

Empowering anomaly detection algorithm: a review

Muhammad Yunus Iqbal Basheer¹, Azliza Mohd Ali¹, Rozianawaty Osman¹,
Nurzeatul Hamimah Abdul Hamid¹, Sharifalillah Nordin¹, Muhammad Azizi Mohd Ariffin¹,
José Antonio Iglesias Martínez²

¹College of Computing, Informatics and Mathematics, Universiti Teknologi MARA, Shah Alam, Malaysia

²Department of Computer Science and Engineering, Universidad Carlos III de Madrid, Madrid, Spain

Article Info

Article history:

Received Aug 25, 2022

Revised Jan 18, 2023

Accepted Mar 10, 2023

Keywords:

Algorithm

Anomaly detection

Autonomous

Real-time

Streaming data

ABSTRACT

Detecting anomalies in a data stream relevant to domains like intrusion detection, fraud detection, security in sensor networks, or event detection in internet of things (IoT) environments is a growing field of research. For instance, the use of surveillance cameras installed everywhere that is usually governed by human experts. However, when many cameras are involved, more human expertise is needed, thus making it expensive. Hence, researchers worldwide are trying to invent the best-automated algorithm to detect abnormal behavior using real-time data. The designed algorithm for this purpose may contain gaps that could differentiate the qualities in specific domains. Therefore, this study presents a review of anomaly detection algorithms, introducing the gap that presents the advantages and disadvantages of these algorithms. Since many works of literature were reviewed in this review, it is expected to aid researchers in closing this gap in the future.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Azliza Mohd Ali

College of Computing, Informatics and Mathematics, Universiti Teknologi MARA

40450 Shah Alam, Selangor, Malaysia

Email: azliza@tmsk.uitm.edu.my

1. INTRODUCTION

Technological advancement and the Internet can significantly affect human activities [1]. As such, the sustainability of the modern industrial era created an urban area requiring camera surveillance systems to secure the targeted public places [2], installing internet of things (IoT) sensors and using smart devices. Streaming data produced daily [1] are stored as big data [3]. Since big data consists of volume, variety, and velocity, data have to be autonomously processed for information and knowledge [4] to benefit the users. Most surveillance cameras or sensor data are classified as normal data behavior. While abnormal data in some situations could provide the user with some information to solve problems related to the case.

Detecting anomalies or unidentified events is crucial and tedious since big data gets too big. Hence, the extraction of wrong data produces faulty information. However, faster and more efficient data processing is needed for real-time data. Therefore, anomaly detection is significant in solving abnormal behaviors while streaming or in real-time data. Many researchers have begun expanding their research on inventing new algorithms for anomaly detection. For instance, Rettig *et al.* [5] created an online anomaly detection in big data, Costa *et al.* [6] created fault detection in a recursive way which is memory efficient, Bose *et al.* [7] detected anomalies using driving patterns, Dharmadhikari and Kolhe [8] used heterogeneous detectors of the anomaly using association rule, while Ali and Angelov [9] used heterogeneous data to detect the abnormality.

Due to time restrictions, algorithms that were proposed between the years 2010 and 2022 only were utilized in this study. Then, the suitable algorithms are selected randomly. Hence, many other studies on

algorithms created in different domains were not mentioned in this study. The availability of many anomaly detection algorithms leads to the need to evaluate and identify the efficient algorithm from the lot. This includes whether they can detect all anomalies in the data world. In [10] demonstrated the availability of nine basic types of anomalies, which consisted of 61 subtypes of anomalies.

Such high numbers in the types of anomalies could raise a valid question as to whether there is an algorithm to date be able to detect all these anomalies. Besides that, data are heterogeneous and are produced in various forms, including images, signals, and videos [3]. These data are also available both online and offline [9]. However, the crucial part of an algorithm is detecting data from online or streaming data because the algorithm that analyses streaming data cannot store data in memory due to limited memory space [11], dynamic data changes in the pipeline [5], dependency on other data [12], and demand faster processing to react when the data arrives.

This study presents a review of anomaly detection algorithms. The review focuses on thirteen algorithms developed by thirteen groups of researchers. Hence, in the future, researchers may evaluate the performance of their algorithm based on the criteria of the anomaly detection algorithm discussed in this study. This paper is prepared in six sections: Section 2 explains the methodology of the selection of literature. Section 3 describes the thirteen algorithms reviewed in this study. Section 4 presents the criterion of the anomaly detection algorithm. Section 5 discusses ways to close the gap in the thirteen algorithms. Finally, section 6 concludes this study with a holistic view.

2. METHODOLOGY

This section details the steps employed in conducting this review. These steps start with research questions which consist of several problems. Then, the keywords and literature are searched according to the needs of previous research questions. Finally, the knowledge from each literature is extracted and differentiated to understand the gap between the algorithms. Figure 1 illustrates the methodology. It consists of five essential steps. Each step is explained in the following subsections.

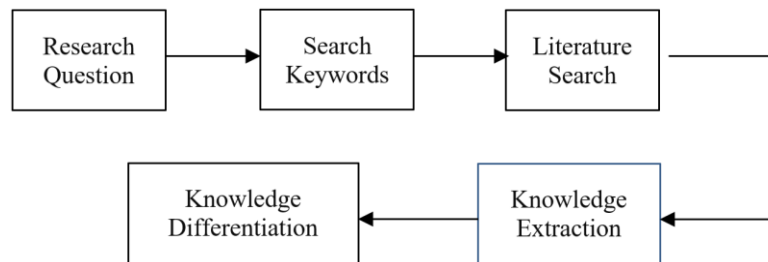


Figure 1. Review's step

2.1. Research questions

Since there are many types of anomalies [10], it is important to draft research question/s to aid with the methodology. So, how can the anomalies be detected when it involves complicated anomalies? Machine learning algorithms are widely used in different scopes. Krammer [4] proposed an algorithm to detect abnormality in a communication platform. Hence, the first question would revolve around the types of algorithms used to detect anomalies. Considering several related algorithms, a key question to be considered would be the criteria used to detect the abnormality. Thus, the second question in this study considered the criteria required by an anomaly detector. The final question would be related to determining the best algorithm to detect all the anomalies in the data world. The differences between the algorithms must be identified to determine the best among the selected algorithms. Thus, the following research questions were drafted in this study,

- Which algorithms were used to detect anomalies?
- What are the criteria an anomaly detector needs?
- What are the differences between the algorithm invented to detect anomalies?

2.2. Keyword and literature search

The databases used for this purpose include IEEE and Science Direct. Most of the research papers were retrieved from the IEEE database. Some journals provided more information than proceeding papers [13].

The following two criteria were used to filter the selected papers in the database,

- The algorithm was developed between 2010 and 2022. The algorithm must be new and unique. If the proposed algorithms were manipulated and differed from other invented anomaly detection algorithms, they will be considered in this study.
- The research article only described a newly invented anomaly detection algorithm and did not describe the application or mechanism of previously developed anomaly detection algorithms.

Figure 2 shows the evolution of keywords. It shows the keywords used with their respective result, which consist of selected research papers. These keywords include anomaly detection, heterogenous detector, and automatic detection.

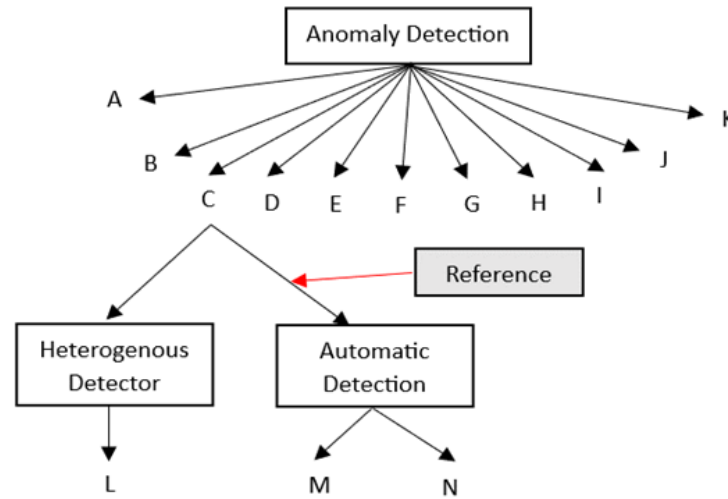


Figure 2. Evolution of keywords used

From the tree in Figure 2, ten papers can be considered using anomaly detection keywords. Research paper [c] [9] is not considered for this review since it does not introduce any new algorithm. However, manuscript title [c] is quite impressive (Anomalous behaviour detection based on heterogeneous data and data fusion) since the term heterogeneous data is used. Thus, the keyword was changed to “heterogeneous detector” to broaden the algorithm search. The paper [L] [8] is found using this keyword. In [c], the term *automatic* from reference is considered, which describes how to detect anomalous data without human intervention. Hence, from [c], the author uses [M] [14]. Further search on automatic detection found [N] [15].

Several articles were removed from the list as they did not meet the requirement. Among the reasons for rejection were: i) no new algorithm found [7], [11], [13]; ii) use empirical data analysis [9], [16]; iii) use previous anomaly detection modal and algorithm [17], [18]; iv) use the previous algorithm to detect anomaly without manipulation [5], and; v) used recursive density estimation, which introduces before 2010 [3]. Thus, only thirteen algorithms were selected to be included in this study.

2.3. Knowledge extraction and knowledge differentiation

The following two steps in this review (“Knowledge Extraction” and “Knowledge Differentiation”) were based not only on the understanding of the proposed algorithms in each of the papers (Section 3) but also on explaining each of the identified uniqueness. The information (i.e., /e.g., assumptions, recursive mechanism, automation, learning type) that was extracted from the different algorithms (Section 4) was influential in differentiating the advantages and disadvantages of the algorithms (Section 5). Finally, the best algorithm was chosen as part of this review.

3. ANOMALY DETECTION ALGORITHMS

3.1. Incremental spatio-temporal learner (ISTL)

ISTL [2] is an algorithm used specifically for real-time surveillance cameras. Figure 3 represents the ISTL approach. Firstly, video surveillance is injected into the ISTL as input represented as normal behavior. Next, the trained model acts as an anomaly detector and localizes the current input stream data. Human experts

then validate the trained algorithm before being aggregated into training data through the fuzzy aggregation method. Finally, the training data is used for other stream surveillance inputs.

As depicted in Figure 3, active learning was used to train the model with the user input continuously. The fuzzy aggregation model supports it to retain the stability of the iteration during learning. ISTL comprises a spatiotemporal autoencoder that will learn the motion of the video streams. The ISTL approach is finally able to detect anomalies with its respective localization.

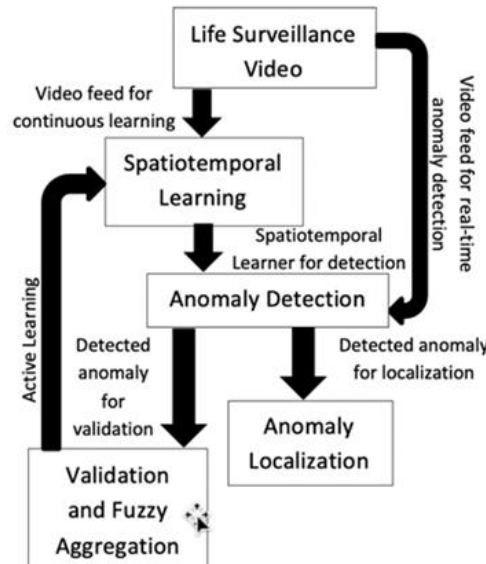


Figure 3. ISTL approach

3.2. Anomaly detection in online detection

This algorithm was used to detect anomalies in communication platforms by differentiating unwanted users in the platforms [4]. The algorithm consists of three phases-multiple canopy clustering, cluster membership analysis, and classification model training. Anomaly is declared when the number of records in the cluster is less than the anomaly threshold. There are four types of thresholds used which are max_density which represents the maximum density a clustering can have, seed_count for limiting how many times a cluster can repeat, significant_ratio for limiting the amount of data in a cluster and finally anomaly_thresholds to declare anomaly. For example, if the parameter is less than this threshold, it will be declared an anomaly. If both max_density and seed_count were set at the maximum level, the method stability can be increased but requires more computational time.

3.3. Anomaly extraction using association rule

This algorithm was built to detect anomalies in network traffic; however, the architecture can also be manipulated to suit other domains [8]. Anomaly extraction using association rules is performed to detect frequent patterns and create rules between them. The first step involves a pre-filter to determine suspicious flow, where it removes the maximum fraction of normal flow. Next, the association rule is employed through the Apriori algorithm. Finally, the heterogenous detector is used to identify the anomaly.

3.4. Eccentricity analysis

Typicality and eccentricity based on data analytics (TEDA) were built to solve a traditional statistical method that is unapplicable to apply in the real world [12]. This algorithm was proposed [12] and published [19] to build an anomaly detector based on the TEDA mechanism, which can be used in any domain. Moreover, TEDA does not use any prior assumptions [12]. Meanwhile, the first time assumption is realistic for the pure random process but not for the real-world process [12]. Even though there is a rationale for using thresholds, it also contains many disadvantages. The disadvantages are described in section 5.

The author proposed σ , which sometimes n represents 3 for anomaly detection, applicable in both TEDA and statistical analysis. In the traditional σ method, the mean and average represent all other data samples. A σ gap appears when data eccentricity becomes higher, declaring the presence of an anomaly. While

the “ ε vicinity” defines anomaly in a stream where the noise is smaller and the anomaly forms abnormal points. The proposed gap accumulated proximities and analyzed the two pairs of suspected regions. Meanwhile, the traditional method only uses average proximity, not including the distance between the outlier data points. The minutiae of the method used to detect anomalies are:

- Normalized eccentricity of the data point is calculated.
- The point with maximum normalized eccentricity keeps one by one, x^y , where $y = 1, 2, \dots, 3$.
- If $(\Delta\zeta^{1,2} > n/k)$ THEN x^1 is an anomaly, where k is the number of normalized eccentricity and $\Delta\zeta^{1,2}$ is calculated using (1).
- Else, if $(\Delta\zeta^{2,3} > n/k)$ THEN x^3 and x^2 are anomalies, where $\Delta\zeta^{2,3}$ is calculated using (2).
- If (3) is satisfied, the x^1 and x^2 are declared anomalies. In (3), μ represents the mean of the respective data.
- Otherwise, continue to check all the data, whether there are anomalies.
- End.

$$\Delta\zeta^{1,2} = \zeta(x^1) - \zeta(x^2) \quad (1)$$

$$\Delta\zeta^{2,3} = \zeta(x^2) - \zeta(x^3) \quad (2)$$

$$(x^1 - \mu_k^1)^T (x^1 - \mu_k^1) - (x^2 - \mu_k^2)(x^2 - \mu_k^2) > n\sigma_k^2 \quad (3)$$

3.5. Abnormal human events on train platforms

This algorithm is built to detect abnormal human events in train platforms using a surveillance camera [14]. However, detecting anomalous behavior using surveillance cameras is challenging since this event is probable. All information on size and shape is used to classify different objects and events. The algorithm used to identify train status is.

- a. Estimate motion vectors in the train bed area.
- b. If the motion is parallel to the edge of the bed track, then.
 - Analyze the motion vector to estimate the speed.
 - Check if the speed is more than the threshold.
 - Update the train status: the train is arriving, the train is departing, the train is discharging passengers, and the train bed is clear.

The line differentiating platforms from the train bed is set manually. The mean of the background image vectors includes unwanted noise viewed by camera sensors and other sources. Hence, the background image is assumed to be more probable than the foreground pixel.

The displacement of vectors in the image indicates the train speed. A high-speed train yields a high displacement value, while a low-speed train yields a low displacement value, whereas a halted train has a zero-displacement value. The detection of a moving object (anomaly) is turned on when no objects or trains are in the track bed. The entire video frame is utilized to determine any blob in the foreground area, which is assumed as an object. The alarm is raised when the detected object moves across the track bed, indicating a suspicious event.

3.6. Dangerous motion detector in human crowds

This is an algorithm to detect dangerous motions in crowded places [15]. It is very crucial to detect abnormal events such as stampedes. The algorithm includes,

- a. Calculate the dense optical flow and the corresponding two-dimensional flow of the motion direction and magnitude histogram.
- b. Averaging the histogram over a short time interval.
- c. An increase in lateral oscillations denotes congestion when using the front view camera as follows:
 - The histogram acts as a congestion indicator, showing motion that goes to the left and right, which shows oscillations.
 - The histogram indicates a high degree of symmetry by marking the area as highly dense.
 - The low value of symmetry indicates the area is congested.
- d. The alarm will be triggered if the result is more than the threshold.
- e. Thresholds, θ , are calculated based on the targeted area's current condition, which is a data-driven threshold.

Shock waves (i.e., anomaly) may occur during the congestion. A single people's movement will cause propagation to make others move in the region. The shock wave is dangerous because people will begin to follow it as they cannot control the movement causing some to fall and get crushed. The shock wave is defined as a sudden increase in the magnitude of optical flow. The standard deviation of direction involving the vicinity

becomes smaller when the shock wave is triggered. Shock waves are detected by comparing the current magnitude with the previous magnitude.

3.7. Transfer deep learning for hyperspectral image

This algorithm is used in images and involves a convolutional neural network-based detector (CNND) that evaluates the same class for similarity and a different class for dissimilarity [20]. The proposed detection involved three steps: i) learning a deep CNN in reference data; ii) measuring similarity between test data and train data; and iii) averaging the detection product as an outcome. Firstly, reference data is inserted with ground truth to generate differences (0-similarity, 1 dissimilarity) between pixels. An anomaly is declared if the output exceeds the threshold when comparing the detection output with the threshold set.

3.8. Improved RX with CNN framework

Reed-Xiaoli (RX) algorithm is used in the image and uses Mahala Nobis distance which considers the mean and covariance of the matrix [21]. It will assume the background point as normal. Meanwhile, the subspace RX (SSRX) algorithm deletes the background subspace and applies a detector on the target subspace. Hence, higher results are chosen in the algorithm to detect the target. However, anomaly detection becomes harder in small targets due to complicated backgrounds. Also, the RX algorithm can improve its performance by increasing its peak value. The three steps involved in detecting an anomaly include: i) Learn using chosen images from airborne visible/infrared imaging spectrometer (AVIRIS) hyperspectral algorithm; ii) The text pixel between the surroundings is compared using CNN to generate an approximation score; and iii) The approximation score is added to the central pixel and transferred to the RX algorithm for improvement and detection.

Then, the algorithm will produce a training dataset. The following steps are needed to produce a training dataset: i) Existing categories will act as training sets that involve manual selection of background and target, for example, road and vehicle, respectively; ii) Background and target class samples are paired and subtracted.

An evaluation score between -1 and 1 is obtained. Indicating the degree of similarity between the background and target. The greater the evaluation score from 0, the pixel is closer to abnormal targets and vice versa.

3.9. Autonomous anomaly detection

The empirical data analysis (EDA) proposed by Angelov *et al.* [1] is an extended version of TEDA [22]. Data in the real world is unknown and probably not labeled. EDA is based on observed data, and it accumulates properties without making any prior assumption. Based on EDA, autonomous anomaly detection (AAD) was created and can be used in any domain. Firstly, assuming $\{x_1, x_2, \dots, x_K\} \in \mathbf{R}^d$ where x_i is i^{th} data sample followed by K number of data samples. In this data, there will be also same data value available possibly more than one denoted by $\exists i \neq j | x_i = x_j$. The unique dataset is $\{u_1, u_2, \dots, u_L\} \in \mathbf{R}^d$ and its frequencies are $\{f_1, f_2, \dots, f_L\} \in \mathbf{R}^d$. The algorithm works.

- i. Firstly mean, μ , and average scalar product, X , is calculated using (4) and (5), respectively.

$$\mu = \frac{1}{K} \sum_{j=1}^K x_i \quad (4)$$

$$X = \frac{1}{K} \sum_{i=1}^K \|x_i\|^2 \quad (5)$$

- ii. Multimodal density is calculated using (6).

$$D^{MM} = f_i D^{UM}(u_i) = \frac{f_i}{\frac{\|u_i - \mu\|^2}{X - \|\mu\|^2}} \quad (6)$$

- iii. The product is ranked in ascending order $\{D^{MM}(x)\}$.
- iv. The first half of the $\frac{1}{n^2}$ of the smallest value from (ii) is selected and declared as potential anomalies $\{x\}_1^{PA}$. The n value is such as in eccentricity analysis, which is set to 3.
- v. D^{MM} is less sensitive to local sparsity. Hence, for the less sensitive D^{MM} , consider there are data $\{x_1, x_2, \dots, x_K\}$, and calculate the Euclidean distance between them using (7).

$$\bar{d} = \frac{\sum_{k=1}^K \pi(x_k)}{K^2} = 2(X - ||\mu||^2) \quad (7)$$

- vi. Next, each unique data sample can obtain hypersphere from $\frac{\bar{d}}{2}$. Data located inside this hypersphere are known as neighbors.
- vii. Consider the neighbors of u_i . Therefore, the neighbors of u_i are $\{u\}_i^L$. Accordingly, the unimodal value can be determined using (8), where η_i^L is the mean of $\{u\}_i^L$ and U_i^L is the average scalar product.

$$D^L(u_i) = \frac{1}{1 + \frac{\|u_i - \eta_i^L\|^2}{U_i^L - \|\eta_i^L\|^2}} \quad (8)$$

- viii. Next, unimodal density is weighted by its frequency using (9). The N_i in (9) represents the cardinality of the set $\{u\}_i^L$. Then, the unimodal product is then arranged in ascending order $\{D^{WL}(x)\}$. The second smallest value selected among them is declared as the second potential anomaly detected. $\{x\}_2^{PA}$.

$$D^{WL}(u_i) = \frac{(N_i-1)}{L} \cdot f_i \cdot D^L(u_i) \quad (9)$$

After that, the algorithm will determine whether the detected potential anomalies can form data clouds. This was done using the autonomous data partitioning (ADP) algorithm introduced by Gu *et al.* [23]. In the final stage, the algorithm will confirm whether the potential anomaly is actual. The potential anomaly will be declared as an anomaly if the potential anomaly cannot form any data clouds. In (11), if the data clouds support is less than average support, then it is formed by the anomaly. In both equations, S represents support or number of members in a data cloud, and c_i represents i^{th} data cloud.

$$IF (S_i < S_{average}) THEN (c_i \text{ is formed by anomalies}) \quad (10)$$

$$S_{average} = \frac{1}{N} \sum_{i=1}^N S_i \quad (11)$$

3.10. Hierarchical pattern matching

Hierarchical pattern matching (HPM) for anomaly detection uses a piecewise linear function to detect abnormal line fitting from streaming patterns [24]. This algorithm used a recursive mechanism which is the same as isolation forest. It is memory efficient since it only stores unique data patterns once, avoiding data redundancy. Firstly, the algorithm extract pattern from streaming data. The size of the scrolling window is predefined before the algorithm is executed. For each window, the best fitting line is found by using a piecewise linear function.

Furthermore, after finding the best fitting line, the algorithm will go through much dipper to find the best fitting again by using the previous piecewise linear function. As a result, a hierarchical tree is formed. In the anomaly detection phase, assume that the streaming data is running. Then the algorithm chooses the time window which contains a specific amount of data from the time series event. The data is compared with the previous hierarchical tree. If a new pattern is found, the alarm is raised, which means an anomaly pattern is detected.

3.11. Multi-aspect data stream anomaly detection

Multi-aspect data stream anomaly detection (MDS_AD) solves issues in many state-of-the-art algorithms [25]. For example, current anomaly detection algorithms overlook the relationship between attributes and the dynamic existence of data in a streaming environment. The salient issue is the current state-of-the-art algorithms do not fulfill multi-aspect requirements. Multi-aspect means each record has multi-type data, such as categorical and numerical.

Firstly, it used principal component analysis (PCA) to reduce dimensionality while preserving the correlations of each attribute. Then, it combines categorical and numerical data using one of the locality-sensitive hashing (LSH) functions called Record Hash. Record Hash can work in streaming data, enabling it to update the model faster. Then, the isolation forest algorithm is fetched, creating multiple trees. After model construction, the algorithm will receive online data.

The data will enter the algorithm along the time in the online anomaly detection phase. Firstly, the data entries will be reduced in their dimensionality using PCA. Then, the output from PCA will transverse along the modeled trees. Along the journey, the path length is calculated. The shorter the path length, the higher the anomaly score will be. The anomaly score is between 0 and 1. The anomaly score that is closer to one is

declared an anomaly. A threshold determines how many anomaly scores are needed for data to become an anomaly.

After that, the model update is conducted. The model is updated after a certain amount of online data is stored. After the model is updated, the new model is used to detect anomalies in the online stage. The stored data used to update the model is emptied. Then, the process is repeated, making the model evolve along the online dynamic environment.

3.12. Anomaly detection using an array of sliding windows and PDDS

This algorithm runs unsupervised in the streaming environment [26]. Firstly, the algorithm assumes the size of the sliding window, sub window, and a number of targets. Using the probability density function (PDF), probability density-based descriptors (PDD) are obtained for each sub window. Target is selected by partitioning the range between maximum and minimum windows. The midpoint of each interval is obtained. Using the midpoint, a set of PDD is obtained for each window.

Then, the distance between PDD is calculated. The more the distance, the more abnormal it is. Then, the expanded maximum distance is also calculated to determine how far the PDD of a sub window can go. If there are three PDDs, assume that there are fw1, fw2, and fw3. If the distance between fw1 and fw2 is more than the expanded maximum distance, and if the distance between fw2 and fw3 is less than or equal to average distance, the corresponding window will be declared an anomaly. Otherwise, it is normal.

3.13. Correlated anomaly detection from large streaming data

It is called correlated anomaly detection (CAD) [27]. It detects correlated anomalies in which the data have a stronger correlation with each other. Meanwhile, normal data do not correlate with each other. There are two new algorithms and a framework. It solved principal component degeneration. Principle component degeneration is, for example, when normal data is more than anomaly data making the anomaly data hard to detect. There are two algorithms: randomized principal score (rPS), which detects suspicious anomalies, and generative principal score (gPS), which detects suspicious and core anomalies.

The principal score is denoted as $\rho(X)$, where X is a sequence of large data. If $\rho > \tilde{\rho}$, there is a possibility of a correlation or anomalous data, which can trigger human attention. The threshold $\tilde{\rho}$ that decreasing can cause a false positive. The threshold is said to be set to 0.7, which is the most ideal threshold. There are also additional thresholds used in robotic process automation (rPA) and grade point average (gPA). For instance, rPA used P to control sampling quantity and correlation sensitivity. The gPA uses α , which should not be far from $\tilde{\rho}$.

It assumes many types of assumptions to build the algorithms. For assumption one, the normal data entries are weakly correlated, and if $\rho(X)$ is closer to one, then X contains anomalous data. Assumption two, there are many correlated normal data which should be anomalies but lower than the threshold $\tilde{\rho}$. Assumption three is when a significant quantity in the data vector is correlated, making it anomalies.

For example, botnet cases where a large quantity tries to enter the server to trigger a distributed denial-of-service (DDoS) attack. The gPS is introduced to tackle assumption four, where anomalous behavior tries to camouflage. Assumptions four is each anomaly set has higher internal correlations than external correlations. The gPS algorithm can detect anomalous sets that are weakly correlated. It solved the rPS algorithm, where it can possibly raise a false alarm. Then rPS and gPS form a framework that accepts entries from large data streams.

4. ANOMALY DETECTION CRITERION

In this section, the criteria required by each algorithm are further explained to differentiate between thirteen algorithms. These six criteria are very important in any anomaly detection algorithm. In addition, these criteria will help to detect anomalies in streaming data since it is dynamic or unknown [12], [19]. As a result, these criteria are believed to help researchers to invent the best anomaly detector. These criteria include: i) No assumptions; ii) Fast computational time; iii) Memory efficient; iv) Automation; v) Type of learning: Semi-supervised and Unsupervised; and vi) Ability to detect all types of anomalies in the data world.

Most of the assumptions in the traditional statistical method are impractical [1]. The assumption for the first time is realistic for the pure random process but not for the real-world process [12]. The assumptions widely used in artificial intelligence algorithms are also known as threshold values. For example, in deep learning, the linear perceptron is made of assumption, but the data labels are only approximated [28]. Hence, assumptions will only provide approximated values and not the exact value. Therefore, for a particular problem that requires thresholds and parameters, especially in industrial applications, assumptions are not suitable [6].

On the other hand, the fast computational time is significant since the algorithm needs to act whenever an anomaly is detected in the data. Recursive storage and update enable the system to operate faster and keep

a large dataset suitable for online streaming data [6]. Hence, whenever the algorithm uses recursive calculation, it is known to have these criteria [16].

- Computational efficient.
- Prevent vigorous usage of memory, which is not needed.
- Reuse and update important information in Fast computational time.

In other words, the recursive calculation can also reduce memory usage, allowing better use of memory consumption [1]. Memory efficiency is an important criterion for an anomaly detection algorithm. Streaming data involves incoming data that cannot be stored in the memory due to limited memory in the computer and simply processing the data since it comes in various forms [11]. Automation can reduce human intervention in decision making. The urban area is transforming into autonomous machinery where human intervention is not required [2].

For instance, human expertise cannot detect all anomalies in a specific video stream [2]. Since a lot of data arrives at every millisecond, autonomous anomaly detection can help reduce this dimensionality by focusing on small data only consisting of rare events compared to human expertise [9]. It does not mean having no human intervention at all because every piece of machinery needs a human touch to decide. This aspect is related to the prevention of mistakes and making intelligent machinery.

To make an algorithm more intelligent, learning is required. There are three types of learning in artificial intelligence, namely supervised learning, semi-supervised learning, and unsupervised learning. Supervised learning is more accurate and effective compared to statistical methods when dealing with anomalous data [22]. But, sometimes, in an anomalous world, data could be unknown or not labeled [22]. So, to detect an anomaly, only semi-supervised and unsupervised learning can be used. It is because supervised learning, like classification, needs labeled data [16]. The supervised and unsupervised method is usable in the semi-automated identification of potential threats [4].

As fully autonomous mentioned before, it does require any assumptions and training dataset [29]. In short, it means that it does not need to learn anomalous data; instead, it only captures normal data patterns to differentiate them. Since abnormal data does not fit in normal data [11], it deviates from normal data to form suspicious abnormal data [8]. Sometimes, normal data also contains anomalous data that is undetectable. Besides, the definition of normal behavior is currently hard to capture as this is one constraint to bring up anomaly detection algorithms [2], [3]. When unexplainable anomaly data is found, the old normal situation becomes wholly different [10]. Therefore, one cannot understand the types of anomalies without referring to the structure of the data [10]. Hence, an anomaly detection algorithm developer needs to understand the data structure to ensure the algorithm's performance.

5. EVALUATION

All the thirteen algorithms reviewed in section 3 were evaluated using the criteria discussed in section 4. Identifying whether all the algorithms can detect all types of anomalies is difficult since the algorithm needs to be tested first. But it is believed that EDA [1], which was further upgraded into the anomaly detector [22] can detect all anomalies [12]. Instead of the algorithm's speed and memory, the recursive calculation was used to differentiate algorithms as shown in Table 1. It is hard to evaluate speed and memory consumption in each algorithm since the authors did not mention them. While recursive storage and update enable a system to operate faster and store large datasets suitable for online streaming data [6].

Prior assumptions cannot be used to close the differentiation gap between all thirteen algorithms and to create a robust anomaly detection algorithm. Since the anomaly is unknown, one cannot simply assume or draw the line between anomaly and normal data. Besides that, supervised learning is inapplicable in this case. For example, human behavior is unknown and hard to predict, which sometimes changes according to their goal [30]. Hence, in this case, semi-supervised is the best learning method. Furthermore, autonomous anomaly detection is better since it does not require human expertise, as it could ease human life and prevent human errors.

To know whether the algorithm is autonomous or not, the algorithm should not have any assumptions and training datasets [29]. Based on Table 1, there are various ways of conducting this method to know whether the algorithm is automatic. Firstly, using the title. If the title contains the word automatic, it will be considered automatic. This includes autonomous anomaly detection [22], automatic detection of human events on train platforms [14], and automatic detection of dangerous motion behavior in human crowds [15].

Then, the algorithm can also be said as automatic if there is content in the introduced algorithm paper describing automatic. For example, ISTL [2] describes automating anomaly detection using deep learning. It uses spatial temporal learning with anomaly detection and localization. The transferred deep learning algorithm used CNND [20], which finds similarities and dissimilarities of images on its own to represent abnormality, making it an automatic algorithm. Eccentricity analysis [19] is entirely based on data and their distribution,

with no user specific thresholds and no kernel require, which further studies make it autonomous fault detection. Meanwhile, the RX algorithm [21] produces its own training dataset, making it an algorithm that does not need human intervention to add more data. Then, [24], [27] uses no training data, and no human intervention is necessary during operation.

Table 1. Algorithm differences based on criteria

	Assumption	Recursive mechanism	Automatic	Learning Type
Autonomous anomaly detection [22]	×	✓	✓	US
Eccentricity analysis [12]	✓	✓	✓	US
Anomaly detection in online detection [4]	✓	×	×	SS
Abnormal human events on train platforms [14]	✓	×	✓	US
Incremental spatio-temporal learner [2]	✓	×	✓	SS
Transfer deep learning for hyperspectral image [20]	✓	×	✓	SS
Dangerous motion detector in human crowds [15]	✓	×	✓	US
Anomaly extraction using association rule [8]	✓	×	×	US
Improved RX with CNN framework [21]	✓	×	✓	SS
Hierarchical pattern matching [24]	✓	✓	✓	US
Multi-aspect data stream anomaly detection [25]	✓	✓	×	SS
Anomaly detection using an array of sliding windows and PDDs [26]	✓	×	×	US
Correlated anomaly detection from large streaming data [27]	✓	×	✓	US

*Legend: US; unsupervised, SS; semi-supervised

5.1. Results

In this subsection, the result based on Table 1 is further explained. Firstly, autonomous anomaly detection does not need any prior assumptions. It brought EDA characteristics which utilize recursive updates such as in mean, average scalar product, and data density calculation. It is an automatic algorithm and unsupervised algorithm which does not need training or labeled data to detect anomaly. Eccentricity analysis was also implemented in streaming data [31]. It used a recursive update. Furthermore, it is automatic and uses unsupervised learning. Unfortunately, it needs assumptions or a threshold which, if the calculated normalized eccentricity is bigger than the calculated threshold, then it is an anomaly [31].

Anomaly detection in online detection was used in social chat with the aid of four thresholds. It does not utilize recursive updates and is not automatic. At the same time, it will label data detected as normal and abnormal and inject it into the machine learning algorithm. Therefore, it used semi-supervised learning. Abnormal human events detection in train platforms is not generic and only used in train platforms. It requires assumption. The algorithm will check abnormal events by using the speed of the train in the train bad and the threshold set. It does not use any recursive method; hence it is assumed not as speed and memory efficient as the algorithm that has it. But it is automatic and utilizes unsupervised learning.

ISTL is used in surveillance cameras which run automatically. It used a semi-supervised method of learning. It used the validated data from the experts, meaning the data was labeled. There is no recursive calculation used, and it requires assumption. For example, in evaluation, two thresholds are used, which are anomaly threshold and temporal threshold. Transferred deep learning for hyperspectral images is used in images using a convolutional based detector (CNND). It used threshold to declare a section of pixel on an image is anomaly or not. It is semi-supervised, where it uses reference data to generate ground truth. It does not have any recursive calculation and is automatic.

Dangerous motion detectors in human crowds are used to avoid stampedes and other dangerous events. It uses assumption. For example, the alarm will be raised if the histogram dense flow exceeds the threshold set. It does not have a recursive update. But it is fully automatic, reducing human intervention as well as using unsupervised learning.

Meanwhile, anomaly extraction using the association rule is not automatic. It is built especially for detecting anomaly events in network pipelines. It uses a heterogenous detector without any recursive calculation and requires assumption to detect the anomaly. But it learns in an unsupervised manner without any aid of labeled data from experts.

Then, improved RX with CNN framework was used to detect anomalies in an image. It uses threshold, and no recursive mechanism is found in the algorithm. It is automatic which requires no human intervention. But it used semi-supervised learning, generating many trainings dataset to use in the algorithm. The HPM algorithm uses thresholds such as predefined amount of data in a window. It uses recursive mechanism, the same as the isolation forest [32]. It is an automatic and unsupervised algorithm where training data is not required.

The MDS_AD algorithm uses assumptions to determine the degree of anomaly score. It uses isolation forest [32], which uses a recursive mechanism. Unfortunately, the model keeps evolving from time to time. It is not automatic and is a semi-supervised algorithm. The anomaly detection using an array of sliding windows and PDDs uses three assumptions. Firstly, increasing the number of windows will affect true and false positive scores. Then increasing the number of sub windows will increase true and false positives. Finally, increasing the number of targets will less affect the algorithm's performance. Therefore, assumptions affect the algorithm's performance. There is no recursive mechanism, and it is not automatic. It is also an unsupervised algorithm.

Finally, the correlated anomaly detection from large streaming data uses assumptions which can affect algorithm performance. Furthermore, they are built based on assumptions problems. In the future, anomalies existence may not know, which will make this algorithm fail. For example, a botnet may modify to attack normal users accessing the server. The normal user may mark it as an anomaly, whereas the access root is from another user trying to freeze the server operations. It does not use any recursive mechanism and is automatic. It is also an unsupervised algorithm.

5.2. Discussion

Hence, based on Table 1, autonomous anomaly detection [22] was the best algorithm to fulfill the requirement for the best anomaly detector. It is the only algorithm that does not use any assumptions. Meanwhile, eccentricity analysis [12] used comparison threshold to differentiate normal and abnormal states [31]. It also has a recursive, unsupervised, and fully automatic mechanism that detects anomalous data without human intervention.

Many additional algorithms could help close this gap apart from the reviewed algorithms. For example, autoencoders that could provide accurate input [33] and CNN-based features are preferred than other hand-crafted algorithms [34]. Additionally, explainable deep neural networks (xDNN) can upgrade the anomaly detection algorithm, combining reasoning and learning in a synergistic way [35]. Besides the training algorithm, both normal and abnormal data need to be balanced.

However, obtaining balanced data in the real world is difficult. But some anomaly detection algorithms can be used [36] to solve this issue. Therefore, a hybrid anomaly detection algorithm can be more powerful than a single anomaly detection algorithm to help close this gap quickly. For example, a hybrid algorithm can ease the burden of collecting balance data which becomes much fairer when training new anomaly detection algorithms. In other words, combining additional algorithms makes anomaly detection algorithms more reliable.

Finally, the anomaly detection algorithm can be improved by implementing all the criteria mentioned in this paper. As cybersecurity and IoT development thrive, anomaly detection is needed, especially in high-speed data. This is to make sure that the anomaly can be detected at the time it arrives. By using the autonomous system, which learns by itself [37] the dynamic existence of data [12], it can help in cybersecurity and IoT in detecting suspicious data.

6. CONCLUSION

This study introduced a review of algorithms related to anomaly detection. This review focused on algorithms that were developed from 2010 to 2022. Although there were other related algorithms designed during that period, six criteria were considered and discussed to select the appropriate algorithms. Hence, this study conducted a literature review for the thirteen algorithms along with the criteria needed for each anomaly detection algorithm to be applicable in the real world. As for the three research questions presented in this review, six criteria were presented to ensure the efficacy of an anomaly detection algorithm. In this sense, AAD was the only algorithm that had no assumptions compared to the other algorithm. This unique characteristic of EDA makes it suitable to be implemented in streaming data. As a recommendation, it will be much easier if an anomaly detection algorithm is implemented in devices to help detect unknown anomalies. It is also recommended for the anomaly detection algorithm be built based on the six criteria mentioned in this review. Consequently, it could reduce human intervention in detecting anomalies by detecting all possible anomalies instantly.

ACKNOWLEDGEMENTS

The authors would like to express gratitude to Institut of Graduate Studies and College of Computing, Informatics and Mathematics, Universiti Teknologi MARA for all the given support. The registration fees is funded by *Pembiayaan Yuran Penerbitan Artikel (PYPA)*, *Tabung Dana Kecemerlangan Pendidikan (DKP)*, Universiti Teknologi MARA. In addition, this work was supported under projects PEAVAUTO-CM-UC3M, and RTI2018-096036-B-C22, and by the Region of Madrid's Excellence Program (EPUC3M17).




REFERENCES

- [1] P. Angelov, X. Gu, D. Kangin, and J. Principe, "Empirical data analysis," *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, pp. 52–59, 2017.
- [2] R. Nawaratne, D. Alahakoon, D. De Silva, and X. Yu, "Spatiotemporal anomaly detection using deep learning for real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 393–402, 2020, doi: 10.1109/TII.2019.2938527.
- [3] A. M. Ali, P. Angelov, and X. Gu, "Detecting anomalous behaviour using heterogeneous data," *Advances in Intelligent Systems and Computing*, vol. 513, pp. 253–273, 2017, doi: 10.1007/978-3-319-46562-3_17.
- [4] P. Krammer, O. Habala, J. Mojžiš, L. Hluchý, and M. Jurkovič, "Anomaly detection method for online discussion," *Procedia Computer Science*, vol. 155, pp. 311–318, 2019, doi: 10.1016/j.procs.2019.08.045.
- [5] L. Rettig, M. Khayati, P. Cudre-Mauroux, and M. Piorkowski, "Online anomaly detection over big data streams," *Proceedings-2015 IEEE International Conference on Big Data, IEEE Big Data 2015*, pp. 1113–1122, 2015, doi: 10.1109/BigData.2015.7363865.
- [6] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, "Real-time fault detection using recursive density estimation," *Journal of Control, Automation and Electrical Systems*, vol. 25, no. 4, pp. 428–437, 2014, doi: 10.1007/s40313-014-0128-4.
- [7] B. Bose, J. Dutta, S. Ghosh, P. Pramanick, and S. Roy, "DRSense: Detection of driving patterns and road anomalies," *Proceedings-2018 3rd International Conference On Internet of Things: Smart Innovation and Usages, IoT-SIU 2018*, 2018, doi: 10.1109/IoT-SIU.2018.8519861.
- [8] M. Dharmadhikari and V. L. Kolhe, "Anomaly extraction using association rule with the heterogeneous detectors," *2014 International Conference on Information Communication and Embedded Systems, ICICES 2014*, 2015, doi: 10.1109/ICICES.2014.7033908.
- [9] A. M. Ali and P. Angelov, "Anomalous behaviour detection based on heterogeneous data and data fusion," *Soft Computing*, vol. 22, no. 10, pp. 3187–3201, 2018, doi: 10.1007/s00500-017-2989-5.
- [10] R. Foorhuis, "On the nature and types of anomalies: a review of deviations in data," *International Journal of Data Science and Analytics*, vol. 12, no. 4, pp. 297–331, 2021, doi: 10.1007/s41060-021-00265-1.
- [11] V. M. Tellis and D. J. D'Souza, "Detecting anomalies in data stream using efficient techniques: A review," *2018 International Conference on Control, Power, Communication and Computing Technologies, ICCPCCT 2018*, pp. 296–298, 2018, doi: 10.1109/ICCPCCT.2018.8574310.
- [12] P. Angelov, "Outside the box: an alternative data analytics framework," *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 8, no. 2, pp. 29–35, Apr. 2014, doi: 10.14313/JAMRIS_2-2014/16.
- [13] B. Kristof and S. Rinderle-ma, "Systematic literature review on anomaly detection in business process runtime behavior," 2017.
- [14] B. Delgado, K. Tabhou, and E. J. Delp, "Automatic detection of abnormal human events on train platforms," *National Aerospace and Electronics Conference, Proceedings of the IEEE*, vol. 2015-Febru, pp. 169–173, 2015, doi: 10.1109/NAECON.2014.7045797.
- [15] B. Krausz and C. Bauchhage, "Automatic detection of dangerous motion behavior in human crowds," *2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2011*, pp. 224–229, 2011, doi: 10.1109/AVSS.2011.6027326.
- [16] X. Wang, A. Mohd Ali, and P. Angelov, "Gender and age classification of human faces for automatic detection of anomalous human behaviour," *2017 3rd IEEE International Conference on Cybernetics, CYBCONF 2017-Proceedings*, 2017, doi: 10.1109/CYBConf.2017.7985780.
- [17] Y. Zhou and J. Li, "Research of network traffic anomaly detection model based on multilevel autoregression," *Proceedings of IEEE 7th International Conference on Computer Science and Network Technology, ICCSNT 2019*, pp. 380–384, 2019, doi: 10.1109/ICCSNT47585.2019.8962517.
- [18] P. Sadeghi-Tehran and P. Angelov, "A real-time approach for novelty detection and trajectories analysis for anomaly recognition in video surveillance systems," *2012 IEEE Conference on Evolving and Adaptive Intelligent Systems, EALS 2012-Proceedings*, pp. 108–113, 2012, doi: 10.1109/EALS.2012.6232814.
- [19] P. Angelov, "Anomaly detection based on eccentricity analysis," *IEEE SSCI 2014-2014 IEEE Symposium Series on Computational Intelligence-EALS 2014: 2014 IEEE Symposium on Evolving and Autonomous Learning Systems, Proceedings*, pp. 1–8, 2014, doi: 10.1109/EALS.2014.7009497.
- [20] W. Li, G. Wu, and Q. Du, "Transferred deep learning for anomaly detection in hyperspectral imagery," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 597–601, 2017, doi: 10.1109/LGRS.2017.2657818.
- [21] Z. Li and Y. Zhang, "Hyperspectral anomaly detection based on improved RX with CNN framework," *International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 2244–2247, 2019, doi: 10.1109/IGARSS.2019.8898327.
- [22] X. Gu and P. Angelov, "Autonomous anomaly detection," *IEEE Conference on Evolving and Adaptive Intelligent Systems*, vol. 2017-May, 2017, doi: 10.1109/EALS.2017.7954831.
- [23] X. Gu, P. P. Angelov, and J. C. Principe, "A method for autonomous data partitioning," *Information Sciences*, vol. 460–461, pp. 65–82, 2018, doi: 10.1016/j.ins.2018.05.030.
- [24] M. Van Onsem *et al.*, "Hierarchical pattern matching for anomaly detection in time series," *Computer Communications*, vol. 193, pp. 75–81, 2022, doi: 10.1016/j.comcom.2022.06.027.
- [25] L. Qi, Y. Yang, X. Zhou, W. Rafique, and J. Ma, "Fast anomaly identification based on multiaspect data streams for intelligent intrusion detection toward secure industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6503–6511, 2022, doi: 10.1109/TII.2021.3139363.
- [26] L. Zhang, J. Zhao, and W. Li, "Online and unsupervised anomaly detection for streaming data using an array of sliding windows and PDDS," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 2284–2289, 2021, doi: 10.1109/TCYB.2019.2935066.
- [27] Z. Chen *et al.*, "Correlated anomaly detection from large streaming data," *Proceedings-2018 IEEE International Conference on Big Data, Big Data 2018*, pp. 982–992, 2019, doi: 10.1109/BigData.2018.8622004.
- [28] H. Wang and B. Raj, "A survey: Time travel in deep learning space: An introduction to deep learning models and how deep learning models evolved from the initial ideas," 2015.
- [29] B. S. J. Costa, P. P. Angelov, and L. A. Guedes, "A new unsupervised approach to fault detection and identification," *Proceedings of the International Joint Conference on Neural Networks*, pp. 1557–1564, 2014, doi: 10.1109/IJCNN.2014.6889973.
- [30] J. A. Iglesias, P. Angelov, A. Ledezma, and A. Sanchis, "Creating evolving user behavior profiles automatically," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 854–867, 2012, doi: 10.1109/TKDE.2011.17.
- [31] L. M. D. Da Silva *et al.*, "Hardware architecture proposal for TEDA algorithm to data streaming anomaly detection," *IEEE Access*, vol. 9, pp. 103141–103152, 2021, doi: 10.1109/ACCESS.2021.3098004.
- [32] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," *Proceedings-IEEE International Conference on Data Mining, ICDM*, pp. 413–422, 2008, doi: 10.1109/ICDM.2008.17.




- [33] L. Arnold, S. Rebecchi, S. Chevallier, and H. Paugam-Moisy, "An introduction to deep learning," *ESANN 2011-19th European Symposium on Artificial Neural Networks*, pp. 477–488, 2011, doi: 10.1201/9780429096280-14.
- [34] G. Özbülak, Y. Aytar, and H. K. Ekenel, "How transferable are CNN-based features for age and gender classification?," *Lecture Notes in Informatics (LNI), Proceedings-Series of the Gesellschaft für Informatik (GI)*, vol. P-260, 2016, doi: 10.1109/BIOSIG.2016.7736925.
- [35] P. Angelov and E. Soares, "Towards explainable deep neural networks (xDNN)," *Neural Networks*, vol. 130, pp. 185–194, 2020, doi: 10.1016/j.neunet.2020.07.010.
- [36] P. Angelov and E. Soares, "Towards deep machine reasoning: A prototype-based deep neural network with decision tree inference," *Conference Proceedings-IEEE International Conference on Systems, Man and Cybernetics*, vol. 2020-Octob, pp. 2092–2099, 2020, doi: 10.1109/SMC42975.2020.9282812.
- [37] M. Fisher, V. Mascardi, K. Y. Rozier, B.-H. Schlingloff, M. Winikoff, and N. Yorke-Smith, "Towards a framework for certification of reliable autonomous systems," *Autonomous Agents and Multi-Agent Systems*, vol. 35, no. 1, p. 8, Apr. 2021, doi: 10.1007/s10458-020-09487-2.

BIOGRAPHIES OF AUTHORS






Muhammad Yunus Iqbal Basheer    is a master student in computer science at Universiti Teknologi MARA (UiTM), Shah Alam. He received his diploma in computer science from UiTM, Perlis. He specializes in artificial intelligence, with an emphasis on intelligence programming and data science. Previously, during bachelor's degree in UiTM, Shah Alam, he focuses on analytics on student university intake (2019). Currently, his research interest on anomaly detection. He can be contacted at email: muhammadyunus185@gmail.com.






Azliza Mohd Ali    received both first and master's degree from Universiti Utara Malaysia (UUM) in Bachelor of Information Technology (2001) and Master of Science (Intelligent Knowledge-Based System (2003). She joined Universiti Teknologi MARA (UiTM) as a lecturer in 2004 and received a PhD in Computer Science from Lancaster University, UK. She dedicates herself to university teaching and conducting research. Currently her research interest on anomaly detection, data mining, machine learning and knowledge-based systems. She can be contacted at email: azliza@tmsk.uitm.edu.my.






Rozianawaty Osman    received PhD in computer science from University of Reading, United Kingdom. She received Master of Science (Information Technology) from Universiti Utara Malaysia (2005). She has been a senior lecturer in Universiti Teknologi MARA (UiTM) since 2009. Currently, her research interest on human computer interaction, user interface and user experience design, and digital health. She can be contacted at email: rozianawaty@uitm.edu.my.






Nurzeatul Hamimah Abdul Hamid    is a senior lecturer of Information System in Universiti Teknologi Mara. She received a master's degree in Intelligent Systems at the University of Sussex, UK in 2005. She teaches courses related to fundamentals of artificial intelligence, artificial intelligence programming paradigm and intelligent agent. Her primary research interests involve software agents, normative multi-agent systems, trust and reputation systems. She can be contacted at email: nurzeatul@tmsk.uitm.edu.my.






Sharifalillah Nordin    received her Bachelor of Information Technology in 2001 from Universiti Utara Malaysia (UUM), Master of Science (Internet Computing) in 2003 from University of Surrey, UK, and PhD in Bioinformatics from Universiti Malaya (UM) in 2010. She is currently a Senior Lecturer at the Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA (UiTM), Shah Alam. She has been an academician in UiTM since 2009. She dedicates herself to university teaching and conducting research. Her research fields include biodiversity informatics, knowledge engineering, and artificial intelligence. She can be contacted at email: sharifa@tmsk.uitm.edu.my.



Muhammad Azizi Mohd Ariffin    received BS degree in data communication and networking from Universiti Teknologi MARA (UiTM) in 2014. He further received a master's degree in cyber security from Lancaster University, UK (2016). During his studies, he was involved in many cyber security competitions. He is currently involved in CherryTree CSR program since 2017 in TIME dotcom Berhad. He has dedicated himself as an academician in UiTM since 2019. He can be contacted at email: mazizi@tmsk.uitm.edu.my.



Jose Antonio Iglesias Martínez    received the BS degree in computer science from Valladolid University in 2002 and a PhD degree in computer science from Carlos III University of Madrid (UC3M) in 2010. He is a member of the CAOS research group at Carlos III University of Madrid, Spain. He has published more than 15 journal and conference papers. He also takes part in several national research projects and is committee member of several international conferences including a member of the Fuzzy Systems Technical Committee (IEEE/CIS). His research interests include agent modeling, plan recognition, sequence learning, machine learning, and evolving fuzzy systems. He can be contacted at email: jiglesia@inf.uc3m.es.