversarial Networks [91] which can construct more non-linear, abstract representations, among the many ways for learning the features representations. The composition of representations creates the deep architecture, with the number of layers being a free parameter that may be changed depending on the task's requirements. We look at recent breakthroughs in this field, concentrating on issues like finding the optimal aims for machine representations (i.e., inferences), geometric connections between feature learning and density estimation [3].

Machine learning is a subset of AI that helps a machine to learn automatically from the past data without programming explicitly [4]. Non-parametric local learners, such as kernel machines with a fixed generic local-response kernel, have been studied for flexibility by machine learning researchers (such as the Gaussian kernel). As previously shown in [5], the majority of these solutions rely on instances to directly map the target function. While smoothness is the desired assumption, it is insufficient since there is a large variation in the target function, and it expands exponentially in proportion to the number of the connected interacting components or input dimensions. However, it is better to construct a linear model or kernel system [6] on top of a learned representation: this is equivalent to learning the kernel, i.e., the function space. Kernel machines are important, but they rely on a predetermined similarity metric or feature space that allows for quick comparisons; we'd want to utilize the data to choose acceptable features as well.

Unsupervised learning is a kind of machine learning that uses as little human supervision as possible to discover data groupings or hidden patterns without using labels. Unsupervised learning, also known as self-organization, allows for the modeling of

probability densities across inputs instead of supervised learning, which generally uses human-labeled data. [7] Along with supervised and reinforced learning, it is one of the three major types of machine learning. A similar form employs both supervised and unsupervised processes. Two often-used unsupervised learning approaches are principal component analysis and cluster analysis. In unsupervised learning, cluster analysis is used to group or segment data sets with comparable properties to identify algorithmic linkages [9]. Cluster analysis is a machine learning method that divides data into clusters that are either unlabeled, categorized, or categorized. Cluster analysis, rather than listening to feedback, looks for patterns in data and reacts to the presence or absence of these patterns in each new piece of data.

This approach makes it easier to find data items that don't fall into either of the two groups. Although statistical density estimation is a frequent unsupervised learning application [10], it also has a variety of other applications, including data summary and interpretation. Furthermore, since we cannot determine how accurate the outputs are because the predicted output is unclear. The cluster method [8, 9] gives poor results when it comes to segmenting and targeting customers. Association mining detects groupings of objects in the data collection. Basket analysis is popular with businesses because it allows analysts to rapidly find frequently bought goods, creating more successful marketing and merchandising strategies. Unsupervised learning varies from classification and regression. The input data isn't labeled (i.e., no labels or classes are given), and the algorithm learns the data structure independently. As a consequence, there are two key grounds of disagreement. To begin with, since the data does not need to be manual, we may examine vast volumes of data. Second, although supervised learning employs an explicit fine measure, assessing the quality of an unsupervised approach may be challenging [10]. Principal Component Analysis projects data onto its orthogonal subspace feature is one of the most fundamental, straightforward, and extensively used dimensional reduction approaches [11]. All observations are ellipsoids in the original feature space subspace, and the new basis set in this subspace is aligned with the ellipsoid axis. Because the basis vectors are orthogonal, we can eliminate the strongly related features. Although the ellipsoid size is generally the same as the original spatial dimensions, in case the data is in a smaller subspace, new projections can be used to eliminate the subspace. We choose each ellipsoid axis in turn, based on the largest dispersion, in a'greedy' manner. One of the most prevalent issues in unsupervised learning is reduction.

Data visualization (e.g., the t-SNA approach) and data preparation for supervised learning algorithms may benefit from dimensional reduction (e.g., decision trees) [12]. While analyzing a time series, some critical questions are: Is there a general tendency toward average measurements? Is there seasonality or a predictable pattern of highs and lows that correlates to calendar time (seasons, quarters, months, days of the week, etc.)? Is there anybody here from out of town? In regression, outliers are data points that deviate significantly from your line. In time-series data, outliers depart significantly from the remainder of the data. Is there any long-term cycle or phase that the seasons aren't affected? Is the variance constant throughout time, or does it fluctuate? Is there a significant difference in the level of volatility in the series? Environmental samples of natural or man-made materials are often used to create unlabeled data. Images, audio

recordings, movies, news articles, tweets, x-rays (if making a medical app), and other unlabeled data may all be used.

On the other hand, unsupervised machine-learning algorithms learn what is normal and then use a statistical test to determine if a data point is abnormal. A device using this form of anomaly detection technology can identify all types of abnormalities, even those never seen before. Determining what is normal to follow the time series is the most difficult part of utilizing unsupervised machine learning algorithms to identify abnormalities. The following are the major contributions of this paper:

- We present an overview of the anomaly detection and briefly describe the deep learning models used for finding the anomalies in complex dataset.
- We study how unsupervised machine learning can be used for finding anomalies in various industrial and research domains.
- We explain the frameworks for anomaly detection and explain how the anomaly can be efficiently detected by using unsupervised machine learning architectures.

2 Unsupervised Machine Learning and Deep Learning

This section covers the unsupervised machine learning models and temporal connection modeling models and approaches. The learning process can create meaning on its own since it is unlabeled. Unsupervised learning can be used as a method of achieving a goal or as a goal in itself (discovering hidden patterns in data). In specific pattern recognition systems, the training data is a collection of input vectors X that do not match the target values. The purpose of these unsupervised learning problems might be to figure out how the data is distributed spatially, as in estimated density, or to cluster similar occurrences in the data. Now we give a brief overview of the various machine learning models.

2.1 Restricted Boltzmann Machines

Boltzmann machines shown in Fig. 1 are stochastic and generative neural networks that, given enough time, can learn internal representations and represent and solve complex problems [13]. The Boltzmann distribution (also known as the Gibbs distribution) is a fundamental concept in statistical mechanics that describes how entropy and temperature influence quantum states in thermodynamics [14]. Restricted Boltzmann machines (RBM's) are non-deterministic (or stochastic) deep Learning models with just two types of nodes: hidden and visible nodes.

All parameters, patterns, and data correlations are available once input is supplied. Consequently, they're as Deep Generative Models and Unsupervised Deep Learning, respectively [15, 16]. RBM's are a two-layer generative artificial neural networks. They can figure out what probability distribution their data falls within. Boltzmann machines with a limited number of visible and hidden unit connections are known.

With many inputs, the first step of RBM training is shown in the Fig. 1 below. The first hidden node will receive a vector multiplication of the inputs multiplied by the first weights column before adding the appropriate bias component [17].

The formulae of the sigmoid function is as follows:

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

So the equation that we get in this step would be,

$$H^{(1)} = S(v^{(0)T} W + a) \tag{2}$$

$$v^{(1)} = S(h^{(1)} W^{T} + a) \tag{3}$$

The hidden and visible layers' vectors with superscription ($v(0)$ signifies network feedback) are $h(1)$ and $v(0)$, respectively. This graphic now depicts the reversal phase, often known as the re-building phase [18]. During the back pass reconstruction, we compute the probability of output $v(1)$ based on input $h(1)$ and weights W depending on:

$$P(h^{(1)} | v^{(0)};\ W) \tag{4}$$

This is referred described as generative learning, as opposed to discriminating learning, which occurs in a classification problem (mapping of label inputs) [19]. Divergence in Contrast Boltzmann Machines (or) is energy-based models with a shared architecture of visible and hidden components [20].

$$E(v, h) = -\sum i \epsilon visible a_i v_i - \sum j \epsilon hidden b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{5}$$

where v, h_j, the binary conditions of the visible unit, hidden unit j, a_i, b_j are their preconditions and w_{ij} is their weight. The likelihood that the network will allocate to a visible vector is calculated by summing up all possible hidden vectors:

$$p(v) = \frac{1}{Z} \sum\nolimits_{h} e^{-E(v,h)} \tag{6}$$

This leads to a very simple learning rule for the stochastic climb in the log chance of the training data: where alpha is a learning rate.

$$\frac{\partial log p(v)}{\partial w_{ij}} = \langle v_i h_j \rangle data - \langle v_i h_j \rangle model \tag{7}$$

2.2 Autoencoders

An unsupervised artificial neural network learns how to compress and encode data efficiently before reassembling data from the reduced encoded representation to the representative representation that is as close as feasible to the original input by learning how to avoid data noise lower data.

An unsupervised artificial neural network learns how to effectively compress and encode data before reassembling data from the reduced encoded representation to a representative representation similar to the original input shown in Fig. 2 [21]. The network architecture can alter depending on whether it is a single FeedForward network, LSTM, or Neural Network.

[22, 23] Because the encoding process is based on correlated data compression features, the approach works well when the data are correlated.

2.3 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a class of artificial neural networks where connections form a directed or undirected graph between nodes along a temporal sequence and allows it to exhibit temporal dynamic behavior. A directed graph is generated by the connections between the nodes of a RNN over time shown in Fig. 3. As a consequence, it may display a variety of temporal behaviors like the trajectory of the states in a state space, followed by the system during a certain time interval [25, 26]. These are based on neural networks and have an internal state that enables them to take a wide range of input length sequences. RNN refers to two sets of networks with a similar general structure, one with a finite impulse and the other with an unbounded impulse [24]. Two applications include speech recognition and networked, unsegmented handwriting recognition [25]. All network groups' temporal behavior is difficult to anticipate. [26] This is known as neural network feedback [27].

The three categories of nodes are input nodes (which take data from outside the network), output nodes (which deliver results), and hidden nodes (which do not supply results) (modifying data) [28]. Sequences of real-time input vectors enter the input nodes one at a time for supervised learning in varied temporal contexts. As a non-linear function of the weighted total of all linked units', the rising non-input unit computes its true activation (outcome) at each time [29, 30]. This might be used to play a game where the number of points scored decides the winner.

Each sequence produces an error equal to the difference between the target signal and network activation. The cumulative error for a training set of distinct sequences is the total of all individual sequence defects [31]. An Elman network is a three-layer network with several backdrop units (shown as x, y, and z in the Fig. 3. The intermediate (hidden) layer is linked to the weighted background units [32]. As a result, the network can retain a state, allowing it to do tasks such as sequence prediction that would be difficult with a traditional multi-layer [33].

2.4 Deep Learning

Deep learning (sometimes called deeply structured learning) is a machine learning system that focuses on artificial neural network representational learning [34]. There are significant contrasts between and biological brains. The biological brains of most living things are fluid (plastic) or comparable, while neural networks seem to be static and symbolic [35–37].

For example, a deep learning system should figure out which qualities to employ to arrange the level better on its own. (Variable layer counts and layer widths, for example, might result in varying degrees of abstraction.) [38, 39]. The limit is a sequence of input-output modifications used to look for potential causal relationships between input and outcome. In a feed-forward neural network, the size is determined by the number of hidden layers plus one layer (as the output layer is also).

When a signal passes through a layer several times, the CAP depth in a recurrent neural network is almost endless [40]. Although there is no general agreement on what separates shallow from deep learning, most studies feel that deep learning necessitates the use of more than two CAP depths. The CAP of depth 2 is universal since it can

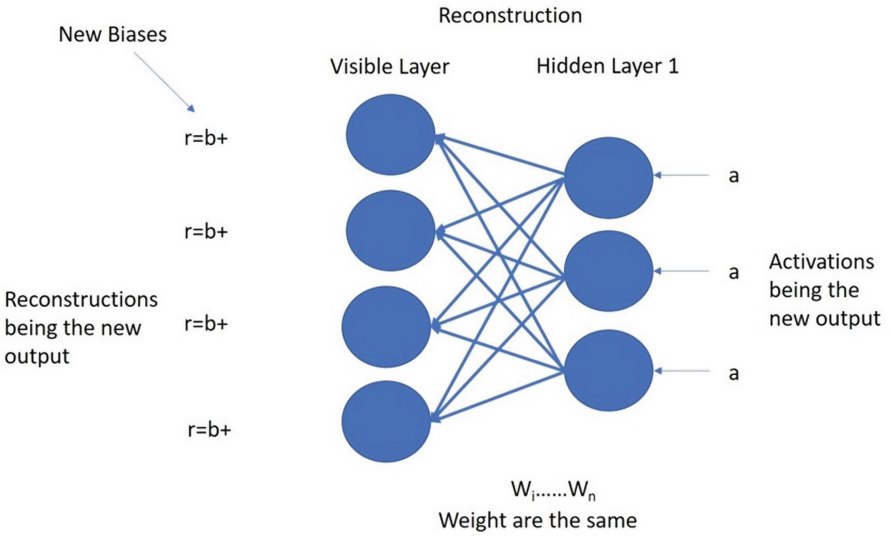

Fig. 1. Training an RBM with multiple inputs

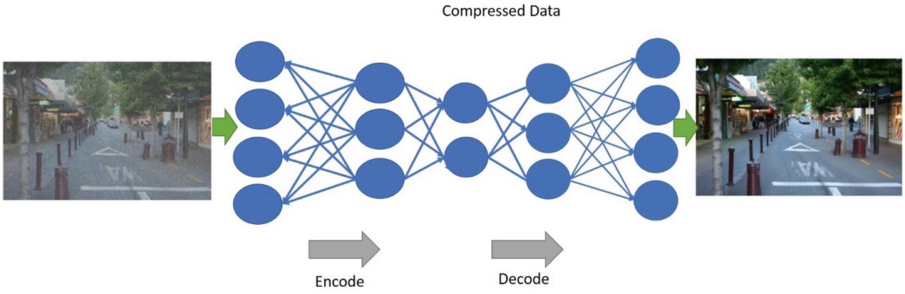

Fig. 2. De-noising of image

imitate any function [41]. On the other hand, the network function is unaffected by additional layers. To create deep learners, a greedy layer-by-layer strategy might be applied [42]. Deep learning aids in disengaging and identifying which brain regions improve performance [43].

Deep structures that can be trained without supervision include neural history compressors [44] and deep faith networks [45]. A neural network (CNN or ConvNet) is a deep neural network often used in deep learning image processing [56]. It's also called invariant ships or spacious artificial neural networks [89] because of its shared-weight design and translation invariance (SIANN) [46, 47]. Before transferring input to the next layer, layers mix it. It's comparable to how a visual brain cell responds to a specific stimulus [48]. In Convolution technique the number of free parameters is reduced in a network, allowing it to evolve faster [49].

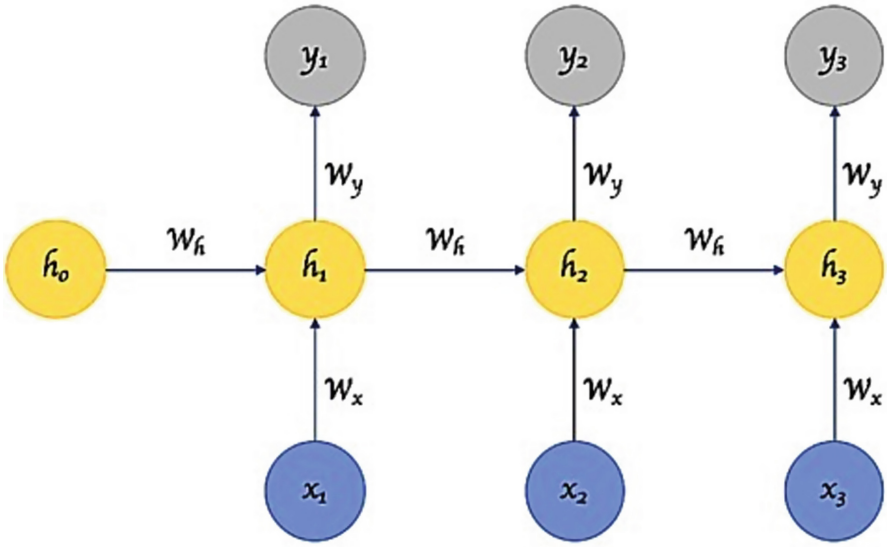

Fig. 3. Recurrent neural network

By integrating neuron cluster outputs from one layer into a single neuron in the next layer, pooling layers lower data. Regional pooling is often used to connect small 2 2 clusters. Pooling also computes a total or average of all layer neurons [50, 51]. In max-pooling [52], the biggest value for each previous layer neuron cluster is chosen [53]. In average pooling [54], the largest value of each previous layer neuron cluster is chosen. This is because convolution is done many times, considering the value of a single pixel and the values of the pixels surrounding it [55].

The memory footprint is minimized since each receptive field has its bios and vector weighting, while all receptive fields employing this filter have a single bias and vector of weighting [56]. Learning from temporal consistency in sequential data such as audio and video provides a natural and plentiful source of data that seems to be a physiologically more trustworthy signal than most present machine learning assignments [57]. The HMM (Hidden Markov Model) shown in Fig. 4 [58] is a Markov Statistical model that assumes the represented system is a Markov process with unknown (i.e., hidden) conditions. The Markov cycle is shown in the figure below by the connection between both are the HIDDEN STATES.

3 Unsupervised Machine Learning Frameworks for Anomaly Detection

In anomaly detection rare events are identified, e.g. observations or items that differ significantly from standard behaviors or patterns. Standard deviations, outliers, noise, novelty, and exceptions are all terms used to describe data anomalies. In this section, we will look at some common anomaly detection problems in various spheres, and

the models that have been used in the literature to tackle them. We explore mostly the industrial applications that are in demand so that these proposed frameworks helps in detecting potential accidents and economic losses by detecting the anomalies on time.

3.1 Unsupervised Machine Learning for Anomaly Detection in Electrical Substation Circuits

Cyber-physical systems (CPS) allow assets to monitor, track, and interact with one another for a physical network, such as smart grids, to function safely and effectively. According to the literature, CPS intrusion detection systems (CPS IDS) should detect attacks in host audit logs and network traffic and in various location (physical plane) measurements of various equipment. Physical limitations can be used to cope with atypical conditions in distributed agreement algorithms in power grids, voltage, and current control [59]. The detector might be programmed directly into a hybrid CPS IDS with CPS-specific physical limits. The current literature provide preliminary findings and an alternative classification technique for normal, fault, and attack states in a smart distribution substation. CPS uses this approach as part of a CPS IDS. The current works use RTDS to simulate the electric distribution system at the substation and collect data for the computer's learning.

The functional vector for each of the three phases comprises RTDS-generated, time-aligned stress and current magnitudes at four separate locations, for a total of 24 features. A time variable load represents a typical load profile for a residential customer in the simulated circuit. Five 24-s simulation time intervals, or five compressed days, replicate the 24-h stress profile. The feeder's real determined delivery substation price decides the rate at which loads are consumed. Samples are often used to train and evaluate data sets in machine learning. Generalization claims exclusively depend on validation set training method testing since the system learns from training set data. Even though it is non-deterministic and event traces with start and end transients that make an assignment to the event difficult, around 15 samples are used per fault trace for scoring and false alarms and 30 samples per attack for injection. If the methods locate 29 samples out of a (nominal) 30-sample assault trial, they'll claim a 96.66% detection rate [60–62].

This technique allows choosing the learning rate, pattern match quality, new pattern classes, and related learning patterns. This technique employs both external and internal learning loops. A sample pattern is supplied to the classifier on each excursion through the inner loop (a measurement track following normalization of the feature). The outer loop alters the learning rates and criteria and connects learned to related classes. Transform samples are created for the designs and unique features in different units vary in size (volts and amperes). Consequently, the medium is erased and divided by default to normalize each feature (column in the matrix). Subtract the mean row from each row (matrix time sample) by aligning the sample around zero and reducing the impact of the load curve. Finally, a function is utilize (squashing) with S equal to 1.0 for our research. The SOM (map) is the name given to the collection of patterns considered by nomenclature [63].

A SOM pattern class is eliminated from SOM if it earns too few data patterns at the end of the inner loop, as indicated by a cutting threshold (pruned). Comparable patterns are seeked at the end of each external loop phase (SOM is run effectively through the SOM). Depending on the number of patterns each model has won, models that match

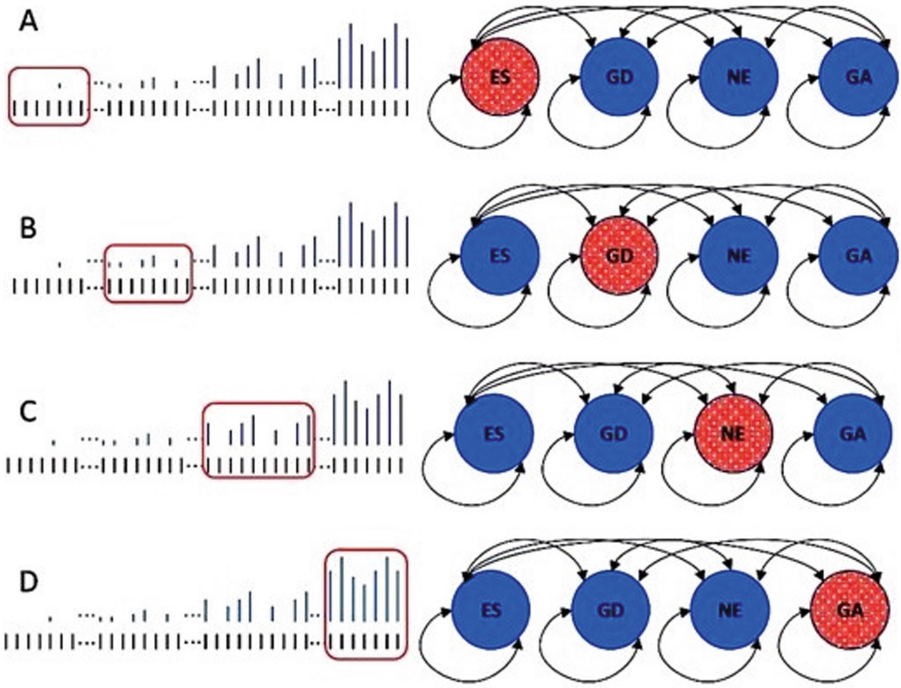

Fig. 4. A hidden markov model

other patterns based on suit criteria may be blended using a weighted average. Except for the last iteration, the results of this article nearly reflect the cutting and pattern mixing of the outer loop. Our study contained 31,250 time-aligned measurement patterns, the first n of which were used for the exercise and the remainder for testing and validation. Training sets are created by varying the number of training samples, which might have two faults, all three defects, and one injection attack. The first 5000, 5500, or 6000 samples are choosen to demonstrate how the machine learning algorithm can distinguish between unique occurrences that are or are not in the training set. Relay faults 91 and 92 can be found in samples 1–5000, relay defects 93 can be found in batches 5001–5500, and relay assaults 90 can be found in samples 5500–6000.

Six actual defects and eight injection assaults were picked at random from the remaining samples [64]. Standardization should be utilized for all steady-state activity samples. Normal operation phases, non-malicious faults (compatible with KCL/KVL), and inaccurate measurement injection should all be distinguishable by the classifier. Teach a pattern that corresponds to injection as a class that does not match any regular or non-malicious fault pattern throughout the training phase [65]. A pattern matching an injection should not belong to either the normal or non-malicious pattern classes during the validation process.

Seven patterns are identified by the classifier. The typical examples are taught as a class, whereas the F91 and F92 cases are learnt as separate pattern classes, as in previous results. F93 is divided into two pattern classes, with 14 samples in the trace's centre

remembered as one pattern and samples at the start and end remembered separately. The attack trace A90 is divided into two pattern classes: one with 26 samples and the other with three samples at the start and end. Sample 17135 comes from the F92 event and is a single false alarm sample. As shown in the n = 5500 experiment, samples for different occurrences of attack A93 match the learned pattern for fault F93, lowering attack detection performance. As in the previous experiment, 21 samples of the A93 event near sample 24411 are classified as anomaly yet have very high scores, whereas the A93 trace near sample 26233 is completely disregarded. This run has a less than 0.1% false alarm rate and a detection rate of 71.11% of samples or 83.33% of traces, with all missed detections happening at position 93. These findings show that including an attack trace in the training phase has no influence on the results or detection performance, implying that attack-free training data is not required.

3.2 Unsupervised Machine Learning System for the Detection of Cyber Based Attacks in Smart Grids

Intelligent grid technologies increase the electrical system's reliability, security, and efficiency. However, its reliance on digital communication technology creates new risks that must be handled for the power delivery to be efficient and dependable [54]. This research argues that issues may be recognized without being seen by using statistical correlations between data. The goal of the current unsupervised machine learning algorithms in this domain is to develop an all-encompassing anomaly detection engine for smart grids to distinguish between a genuine outrage and a disturbance or sophisticated. The proposed method employs symbolic dynamic filtering (SDF) to reduce processing requirements while revealing causal linkages across subsystems [59].

Simulation findings on the bus systems IEEE 39, 118, and 2848 confirm the proposed technique's performance under various operating situations. According to the data, the method is 99% accurate, with 98% true positives and just 2% false positives which shows the better performance of the method [66]. The following are the work's key contributions: Without the ability to categorize data sets, a mechanism for identifying a problem in smart grids arises that is unmanageable. SDF data reduction is indicated as an approach for reducing computing effort. Creating a DBN-based learning paradigm that works [67]. The authors provide a model-free approach to integrating smart grids into hierarchical and topological networks; the smart grid is seen as a multi-agent system in Fig. 5.

These agents include a generator, a measuring unit, a distributed control agent, and an energy storage system that may inject or absorb the system's actual power [67]. There are two states in which the system may exist: dynamic and static. The system condition (X) shows both the dynamic state of the generator (e.g., rotor speed, rotor angle) and the static status of the network (voltage magnitude and phase angle). The measurement of the nonlinear function is indicated by *h*, and the nonlinear, dynamic behavior of the generators is marked by $\boldsymbol{f}$ (). The letters u and z stand for vector performance and measurement, respectively. This research aims to understand better and predict the intelligent power grid (as shown in this section) to identify anomalies. The fourth (two-axis) model of the generator is shown in the Fig. 6. SDF, DBN, and RBM are used to provide a computer-efficient approach for detecting subsystem linkages [81]. This is

based on the notion that the invader has limited resources and can only use a restricted set of tactics.

This is a reasonable assumption since it's challenging to think that all sensors offer inaccurate data when utilizing power networks. On the other hand, changing all metrics takes a long time and costs a lot of money for attackers. It would be difficult for an outsider to comprehend the software. Consequently, the attacker only has a rudimentary grasp of the architecture and security measures of the system. This data may be collected by statistical analysis of data sent to the control center by remote terminal units (RTU) or by physically organizing the node's safety data. This section shows how to train data for a detection system using DBN modeling, MI feature removal, and RBM. The unattended DBN model records system behavior patterns, while DBN and MI evaluate smart grid test systems with massive amounts of data [68].

The system is first separated. After then, the SDF is used to figure out what's causing the nominal subsystem characteristics. Of dealing with whole systems at once, the recommended technique is a computer-friendly tool that saves time and money by 1) choosing a subset of measures by selecting features and SDF, as well as domain breakdown and parallel data processing to selected subsystems. The research develops an anomaly detection tool that uses a feature extraction method and time series to identify causal links between subsystems in real-time and with little processing cost. DBN and Boltzmann-based learning approaches uncover non-observable dangers using free energy as an anomaly index. The performance of the recommended method was evaluated using a variety of IEEE test systems and operating settings (TPR, FPR, and ACC). According to statistics, the device has a 99% accuracy rate, a 98% TPR rate, and a less than 2% FPR rate [69]. In order to verify the efficacy of the proposed technique, four potential scenarios are investigated: FDI attacks on lines 6–31 and 11–12: FDI attacks on lines 6–31 and 11–12: FDI attacks on lines 6–31 and 11–12: FDI attacks on lines 6–31 and 11–12: 1) no attack, 2) random attack, 3) single FDI attack on 6–31, 4) numerous, simultaneous FDI attacks on lines 6–31 and 11–12. The proposed technique is compared to the LNR and Chi-Square tests, the two most often used BDD approaches.

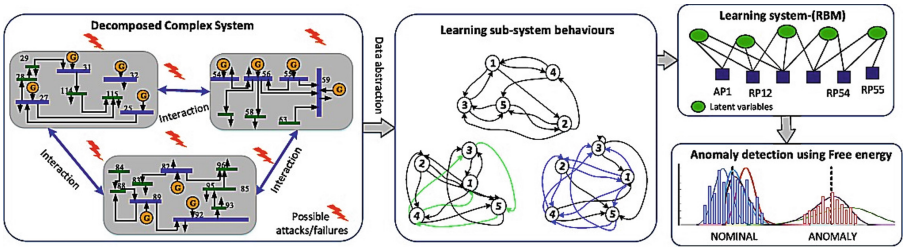

Fig. 5. Unsupervised machine learning framework for detection of cyber-attacks [54]

To reduce false positives due to noise, the threshold is set at 3 and the standard deviation is set at, resulting in an FPR of less than 1% [44]. The threshold for all detectors has been standardised for accurate and wide comparison. The LNR test uses the same methods to establish the threshold. Because the attack is unintelligent, it will leave a trail in the data sets, informing the operator that an attack has taken place. The measurement

set's random anomaly data causes significant changes in the measurement residual vector, resulting in a cost function increase. We look at the cost function based on the data residual in optimum state estimation. Under normal functioning, the cost function follows a normal distribution with a zero mean when no anomaly data is accessible in the system.

The cost function will pass the optimum state estimation threshold in a random attack. As a result, both the LNR and the chi-square tests will set off the alert. In the face of single or multiple FDI attacks, the cost function for both the LNR and the Chi-Square detectors stayed within the real range of predefined thresholds, resulting in normalized residue levels that were lower than the specified threshold, making it impossible to detect the attack in the system. The output of the suggested detector, however, exceeds the provided threshold with the identical setup, resulting in an alert.

The residual vector of the measurement vector is used in the LNR and Chi-Square tests, however cyber-attacks are designed to leave no trace in the residual vector. All of the case studies had the same outcomes. The average detection time in all case studied was 1 ms, with ranges of 0.2 ms. Any FDI attack on a line or system architecture, in general, leads in the same network alterations, with slight variations. As a result, the suggested method can detect FDI attacks coming from a range of sources. The suggested system's success rate is also independent of attack situations since it analyses patterns in both compromised and regular data. The methodologies for identifying smart grid anomalies discussed in the literature are mostly machine approaches with limits for dealing with constantly changing cyber threats. Using a feature extraction approach and time series partitioning, it presents a real-time and computationally efficient anomaly identification tool that identifies causal relationships across subsystems. Hidden attacks that employ free energy as the anomalous index are discovered using the DBN concept and learning algorithms based on the Boltzmann Machine. The performance of the suggested approach was examined for a range of criteria on a variety of IEEE test systems and in a variety of operating conditions (TPR, FPR, and ACC). According to the numbers, the system has a 99% accuracy, a 98% TPR, and an FPR of less than 2%.

3.3 Unsupervised Machine Learning for Anomaly Detection in Network Centric Architecture Based on IoT

The vital infrastructure networks should be designed in a way so that all the cyber based attacks can be prevented. For example, patches and software updates for antivirus software have failed to protect IoT apps from security flaws. The authors propose a behavior-learning approach for detecting sensitive situations [70]. The current literature demonstrated [70] that they could utilize unsupervised machine learning to identify different forms of assaults in real-time, utilizing the predictability of TCP traffic in IoT devices. The machine learning classifier can distinguish between normal and abnormal traffic based on a small number of variables. The current research concepts can be incorporated into a larger network through IP spoofing, allowing SDN-based processes to avoid attack traffic close to the source to be adopted. In terms of identifying new and unexpected attacks, unsupervised ML systems beat supervised ML techniques by an accuracy of around 15% [70].

The research show that ML models can learn from IoT network data by exploiting reconstruction faults. Previously, these methods were primarily used in the security

business for feature selection. Our unconstrained machine learning classification system was built to spot SYN floods and slow HTTP assaults in any IoT networks with near-perfect accuracy and minimal latency. The current literature looked at how well deep learning (auto-encoder) and statistical classifications performed in detection machine learning (ML) classifiers (PCA). It also demonstrated that both of these non-controlled ML classifiers outperformed a supervised classifier regarding fresh and unexpected hazards (SVM). The research show how the solution could be incorporated into a broader network to identify weak endpoints in the face of IP spoofing and block attack traffic near the source using SDN-based procedures. The focus on a range of retraining techniques for keeping our classifiers current and coping with network abnormalities is shown in Fig. 6 [71, 72].

Three types of data sets are gathered from three separate hosts in our simulated network: Type A is a benign data set. Type B is packets caught during an SYN attack. Type C is packets collected during a slow HTTP attack. The attack epoch for Type B data sets is generally 40–60%, while the attack epoch for Type C data sets is 70%. Type A data sets utilize two unsupervised classifiers per host (one using and the other using PCA). To train supervised SVM-based classifiers, Type B data sets are employed. We put both of our classifiers to the test on Type B and C data sets to see whether the attacks could be detected. Python scientific library and TensorFlow is used to create machine learning models. All layers employ the ReLU activation function except the output layer, which uses the activation function. The middle layer is the bottleneck layer.

The ReLU activation function is operated on every level and the Mean Square error loss is minimized using the Adam optimizer, then train them in 32 sizes over 100 epochs. In the loss function, there are no terms. The library's default learning and weight loss settings are utilized. The layer's measurements are as follows: The system comprises seven input and output layers, four bottleneck levels, and fourteen extra layers. Science-learn library model is used with default PCA values, followed by a polynomial kernel for SVM that outperforms other kernels (e.g., linear, RBF). When trained on innocuous traffic data, auto-decoder-based classifiers can anticipate network activity, identify irregularities, and detect attacks on the industrial Internet [73].

Some supervised ML classifiers outperform when it comes to detecting new and previously unknown dangers. Machine learning technique are also created for recognizing affected sources as IP spoofing, the method want to broaden the scope of this first investigation in the future. We'll need to expand the quantity and types of IoT devices we utilize to collect data. Even though the examined flows were limited to TCP, future protocols are fore-shadowed and risks. Although comparisons with other unsupervised processes such as single class and clustering should be made, the focus of this study was on re-training approaches and source identification. Another area that has to be investigated further is assessing source behavior. Consider changing settings throughout the classifier training phase to enhance attack detection.

Unsupervised classifiers like autoencoders and PCAs work well on Type B test data sets, which contain attack traffic, after being trained on Type A benign data sets. The results indicates that when trained on Type B data sets, the supervised SVM classifier performs well (known attack). On Type C data sets, the unsupervised Autoencoder and PCA classifiers continue to beat the supervised SVM classifier, while the supervised

SVM classifier shows a considerable reduction in performance. These findings demonstrate that the Autoencoder-based classifier, which was trained only on benign traffic data, can recognize a broad spectrum of DDoS attacks. On the other hand, these data suggest that the SVM-based supervised classifier is incapable of categorizing unknown attacks for which it has not been trained.

3.4 Unsupervised Anomaly for the Detection and Diagnosis in Multivariate Time Series Data

Many real-world systems, including power plants and wearable devices [88], acquire multivariate time series data rapidly. The purpose of multivariate time series anomaly detection and diagnosis in certain stages is to find out what's wrong and why it's happening. As a result, developing such a system is challenging since it requires the recording of time dependency in expanding time series and the storing of linkages between different time series pairs. The applications used should also be noise-resistant and provide operators with varying degrees of anomaly depending on how often specific occurrences occur. While various unattended anomaly detection algorithms have been created, only a handful are capable of addressing all of these problems simultaneously.

The authors provide an MSCRED (multi-scale innovative encoder-decoder) method for detecting and diagnosing multi-variate time series data problems in this work. MSCRED first produces multi-scale (resolution) signature matrices to determine numerous device status rates at various time phases. The inter-sensor (time series) correlations are then encoded using an encoder, and temporal patterns are stored using a focus-based Long-Short Term Memory network (ConvLSTM) [74]. Finally, a decoder reconstructs the input signature matrices using feature charts that include correlations and temporal information, and the residual signature matrices are utilized to detect and diagnose problems. MSCRED beats state-of-the-art baseline approaches, according to a detailed empirical assessment using synthetic and data from real-world power plants shown in Fig. 7.

Use a recurrent encoder-decoder to avoid the issues mentioned above (MSCRED) [75]. MSCRED creates multi-scale signature matrices to characterize different degrees of device status (resolution). The passage of time may be broken down into several phases. Multiple degrees of gadget status, in actuality, signify the risk of some unplanned occurrences. The correlation patterns (time series) are then encoded using an encoder, and the temporal patterns are aggregated using a focus-based Long-Short Term Memory (ConvLSTM) network [76].

In contrast, a decoder is a function map that stores both temporal and correlations. Signature matrices and residual signature matrices used for reconstruction may be utilized to identify and address anomalies. According to the idea, if MSCRED has never experienced a similar device state before, it will not recreate the signature matrices effectively. Anomaly detection, root cause identification, and anomaly duration are the three main tasks in anomaly detection and diagnosis [54]. In contrast to previous research, which focused on each problem independently, the methods are tackling all of these issues simultaneously. An encoder for inter-sensor correlations, cautious ConvLS networks for temporal pattern integration, and a decoder for signature matrix reconstruction are used by the authors to generate. MSCRED is the only model which uses multivariate

time series similarities to identify anomalies and achieves all three goals simultaneously. MSCRED out-performs state-of-the-art fundamental approaches, according to our results [77].

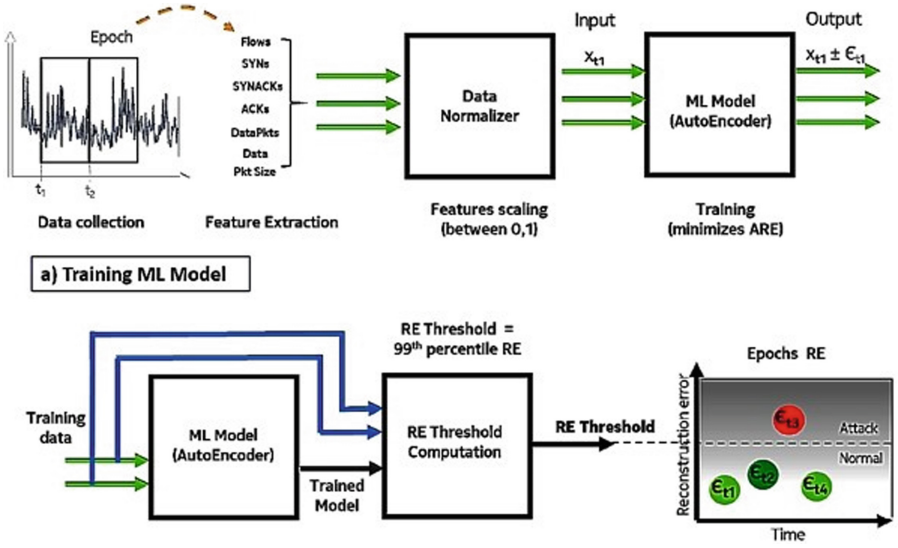

Fig. 6. Training of model and epoch classification based on the reconstruction error [58].

In this work, MSCRED signature matrices includes channels (s = 3) for capturing system status in varied sizes. To determine the severity of an anomaly, MSCRED(S), MS-CRED(M), and MSCRED(L) anomaly scores are created based on the residual signature matrices of three channels, small, medium, and large, with segment sizes w = 10, 30, and 60, respectively (L). Then we assess how well they do on three different sorts of anomalies: short, medium, and long, which last 10, 30, and 60 s, respectively. MSCRED(S) detects all forms of anomalies, while MSCRED(M) detects anomalies that persist for a long or short period of time. MSCRED(L), on the other hand, is only capable of detecting long-term issues. As a result, the three anomaly ratings are utilized to determine the severity of an anomaly. It's more probable that the aberration will continue if it can be observed in all three channels.

It can also be a one-off or short-term occurrence. MS-CRED(S) finds all five anomalies in this case: three short-duration anomalies, one medium-duration anomaly, and one long-duration anomaly. MSCRED(M) misses two short-duration anomalies, whereas MSCRED(L) identifies just one long-duration anomaly. In four injected anomalous event residual signature matrices, the outcomes of the root cause inquiry are also shown. In this situation, we can clearly identify more than half of the uncommon underlying reasons (shown by red rectangles in the rows and columns).

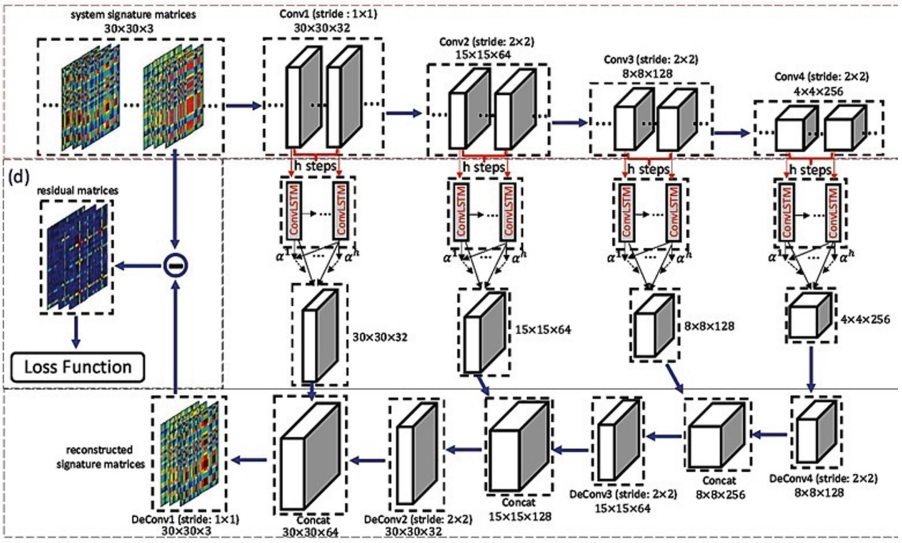

Fig. 7. Framework of the model that has been proposed: (a) In this the signature matrices are being encoded via fully convolutional neural networks. (b) Describes the temporal patterns that are being modelled by attention based convolutional LSTM networks. (c) Signature matrices being decoded via deconvolutional neural networks. (d) Loss function [59].

3.5 Unsupervised Machine Learning for Anomaly Detection in Unmanned Aerial Vehicles

To address a variety of resource and latency restrictions, a real-time anomaly detection system needs a steady supply of labeled and operational data. Most solutions rely on set rules that vary based on the circumstance, whereas traditional methods to the issue rely on well-defined qualities and supervised historical experience shown in Fig. 8. These principles work well in controlled conditions, but they can't be employed outside of known instances since they rely on a large amount of data to detect abnormalities. Existing literature is examined to find known and unknown anomalous events and think outside the box to improve decision-making [78].

The isolation forest's value in engineering applications is evaluated using the Aero-Propulsion System Simulation to outperform other uncontrolled distance-based approaches. The scientists employed an unmanned aerial aircraft to show alternate system utilization to conduct real-time testing. Because of the conditionality curve, the most widely used detection algorithms depend on distance measurements, which might be erroneous in high-dimensional scenarios. As a result, these systems aren't built to detect abnormalities, false alerts or alarms can be issued. Feature elimination is common, and PCA and auto-encoders are employed to reduce the data set; nevertheless, real-time solutions are difficult due to the computational cost. A summary of current developments in aircraft anomaly detection systems.

In most anomaly detection systems, the isolation forest is an out-of-the-box solution for dealing with and correlation challenges that do not need expert knowledge in the setup

process. Over time, it has grown to encompass a range of more powerful algorithms, such as random forests and isolation forests, under the banner of ensemble techniques. In various feature and training environment sectors, the former creates a flurry of decision-making agreements. The findings of each tree, as well as vote for the best prediction, are combined. As a result, predictor instability is significantly reduced. The latter goes a step further by analyzing data in high-dimensional space by constructing a random binary tree and separating the input space into patterns, assuming that anomalous behavior falls into the separated regions [79].

Isolating aberrant data from ordinary data should be easier using the forest isolation strategy. This situation has been recursively partitioned until each data point has its leaf inside the tree. When isolated, the depth of a data point in this tree is the statistic that matters here (the number of iterations to reach from the input data to the sample). This approach is used to create more decision trees to isolate an exception with just a few divisions if it occurs. The distance traveled to an anomaly is often substantially less than the distance traveled to ordinary data because anomalies are regarded as unusual or notable. Finally, the distance from the path is normalized and crosses the depth in all of the trees to determine the anomaly. Because low average tree depth data sets may isolate fewer splits, the methods have a larger anomaly score, implying that a higher score is anomalous. Data is carried to a terminal node or maximum depth via each isolation tree during preparation (forest). The depth option controls the level of detail on the anomalous screen.

The split attribute is used to divide the tree while the tree.height property is used to determine the node's height. The is the point when two objects are split apart. Left child trees make up half of the space tree model, while right child trees make up the other half. A data window is employed during the online operation to send data samples to the server. The amount of data points necessary to study this data determines the global window [80]. This method may also be used to evaluate multivariate data. The rest of the computations are completed using the standard procedure for calculating results. Although the method works effectively with high-dimensional data, correlation analysis and sorting of the data are still required before searching for anomalies. The correlation coefficients are employed in this feature selection technique to examine and consolidate the relationships between variables. A pair of synthetic anomalies were constructed at $t = 5000$ and $t = 10000$ for evaluation purposes. It causes anomalies by altering the random mean and variance of Gaussian distributions [2]. In the isolation forest, a hundred trees were trained. The results are depicted, as well as the distribution of points assigned to anomalous and non-anomalous data. The bulk of normal data is graded between 0.6 and 0.7, despite the fact that the distribution of anomalous observations exceeds 0.75.

25 However, locating it remains a struggle. This approach is useful not just for labeled data, but it can also be used to provide warnings when the probability surpasses a certain threshold, such as the 95th percentile. The outcomes follow looks at all nine characteristics that affect system behavor during UAV takeoff and hovering. Many of these messages are caused by intermittent sensor connectivity issues. The PCA results three main components, shows anomaly-tagged points. Alarms were raised between 189 and 206 s and 325 and 345 s, according to the data. It is impossible to discern why an

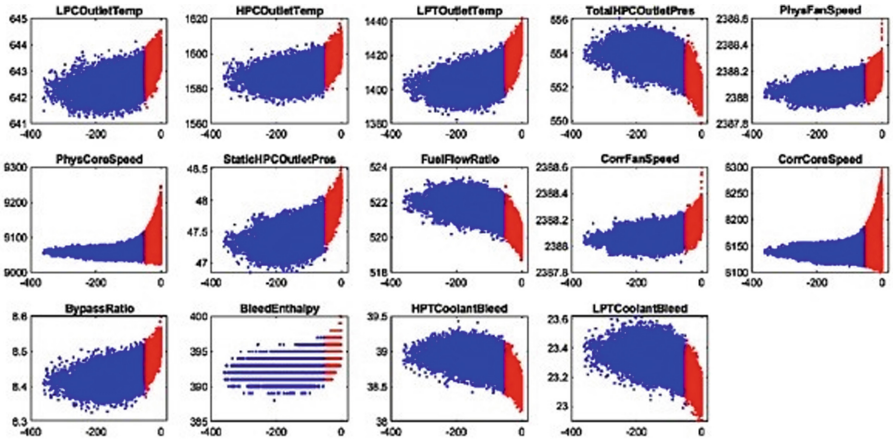

Fig. 8. Simulation of commercial modular aero-propulsion dataset [68].

event was unusual just by looking at the number at the time. As a consequence, the analyst is stumped as to where to start their investigation.

A number of criteria were analyzed and grouped together to locate and pinpoint anomalies within that group in order to fix this issue (of variables). This implies that the algorithm would have to be performed individually (and in parallel) on each group in order to discover anomalies in the incoming data. Despite the fact that this seems to be the best strategy for making a real-time decision, the authors decided to do a post-offline analysis by statistically examining the odd occurrences in the data using the violin plot, a visually attractive technique. This approach can also be used to rank potentially anomalous variables by their spread and skewness, as well as those with the greatest number of points outside the min/max quartile range. The most changeable variables are gyro readings 4,5,6, and variable 9, with variable 9 having the largest variation to contribute to the isolated forest score. When the video from the UAV experiment was analyzed at these points, it was determined that the system was attempting to restore its height after losing it. Although this strategy cannot ensure a definitive diagnosis of a problem's root cause, it helps to get a better knowledge of the possibilities and therefore narrow down the search.

3.6 Unsupervised Machine Learning Algorithm for Anomaly Detection in Real-Time Video Surveillance

The need for enhanced real-time video surveillance has risen due to rapid urbanization and self-driving manufacturing settings. Recent improvements in artificial intelligence for video surveillance anomaly identification directly address these difficulties, disregarding the changing presence of aberrant activity for the most part. Another issue is the sparse assessment based on a reconstruction error and the dependency on a known normality training. To address the constraints and limits of real-time video surveillance anomaly detection and localization, the authors suggest an ISTL. ISTL is uncontrolled

deep learning that uses active, fuzzy aggregating learning to continuously update and discriminate between new anomalies and normalcy as they emerge over time.

The accuracy, robustness, total computational, and contextual elements of ISTL are shown and assessed using three benchmark data sets. These findings back up our participation and the technology's potential for real-time video monitoring.

A deep learning model for online anomaly detection and localization learns typical behavior patterns from video surveillance data. To adapt swiftly to changing unknown/new normative behaviors, rapid accumulation of active learning outcomes in the continuous learning cycle is essential. Analyze the video surveillance stream utilizing two criteria: anomaly threshold and temporal threshold, rather than making an arbitrary judgment based only on reconstruction mistakes. The Chinese University of Hong Kong's Avenue [81] and the UCSD Pedestrian [82] (Pedestrian 1 and 2) are utilized as benchmark Video Surveillance to show and assess the essential components of ISTL (CUHK) shown in Fig. 9 and Fig. 10.

The picture measures 224 pixels by 224 pixels and has a pixel normalization range of 0 to 1. Based on the frame rate of the needed training data, which is roughly one-third of a second longitude, we build a temporal cuboid range of $T = 8$ (i.e., 26 FPS). Due to the huge depth of the input cuboids, T selection is focused on enhancing the movement to be taken in following frameworks while restricting deep learning model convergence. When the input surveillance data has lower frame rates, long movements may be caught with limited temporal depths. In this work, we used deep learning and active learning to create a new approach for identifying spatio-temporal abnormalities in real-time video surveillance. The methodology addressed three significant challenges: detecting abnormal behavior in video surveillance streams while managing high-dimensional data streams in real-time, formulating abnormality identification to learn normal, and adapting to dynamically evolving normal behavior using fluid integration and active learning. The suggested ISTL method used a self-encoder model with convolution layers to learn spatial regularities and ConvLSTM layers to learn temporal frequencies while keeping the video stream's spatial structure. Dynamic integration of input from human observers is integrated into a continuous, active learning process to address the issues associated with ISTL. According to the results of three studies, the suggested approach is accurate, resilient, low-cost to process, and incorporates contextual indications, suggesting that it is acceptable for use in industrial and urban contexts. A Gaussian mixed model was used in this experiment.

The first parametric technique uses several multivariate dispensations for widespread modeling addiction between two distinct photographs taken separately. The goal of third-family approaches is to evaluate the link between historical and varied photographs and current places before classifying and discovering changes in the two images using invariant measures of similarity through image mode (such as correlation and mutual information). The purpose of the anomaly-based CD problem is to find (typically rare) variations in ground characteristics across two heterogeneous images collected in the exact location using two different imaging modalities. It's a binary categorization activity in which (small) local spatial variances are probable signs of anything that's changed over time in the region of interest, and anomalies may be detected as a consequence (i.e., varying data seen through two different image modalities). [83] In contrast, the test

phase preserves the solidity to recognize the minority class, i.e., the shift class's unusual events, as anomalies [58].

Learning a compressed representation in the least-squares sense, reducing reconstruction errors in residual space for the two imaging modalities, and estimating the reconstruction error of any bi-temporal input pattern as an accurate anomaly value from a local gray-level set are just a few of its main features. This score is then used to differentiate between patterns that haven't changed and abnormal (abnormal) patterns created by an abnormality (shift mark detection). The authors suggest learning a layered, limited neural system model which may be learned in phases and serves as a good representation for improving our anomalous pattern-based model. They also recommend employing a stacked sparse, which may find intriguing structures in image data and offers an unsupervised reconstruction framework made up of many sparse layers [76]. It enables us to build a trustworthy anomalous CD model for identifying weird and irregular properties with a minimal margin of error. Cross-modality and functionality were learned deep and other deep learning methods. This model includes several intriguing features. A stacked sparse model autoencoder with satin and purine neural functions is utilized before and after training to learn about and infer an efficient latent representation of common visual patterns in pictures. By encoding and decoding the pair's inputs with its secret, stacked images, the anomalous CD model trains regular image patterns (belonging to the class label), and the changing class is distinguished from irregular feature patterns in the residual space to recognize and distinguish it from regular image patterns (belonging to the class label).

The results show a qualitative assessment of the anomaly places. In the UCSD Ped 1 dataset, ISTL finds anomalies such as bicycles and automobiles on the routes, pedestrians crossing pathways, crowd lingering, and persons pulling trolleys. Negative skateboarding detections in the Ped 1 dataset were incorrect. Only 10 of the 12 test video clips featuring skateboarders were recognized by the ISTL model. All video frames, including skateboarding, were recognized by the Ped 2 dataset. The camera viewpoint in the Ped 1 datasets explains this since the height makes distinguishing between pedestrians and skateboarders difficult. According to the UCSD Ped 2 test samples, bicyclists, automobiles, and pedestrians all go in opposite directions. Biker anomalies were the most prevalent in the Ped 2 test samples, occurring in 11 of the 12 cases. The CUHK avenue dataset contains an abandoned bag, a person tossing a bag, a little kid playing in the surveillance area, people walking in the other direction, and individuals sprinting. To show ISTL's active learning capacity, pedestrian route scenarios were explored from the UCSD Ped 1 and Ped 2 datasets. Since bicycling through pedestrian walkways was considered a common activity in this study, all anomaly detections from rider test samples were deemed normal. To train the ISTL model using human observer verification, four samples are tested from each of the Ped 1 and Ped 2 datasets. After the training phase, anomalies in the test samples are looked at and the four samples are rejected that were chosen for further training.

Two test samples involving cyclists were identified as abnormal during the analysis of the Ped 1 dataset due to crossing sidewalk cycling motions. Two previously recognised as uncommon test situations are utilized to further explore the efficiency of the active learning technique: 1) on a pedestrian walkway, a cyclist pedaling alone; 2) on

a pedestrian walkway, a cyclist riding beside a vehicle. Test video A was judged to be okay after the evaluation, however test video B was found to be anomaly. Video B was ruled anomaly due to the moving car, however video C was deemed normal. The anomaly detection technique's real-time video surveillance capacity, as well as the compute overheads for the sequential process of anomaly identification and localization, were assessed. The average time it takes to detect and locate anomalies is 37 ms. At a frame rate of roughly 27 frames per second, ISTL has shown the capacity to identify anomalies in video surveillance feeds in real time. Although frames are expanded for anomaly detection, localization is relied on the original frame resolution, hence differences in initial resolution have been linked to differences in dataset processing time. The ISTL was used in video surviellance in a sequential manner. On the other side, detection and localization are parallelized, lowering run time and allowing for greater FPS rates.

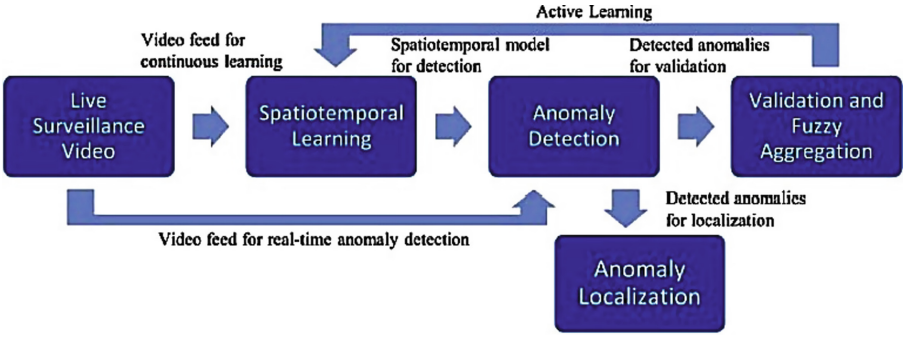

Fig. 9. Proposed framework for anomaly detection in real-time video surveillance [75].

Fig. 10. Localized anomalies described as. (a) UCSD Ped 1 dataset, (b) UCSD Ped 2 dataset, and (c) CUHK avenue dataset. That is best being viewed in color [75].

3.7 Unsupervised Machine Learning Approach for Anomaly Detection in Hyperspectral Imaging

Due to its high, redundant data and restricted ranges, image anomaly detection (HSI) faces various challenges such as lack of a common standard for manufacturing of hyperspectral sensors, insufficient labeled data for training, high volume of produced data and

the high cost of satellites and hyperspectral technologies. To overcome these issues, the authors propose a novel unattended feature representation technique based on a spectrum limiting methodology in adverse (AAE) that requires no previous information. To improve hidden node discrimination, we developed SC AAE, a method based on HSI characteristics. The current method [79] employs a spectral angle distance to the AAE's loss function to attain spectral precision. Due to the differences in contribution levels of each hidden node to anomaly detection, they fuse the hidden nodes individually using an adaptive weighting approach. The BKG is removed using a two-layer design while retaining its unique features. Our proposed method outperforms the current procedures, according to the testing results. For the first time, one of the generative models, AAE, is depicted in this article. A spectral restriction (SC AAE) approach is suggested to guarantee that deep-layer hidden nodes appropriately characterize both the anomalies and the BKG, given the anomalous and BKG pixels in the original feature space shown in Fig. 11.

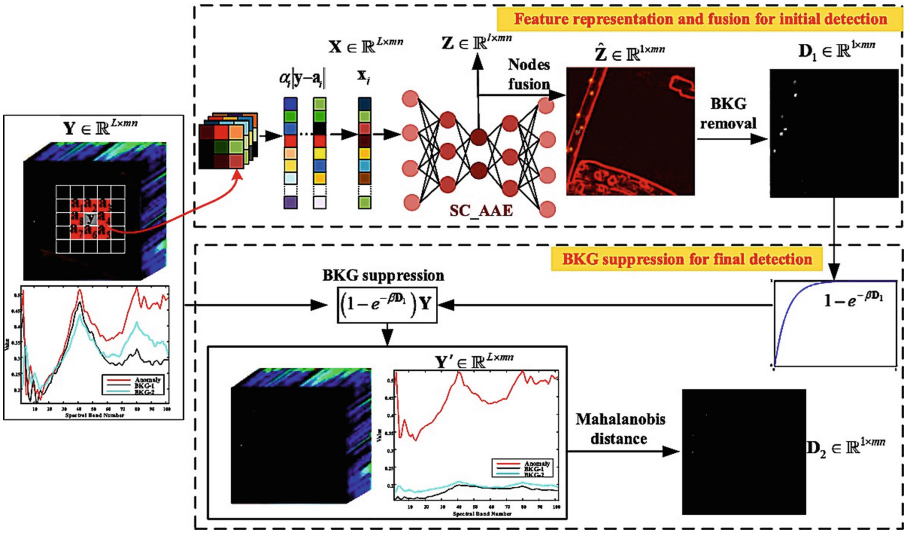

Fig. 11. Proposed SC_AAE-based anomaly detection method described in HSI [79].

Because each hidden node contributes to anomaly detection differently, the method is combined with an adaptive weighting approach to give capacity. BKG removal, in addition to feature identification, is critical for success in anomaly detection since it is an effective method for maximizing the distance between the anomaly and the BKG. The fused node is utilized to create a two-layer design that decreases BKG while preserving anomalous characteristics. Finally, this study contributes to four significant contributions: (1) A SC AAE anomaly detection framework that prioritizes detection while limiting false alarm rates; and (2) a WGAN-GP-based SC AAE that performs the spectral mapping from a high-dimensional spectral input vector to low-dimensional low profiles. The method devises a bi-layer architecture that reduces BKG while boosting anomalous properties. The proposed SC AAE-based anomaly detection technique is divided into

four phases: The projected SC AAE is utilized to represent features, with the caveat that anomalies are sometimes injected into the local smooth BKG. The first map BKG elimination, uses node fusion and the constructed non-linear function.

RL, a space-based HSI with L spectral bands and pixels, is represented by Y. Y = [y1, y2,…,] may be expressed as an L-dimensional vector, for example. Y = [Y1, Y2,… Y =], YL], may also be seen as a collection of L 2D photographs. The authors propose SC AAE, a robust feature representation technique for anomaly detection that differentiates the fundamental properties that induce. The suggested SC AAE technique fully leverages spectrum information and adequately reflects the properties of a wide spectral vector by using a spectral restriction loss. The hidden nodes are joined together to help in the discovery process. A two-layer technique is developed based on hidden node fusion to minimize BKG volatility while preserving anomalies.

By taking advantage of a considerable difference between BKG and anomaly, the suggested strategy would outperform current techniques in terms of efficiency. The method also compared the benefits of AAE against AE for identifying abnormalities. Additional testing in the real world demonstrates that the proposed SC AAE anomaly detection technique applies to a diverse set. The methodology shown in the illustration, which indicates that our suggested method is especially promising in monitoring and safety management, may reveal anomalies in certain bands that would otherwise go unnoticed. They method also intends to add geographic data to the SC AAE in the future.

The better the detection, the higher the AUC value of (P_D, P_F) and the lower the AUC value of (P_D, P_F) (P_F, τ). The AUC values of (P_D, P_F) and (P_F, τ) for the test HSIs are shown. (P_D, P_F) has an optimal AUC of 1, while (P_F, τ) has an optimal AUC of 0. The results shows that in all instances, the proposed technique and the STGF method are close to the ideal value, demonstrating that the SC AAE and STGF approaches can maintain detection capabilities (0.993251 and 0.997928 on average for the SC AAE and STGF methods, respectively). SC AAE can detect more anomalies than SC AE (0.977680 on average), proving AAE's superiority in hyperspectral anomaly detection.

As previously indicated, the AUC value of (P_F, τ) is utilized to measure the efficiency of BKG suppression. The results demonstrate that the recommended strategy results in reduced AUC values on average, showing that it suppresses BKG effectively. The AUC of (P_F, τ) for the recommended SC AAE approach is 0.013242, which is much lower than the 0.021113 (SC AE method) and 0.038077 (second and third best methods, respectively) (STGF method). Despite the fact that the STGF and SC AAE methods have similar detection accuracy, the STGF technique has around 2.87 times the false alarm rate of the SC AAE method. As a consequence, the proposed strategy reduces false alarms while preserving detection. Furthermore, although SC AE's performance is usually consistent, its AUC values aren't the best. The results show a strong correlation between detection maps and AUC ratings. As a consequence, we may deduce that the proposed technique is capable of detecting HSI anomalies.

4 Future Directions

Because anomalies often include a huge amount of data, understanding the difficult problem of detecting anomalies in moving data streams [25] is essential. Recognizing

data streams with limited memory and time, updating data as it comes, and retaining data in a dynamic way to capture fundamental changes while recognizing them are all examples of external detection challenges [29]. Data evolution algorithms are those that adjust their setup and parameters over time and in response to fresh data. Detection methods, unlike static data, have a hard time adapting to dynamic situations like the ever-changing IoT domain [64]. In addition, the great majority of existing systems are inadequate at detecting anomalies in data streams and have very basic capabilities [15]. In the IoT data stream environment, which is recognised for its continuously changing features, the detection accuracy of anomalies is poor, and the falsepositive rate is high [43]. In the context of IoT anomaly detection, the dynamic data stream is a problem that must be handled [24, 65]. Dealing with the difficulty of anomaly detection with a feature-evolving data source is another stumbling block. The issue is that data, as well as its quality, deteriorates with time. On the other hand, new and old data dimensions appear and disappear throughout time. Outlier detection in IoT systems where sensors alternately turn on and off (indicating the number of dimensions) [31] is an interesting topic with many applications. Because of the short data processing time based on fixed interval timing [59], the accuracy (windowing) is reduced. Because the majority of available approaches employ fixed interval timing, identifying the appropriate frequency for retraining the models is also a challenge [59, 66].

Ensemble approaches are well recognized for their ability to improve anomaly detection by detecting and running the accuracy of time [41]. Ensemble deviation detection is another fascinating area of study, with the potential to greatly improve algorithm detection accuracy. For resolving undiscovered areas, more specialized models are suggested. For finding anomalies in the data stream's environment, preliminary ensemble studies are advised. However, this field of research is largely untapped, necessitating the construction of more complete models. There are many existing IoT anomaly detection challenges that must be solved. Because anomalies do not often occur, labeled data availability is a major barrier in IoT anomaly identification. Obtaining real system data is likewise time-consuming and time-consuming [19]. Between formalizing the acquisition of knowledge logs and sensory data flow, developing a model, and testing it in real-world settings, there is a significant gap. Throughout the evaluation, many tests were carried out, the bulk of them were connected to the system's usual functioning [19].

The most advanced systems are based on typical behavior training, with anything that deviates from the norm being considered abnormal. To deal with complex datasets from real-world scenarios, more precise and reliable procedures are necessary. Furthermore, while training and assessing real-time anomaly detection algorithms, the availability of a good dataset for public anomaly detection is often a critical factor [68]. To avoid the creation of new forms of anomalous behavioral hazards, such databases must include a broad range of new normal and anomal behaviours, and they must be appropriately labeled and updated on a regular basis. The great majority of anomaly detection datasets is mislabeled, lack attack variety, and are unsuitable for real-time detection [69]. A realistic context with a range of normal and abnormal occurrences is required for new data sets for anomaly detection. Furthermore, while evaluating a new anomaly detection system, the key truth that integrates anomalies must be produced in order to boost the dataset's trustworthiness. Data complexity, which includes skewed datasets, unexpected

sounds, and data redundancy [40], is one of the most challenging difficulties in creating an anomaly detection algorithm.

For gaining meaningful information and knowledge, well-developed methodologies for curating datasets are essential. The choosing of an acceptable set of model parameters for anomaly identification is hampered by the fact that IoT data streams are often created from non-stationary settings with no previous knowledge of the data distribution [25]. The anomaly analysis display indicated a hole. For the use of visual system analysis, new methodologies and solutions are required. As a result, the flaws in the anomaly detection approach must be investigated [8]. Light, temperature, humidity, noise, electric current, voltage, and power are just a few of the environmental elements that IoT sensors and devices exhibit in their data streams [28]. Such a data stream demands speedy processing in order to handle urgent and severe circumstances, such as patient monitoring and environmental safety monitoring. With a large number of connected devices, a common data processing infrastructure to handle billions of incoming events per day may be required [71].

The daily inflow of vast amounts of data is a significant component of the data stream, necessitating real-time algorithm execution. However, since accuracy and time complexity are always a trade-off, the time complexity of identifying anomalies would be a major concern [14, 72, 73]. Despite learning algorithms' ability to identify and categorize anomalous behavior in real time, they must be tweaked to increase accuracy, such as by lowering the rate of false positive detection, particularly in large-scale sensor networks. Because many algorithms lose efficiency when dealing with large amounts of data, scalability is another important feature for anomaly detection systems. When dealing with high-dimensional data, most existing data stream techniques for anomaly detection lose their efficacy [25]. As a result, current models will need to be tweaked in order to identify outliers more consistently and efficiently. When a large number of features are recognised, a cluster of outliers in a restricted number of dimensions can appear at any moment. This collection of outliers seems to be natural in terms of the numerous subgroup dimensions and/or time period. Anomaly detection algorithms have a tougher difficulty discovering the most essential data characteristics due to the large number of variables [37]. As a result, when selecting the most significant qualities to characterize the whole data set, feature reduction is necessary.

5 Conclusions

As wide range of industries grow more automated (e.g. industrial warehousing [84], textile industries [85], Human Resources activities [86], supply chain in general [87] and the connectivity technologies advance, a wide range of systems are generating massive amounts of data. The huge amount of data has driven principal indicators method development for the entire system state modeling have been developed. The principal indicators are used to prevent potential accidents and economic losses through detection of anomalies and outliers as signs of possible near future equipment failure, system crash, human actions errors etc.

In the anomaly detection field, the multivariate time series data is especially experienced to be highly complex task due to the simultaneous consideration of temporal

dependencies and variables cross relationships matters. Deep Learning methods are especially adept at detecting anomalies and constructing unsupervised representations of large-scale data sequences. The great majority of them, however, are focused on a specific use case and need domain knowledge to develop. Because of the historical interest in anomaly detection in time-series data, we briefly explored various traditional approaches and uncovered significant issues in this domain. This research work has explored the anomaly detection in time series context and explained the popular frameworks used in real-world applications. The need for unsupervised deep learning-based time series anomaly detection continues as the system's complexity grows, yet the refined data and labels for analysis remain insufficient. Finally, we also describe how we can appropriately select the model and the training strategy for deep learning-based anomaly detection.

References

1. Ghoreishi, M., Ari, H.: Key enablers for deploying artificial intelligence for circular economy embracing sustainable product design: three case studies. In: AIP Conference Proceedings. vol. 2233, issue 1. AIP Publishing LLC (2020)
2. Yuwono, M., Moulton, B.D., Su, S.W., Celler, B.G., Nguyen, H.T.: Unsupervised machine-learning method for improving the performance of ambulatory fall-detection systems. BioMed. Eng. OnLine **11**(1), 1–11 (2012)
3. Inoue, J., Yamagata, Y., Chen, Y., Poskitt, C.M., Sun, J.: Anomaly detection for a water treatment system using unsupervised machine learning. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE (2017)
4. Jaeger, S., Fulle, S., Turk, S.: Mol2vec: unsupervised machine learning approach with chemical intuition. J. Chem. Inf. Model. **58**(1), 27–35 (2018)
5. Usmani, U.A., Watada, J., Jaafar, J., Aziz, I.A., Roy, A.: A reinforcement learning algorithm for automated detection of skin lesions. Appl. Sci. **11**(20), 9367 (2021)
6. Usmani, U.A., Roy, A., Watada, J., Jaafar, J., Aziz, I.A.: Enhanced reinforcement learning model for extraction of objects in complex imaging. In: Arai, K. (ed.) Intelligent Computing. LNNS, vol. 283, pp. 946–964. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-80119-9_63
7. Hu, W., Rajiv, R.P.S., Richard, T.S.: Discovering phases, phase transitions, and crossovers through unsupervised machine learning: a critical examination. Phys. Rev. E **95**(6), 062122 (2017)
8. Cai, Y., Guan, K., Peng, J., Wang, S., Seifert, C., Wardlow, B., Li, Z.: A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach. Remote Sens. Environ. **210**, 35–47 (2018)
9. Ayodele, T.O.: Types of machine learning algorithms. New Adv. Mach. Learn. **3**, 19–48 (2010)
10. Chandola, V., Mithal, V., Kumar, V.: Comparative evaluation of anomaly detection techniques for sequence data. In: 2008 Eighth IEEE international conference on data mining. IEEE (2008)
11. Lane, T., Brodley, C.E.: Temporal sequence learning and data reduction for anomaly detection. ACM Trans. Inf. Syst. Secur. (TISSEC) **2**(3), 295–331 (1999)
12. Eskin, E.: Anomaly detection over noisy data using learned probability distributions. In: Proceedings of the International Conference on Machine Learning, pp. 255–262. Morgan Kaufmann (2000)
13. Shon, T., Kim, Y., Lee, C., Moon, J.: A machine learning framework for network anomaly detection using SVM and GA. In: Proceedings from the sixth annual IEEE SMC information assurance workshop. IEEE (2005)

14. Lane, T.D.: Machine Learning Techniques for the Computer Security Domain of Anomaly Detection. Purdue University (2000)
15. Lane, T., Carla, E.B.: An application of machine learning to anomaly detection. In: Proceedings of the 20th National Information Systems Security Conference, vol. 377. Baltimore, USA (1997)
16. Shon, T., Moon, J.: A hybrid machine learning approach to network anomaly detection. Inf. Sci. **177**(18), 3799–3821 (2007)
17. Usmani, U.A., Haron, N.S., Jaafar, J.: A natural language processing approach to mine online reviews using topic modelling. In: Chaubey, N., Parikh, S., Amin, K. (eds.) COMS2 2021. CCIS, vol. 1416, pp. 82–98. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-76776-1_6
18. Usama, M., et al.: Unsupervised machine learning for networking: techniques, applications and research challenges. IEEE Access **7**, 65579–65615 (2019)
19. Maruhashi, K., Guo, F., Faloutsos, C.: Multiaspectforensics: Pattern mining on large-scale heterogeneous networks with tensor analysis. In: 2011 International Conference on Advances in Social Networks Analysis and Mining. IEEE (2011)
20. Goernitz, N., Kloft, M., Rieck, K., Brefeld, U.: Toward supervised anomaly detection. J. Artif. Intell. Res. **46**, 235–262 (2013)
21. Usmani, U.A., Watada, J., Jaafar, J., Aziz, I.A., Roy, A.: Particle swarm optimization with deep learning for human action recognition. Int. J. Innovative Comput. Inform. Control **17**(6), 1843–1870 (2021)
22. Choudhary, T., Bhuyan, M.K., Sharma, L.N.: Orthogonal subspace projection based framework to extract heart cycles from SCG signal. Biomed. Signal. Process. Control **50**, 45–51 (2019)
23. Zhang, Q., Yang, Y., Ma, H., Wu, Y.N.: Interpreting CNNS via decision trees. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
24. Yi, X., Zhou, H., Zhang, Z., Xiong, S., Yang, K.: X-rays-optimized delivery of radiolabeled albumin for cancer theranostics. Biomaterials **233**, 119764 (2020)
25. Jaitly, N., Hinton, G.: Learning a better representation of speech soundwaves using restricted boltzmann machines. In: 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE (2011)
26. Zhang, N., Ding, S., Zhang, J., Xue, Y.: An overview on restricted boltzmann machines. Neurocomputing **275**, 1186–1199 (2018)
27. Salakhutdinov, R, Mnih, A., Hinton, G.: Restricted Boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning (2007)
28. Papa, J.P., Rosa, G.H., Marana, A.N., Scheirer, W., Cox, D.D.: Model selection for discriminative restricted boltzmann machines through meta-heuristic techniques. J. Comput. Sci. **9**, 14–18 (2015)
29. Tanaka, M., Okutomi, M.: A novel inference of a restricted boltzmann machine. In: 2014 22nd International Conference on Pattern Recognition. IEEE (2014)
30. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B.: Adversarial autoencoders. arXiv preprint arXiv:1511.05644 (2015)
31. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning (2008)
32. Burda, Y., Grosse, R., Salakhutdinov, R.: Importance weighted autoencoders. arXiv preprint arXiv:1509.00519 (2015)
33. Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)

34. Gregor, K., Danihelka, I., Graves, A., Jimenez Rezende, D., Wierstra, D.: Draw: A recurrent neural network for image generation. In: International Conference on Machine Learning. PMLR (2015)
35. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
36. Sun, Y., Liu, Y., Wang, G., Zhang, H.: Deep learning for plant identification in natural environment. Comput. Intell. Neurosci. **2017**, 1–6 (2017)
37. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML 2011: Proceedings of the 28th International Conference on International Conference on Machine Learning, June 2011, pp. 689–696, Bellevue, Washington, USA (2011)
38. Schmidhuber, J.: Deep learning in neural networks: an overview. Neural Netw. **61**, 85–117 (2015)
39. Guo, X., Chen, L., Shen, C.: Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. Measurement **93**, 490–502 (2016)
40. Lo, S.-C.B., Chan, H.-P., Lin, J.-S., Li, H., Freedman, M.T., Mun, S.K.: Artificial convolution neural network for medical image pattern recognition. Neural Networks **8**(7–8), 1201–1214 (1995)
41. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 346–361. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_23
42. Boureau, Y.-L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: Proceedings of the 27th International Conference on Machine Learning (ICML-10) (2010)
43. Mundlak, Y.: On the pooling of time series and cross section data. Econometrica **46**(1), 69–85 (1978)
44. Sahiner, B., Heang-Ping Chan, N., Petrick, D.W., Helvie, M.A., Adler, D.D., Goodsitt, M.M.: Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. IEEE Trans. Med. Imaging **15**(5), 598–610 (1996)
45. Mishkin, D., Sergievskiy, N., Matas, J.: Systematic evaluation of convolution neural network advances on the imagenet. Comput. Vis. Image Underst. **161**, 11–19 (2017)
46. Traore, B.B., Kamsu-Foguem, B., Tangara, F.: Deep convolution neural network for image recognition. Ecol. Inform. **48**, 257–268 (2018)
47. Jianqiang, Z., Xiaolin, G., Xuejun, Z.: Deep convolution neural networks for twitter sentiment analysis. IEEE Access **6**, 23253–23260 (2018)
48. Sonnhammer, E.L.L., Von Heijne, G., Krogh, A.: A hidden Markov model for predicting transmembrane helices in protein sequences. Proc. Int. Conf. Intell. Syst. Mol. Biol. **6**, 175–182 (1998)
49. Krogh, A., Larsson, B., von Heijne, G., Sonnhammer, E.L.L.: Predicting transmembrane protein topology with a hidden markov model: application to complete genomes. J. Mol. Biol. **305**(3), 567–580 (2001)
50. Bahl, L,. Brown, P., de Souza, P., Mercer, R.: Maximum mutual information estimation of hidden Markov model parameters for speech recognition. In: ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 11, pp. 49–52. IEEE (1986)
51. Valdes, A., Macwan, R., Backes, M.: Anomaly detection in electrical substation circuits via unsupervised machine learning. In: 2016 IEEE 17th International Conference on Information Reuse and Integration (IRI). IEEE (2016)
52. Mohd, R.Z.A., Zuhairi, M.F., Shadil, A.Z.A., Dao, H.: Anomaly-based nids: A review of machine learning methods on malware detection. In: 2016 International Conference on Information and Communication Technology (ICICTM), pp. 266–270. IEEE (2016)

53. Dhivyaprabha, T.T., Subashini, P., Krishnaveni., M.: Computational intelligence based machine learning methods for rule-based reasoning in computer vision applications. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE (2016)
54. Karimipour, H., Dehghantanha, A., Parizi, R.M., Choo, K.-K.R., Leung, H.: A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids. IEEE Access **7**, 80778–80788 (2019)
55. Mohammadi, S., Mirvaziri, H., Ghazizadeh-Ahsaee, M., Karimipour, H.: Cyber intrusion detection by combined feature selection algorithm. J. Inf. Secur. Appl. **44**, 80–88 (2019)
56. Sakhnini, J., Karimipour, H., Dehghantanha, A.: Smart grid cyber attacks detection using supervised learning and heuristic feature selection. In: 2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE). IEEE (2019)
57. Karimipour, H., Dinavahi, V.: Robust massively parallel dynamic state estimation of power systems against cyber-attack. IEEE Access **6**, 2984–2995 (2017)
58. Bhatia, R., Benno, S., Esteban, J., Lakshman, T.V., Grogan, J.: Unsupervised machine learning for network-centric anomaly detection in IoT. In: Proceedings of the 3rd ACM CoNEXT Workshop on Big Data, Machine Learning and Artificial Intelligence for Data Communication Networks (Big-DAMA'19), pp. 42–48. Association for Computing Machinery, New York, NY, USA (2019)
59. Zhang, C., et al.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. Proc. AAAI Conf. Artif. Intell. **33**, 1409–1416 (2019)
60. Song, D., Xia, N., Cheng, W., Chen, H., Tao, D.: Deep r-th root of rank supervised joint binary embedding for multivariate time series retrieval. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018)
61. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)
62. Tariq, S., et al.: Detecting anomalies in space using multivariate convolutional LSTM with mixtures of probabilistic PCA. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019)
63. Yuguang, F., Peng, C., Gomez, F., Narazaki, Y., Spencer, B.F.: Sensor fault management techniques for wireless smart sensor networks in structural health monitoring. Struct. Control Health Monit. **26**(7), e2362 (2019)
64. Tsai, F.-K., Chen, C.-C., Chen, T.-F., Lin, T.-J.: Sensor abnormal detection and recovery using machine learning for IoT sensing systems. In: 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA). IEEE (2019)
65. Struye, J., Latré, S.: Hierarchical temporal memory and recurrent neural networks for time series prediction: an empirical validation and reduction to multilayer perceptrons. Neurocomputing **396**, 291–301 (2020)
66. Le, T.A., Nguyen, H., Zhang, H.: EvalSVC—an evaluation platform for scalable video coding transmission. In: IEEE International Symposium on Consumer Electronics (ISCE 2010). IEEE (2010)
67. Bandaragoda, T., et al.: Artificial intelligence based commuter behaviour profiling framework using Internet of things for real-time decision-making. Neural Comput. Appl. **32**(20), 16057–16071 (2020)
68. Khan, S., Liew, C.F., Yairi, T., McWilliam, R.: Unsupervised anomaly detection in unmanned aerial vehicles. Appl. Soft Comput. **83**, 105650 (2019)
69. Albusac, J., Vallejo, D., Jimenez-Linares, L., Castro-Schez, J.J., Rodriguez-Benitez, L.: Intelligent surveillance based on normality analysis to detect abnormal behaviors. Int. J. Patt. Recogn. Artif. Intell. **23**(07), 1223–1244 (2009)
70. Kind, A., Stoecklin, M., Dimitropoulos, X.: Histogram-based traffic anomaly detection. IEEE Trans. Netw. Serv. Manag. **6**(2), 110–121 (2009)

71. Huang, J., Li, J., Yu, D., Deng, L., Gong, Y.: Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 7304–7308. IEEE (2013)
72. Shimamura, Y.: Mixed text and image data processing. US Patent 5,204,946, 20 Apr 1993
73. Walker, J.R., Marmora Jr., A.J., Cheek, R.D.: Filtering method to reduce pixel density. US Patent 6,707,572, 16 Mar 2004
74. Takahashi, K.: Print device capable of printing a format sheet in which items about a print device and a document processor are completed. US Patent 5,502,796, 26 Mar 1996
75. Nawaratne, R., Alahakoon, D., De Silva, D., Xinghuo, Y.: Spatiotemporal anomaly detection using deep learning for real-time video surveillance. IEEE Trans. Ind. Inf. **16**(1), 393–402 (2020)
76. Zhao, W., Shihong, D.: Spectral–spatial feature extraction for hyperspectral image classification: a dimension reduction and deep learning approach. IEEE Trans. Geosci. Remote Sens. **54**(8), 4544–4554 (2016)
77. Li, W., Abtahi, F., Zhu, Z.: A deep feature based multi-kernel learning approach for video emotion recognition. In: Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (2015)
78. Touati, R., Mignotte, M., Dahmane, M.: Anomaly feature learning for unsupervised change detection in heterogeneous images: a deep sparse residual model. IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing **13**, 588–600 (2020)
79. Wen, L., Gao, L., Li, X.: A new deep transfer learning based on sparse auto-encoder for fault diagnosis. IEEE Trans. Syst. Man Cybern. Syst. **49**(1), 136–144 (2017)
80. Lee, H., Battle, A., Raina, R., Ng, A.: Efficient sparse coding algorithms. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems, vol. 19. MIT Press (2006)
81. Xie, W., Lei, J., Liu, B., Li, Y., Jia, X.: Spectral constraint adversarial autoencoders approach to feature representation in hyperspectral anomaly detection. Neural Netw. **119**, 222–234 (2019)
82. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011. LNCS, vol. 6791, pp. 52–59. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21735-7_7
83. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
84. Minashkina, D., Happonen, A.: Decarbonizing warehousing activities through digitalization and automatization with WMS integration for sustainability supporting operations. E3S Web of Conf. **158**, 1–7 (2020). https://doi.org/10.1051/e3sconf/202015803002
85. Ghoreishi, M., Happonen, A.: The case of fabric and textile industry: The emerging role of digitalization, internet-of-Things and industry 4.0 for circularity. In: Yang, X.-S., Sherratt, S., Dey, N., Joshi, A. (eds.) Proceedings of Sixth International Congress on Information and Communication Technology: ICICT 2021, London, Volume 3, pp. 189–200. Springer Singapore, Singapore (2022). https://doi.org/10.1007/978-981-16-1781-2_18
86. Hämäläinen, H., Happonen, A., Salmela, E.: CPFR-technology and automated data flows in technical wholesale supply chain of finnish machinery industry. In: The 3rd International Congress on Logistics and SCM Systems (ICLS 2007), vol. 28, pp. 279–286 (2007). https://doi.org/10.5281/zenodo.3377590
87. Vatousios, A., Happonen, A.: Transforming HR and improving talent profiling with qualitative analysis digitalization on candidates for career and team development efforts. Intell. Comput. **283**, 1149–1166 (2022). https://doi.org/10.1007/978-3-030-80119-9_78
88. Usmani, U.A., Usmani, M.U.: Future market trends and opportunities for wearable sensor technology. Int. J. Eng. Technol. **6**, 326–330 (2014)

89. Arbelle, A., Riklin Raviv, T.: Microscopy cell segmentation via adversarial neural networks. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018). IEEE (2018)
90. Wang, Z., Li, C., Wang, X.: Convolutional neural network pruning with structural redundancy reduction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
91. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018)