Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.039984
Article

Intelligent Intrusion Detection System for the Internet of Medical Things Based on Data-Driven Techniques

Okba Taouali[1,*], Sawcen Bacha[2], Khaoula Ben Abdellafou[1], Ahamed Aljuhani[1], Kamel Zidi[3], Rehab Alanazi[1] and Mohamed Faouzi Harkat[4]

[1]Faculty of Computers and Information Technology, University of Tabuk, Tabuk, 71491, Saudi Arabia
[2]National Engineering School of Monastir, University of Monastir, Monastir, 5000, Tunisia
[3]Applied College, University of Tabuk, Tabuk, 71491, Saudi Arabia
[4]Department of Electronics, Faculty of Engineering Annaba, Annaba, 23000, Algeria
*Corresponding Author: Okba Taouali. Email: otawali@ut.edu.sa
Received: 27 February 2023; Accepted: 27 April 2023; Published: 28 July 2023

Abstract: Introducing IoT devices to healthcare fields has made it possible to remotely monitor patients' information and provide a proper diagnosis as needed, resulting in the Internet of Medical Things (IoMT). However, obtaining good security features that ensure the integrity and confidentiality of patient's information is a significant challenge. However, due to the computational resources being limited, an edge device may struggle to handle heavy detection tasks such as complex machine learning algorithms. Therefore, designing and developing a lightweight detection mechanism is crucial. To address the aforementioned challenges, a new lightweight IDS approach is developed to effectively combat a diverse range of cyberattacks in IoMT networks. The proposed anomaly-based IDS is divided into three steps: pre-processing, feature selection, and decision. In the pre-processing phase, data cleaning and normalization are performed. In the feature selection step, the proposed approach uses two data-driven kernel techniques: kernel principal component analysis and kernel partial least square techniques to reduce the dimension of extracted features and to ameliorate the detection results. Therefore, in decision step, in order to classify whether the traffic flow is normal or malicious the kernel extreme learning machine is used. To check the efficiency of the developed detection scheme, a modern IoMT dataset named WUSTL-EHMS-2020 is considered to evaluate and discuss the achieved results. The proposed method achieved 99.9% accuracy, 99.8% specificity, 100% Sensitivity, 99.9 F-score.

Keywords: Machine learning; data-driven technique; KPCA; KPLS; intrusion detection; IoT; Internet of Medical Things (IoMT)

1 Introduction

The Internet of Things (IoT) technology has transformed the digital world and is considered one of the most significant revolutions in the communication and information technology area. The IoT has been rapidly developed and deployed in a variety of important domains like transportation, agriculture, energy, and healthcare owing to the significant benefits that such technology offers [1–4]. Precisely, IoT technology has tremendously impacted the healthcare sector, resulting in the Internet of Medical Things (IoMT) [5]. The IoMT has revolutionized the healthcare and public health domain as it provides cost-effective, efficient, and smart healthcare environments. The IoMT is a collection of smart connected devices, hardware infrastructure, and medical applications connected to healthcare systems using different wireless technologies (e.g., Bluetooth, Sigfox, LoRa, and 5G/LTE) [6,7]. The global market of IoT is expected to reach $254.2 billion in 2026 [8].

The IoMT applications require superior computing capabilities to provide sustainable, resilient, and secure healthcare systems; however, cloud computing cannot efficiently fulfill these requirements due to several limitations such as latency, Internet connectivity, and lack of mobility support [9]. To address these issues, edge computing is proposed to overcome several limitations in the cloud by offloading the computational tasks to the edge [10]. However, edge computing does not replace cloud technology; rather, it serves as an extension of the cloud. Edge computing can significantly improve the quality of medical services in IoMT applications which requires low latency, mobility support, and location awareness [11,12]. Because the IoMT includes heterogeneous and interconnected devices, a massive amount of data is generated, transmitted, and stored in the edge instead of a centralized cloud as it improves medical services with low latency time, flexible access, and real-time processing [13].

Although IoMT significantly improves the medical services to patients, cyberattacks stay to pose a significant risk to healthcare providers and cause enormous damage [14]. An Attack such as Distributed Denial of Service (DDoS) performs a major threat to the healthcare system as it disrupts connected devices and makes medical services unavailable to legitimate clients [15,16]. Another type of cyberattack is man-in-the-middle, in which an adversary intercepts communication between IoMT devices to manipulate and steal sensitive data. Such an attack has a significant impact on the integrity and confidentiality of patient data. Another type of cyberattack that endangers the healthcare system is ransomware. In this case, an adversary encrypts IoMT systems, making it difficult for healthcare providers and stakeholders to access medical records and provide healthcare services for patients [17].

To ensure the security and privacy-preserving of information sharing in the IoMT environments, the intrusion detection system (IDS) represent a perfect security tool to overcome a variety of cyber-attacks. The IDS can quickly detect anomalies and alert the system to prevent further damage. The critical part of IDS is the detection algorithm and its ability to detect different types of cyberattacks with good accuracy and minimum false alarm rates. The IDS could be integrated with edge computing to provide effective and efficient attack detection close to the data source. Additionally, the IDS can benefit from computational resources at the edge, allowing it to use complex detection algorithms and more storage capacity to store and analyze log data [18]. More importantly, edge computing provides adequate network bandwidth and low latency, both of which are critical for real-time detection.

Although edge computing provides excellent computing capabilities, it lacks the required resources to complete intensive tasks such as heavy-weight machine learning models [18]. Additionally, the IoMT includes wearable devices and sensors that are wirelessly connected and generate heterogeneous and homogeneous data [19], the massive amount of generated data must be processed and analyzed in real time without significantly consuming the computational power and storage capacity of such

a detection mechanism [20]. Therefore, developing a lightweight IDS is crucial to effectively and efficiently overcome different types of cyberattacks in IoMT networks.

To meet the abovementioned challenges, this paper suggested a novel and lightweight IDS approach to efficiently overcome cyberattacks in IoMT networks. The contribution of the paper are:

- As the capacity of any classifier mainly count on the features provided as input, two data-driven kernel techniques entitled Kernel Principal Component Analysis (KPCA) and Kernel Partial Least Square (KPLS) are applied to choose important features from the feature vector
- To improve the classifier performance in detecting cyberattacks. Because edge devices have limited computational resources, fast training speed [21,22], and an efficient learning model are required. Therefore, the kernel extreme learning machine (KELM) is suggested as classifier to divine whether the traffic flow is benign or malicious.
- As the dataset plays a vital role to test the robustness and effectiveness of the detection model, our developed approach uses a modern IoMT dataset named WUSTL-EHMS-2020. The developed approach outperforms the other suggested methods in terms of accuracy, specificity, and sensitivity rate, as well as training speed and prediction time. The results show that our suggested IDS model has a high potential for use in the context of edge computing in IoMT networks.

This article is structured as follows. The related works and discusses the limitations are presented in Section 2. The proposed methodology is presented and illustrated in Section 3. Section 4 discusses and evaluates the proposed work using different performance metrics. Finally, Section 5 concludes the article and proposes future work.

2 Related Works

Several studies on intrusion detection have been conducted to overcome cyberattacks in IoMT networks. An et al. [21] suggested a lightweight intrusion detection technique in fog computing and mobile edge computing. The developed model used sample selected extreme learning machine for attack detection and classification. The proposed framework deployed the detection classifier on the fog node while the training dataset was stored in the cloud server to minimize the training time and improve the detection performance. According to the achievement results which prove the efficiency of the proposed model against cyberattacks. Alatawi et al. [22] developed an anomaly detection approach in fog-to-things communications to overcome cyberattacks in smart environments. To ameliorate anomaly detection performance, the proposed work used two feature selection methods: minimum redundancy maximum relevance (MRMR), and principal component analysis (PCA). The proposed model used ensemble learning techniques for attack classification. The performance results demonstrated that the proposed model was effective at detecting cyberattacks.

Grammatikis et al. [23] developed an anomaly detection model to identify cyberattacks in IoMT networks. The suggested algorithm used an active learning to select the main features and a Random Forest in classification phase. The performance results demonstrated that the proposed approach was effective at detecting cyberattack.

Bacha et al. [24] suggest a novel intrusion detection model in IoT networks. The suggested work employed a kernel extreme learning machines for binary and multiclass classification. Two datasets were used in learning and testing steps to validate the efficiency of the proposed anomaly detection model. When compared to other existing approaches, the proposed model achieved a higher accuracy rate. Saheed et al. [25] suggested an intrusion detection model overcome cyberattacks in IoT

networks. The proposed model used a deep recurrent neural network and supervised machine learning approaches such as random forest, K-nearest neighbors, decision trees, and ridge classifiers. The Authors utilized particle swarm optimization in the feature selection phase to improve the detection results of the developed model. The evaluation of obtained results proves the effectiveness of the developed IDS against IoT/IoMT cyberattacks.

Ketu et al. [26] proposed a new classification algorithm for air quality detection. The proposed approach used the scalable kernel-based SVM (Support Vector Machine) classification algorithm which is capable of dealing with the multi-class data imbalance issue. The performance evaluation shows the sufficiency of the suggested technique.

Alrashdi et al. [27] developed an anomaly detection model to identify cyberattacks in IoT networks. The developed detection model used machine learning algorithms such as random forest and extra tree classifiers to be deployed in the fog layer. When anomalies are detected, the suggested framework will alert the cloud server for further security analysis. The obtained results proved the efficiency of the proposed detection model. Hady et al. [28] developed an intrusion detection system to protect healthcare networks from cyberattacks. The authors created a healthcare testbed to collect and analyze combined network traffic and biometrics data. The proposed security system used different machine learning models to detect cyberattacks in such environments. The results showed that combining flow metrics and biometrics information improved accuracy by 25%. Kumar et al. [29] proposed a detection approach for protecting IoMT systems from various types of cyberattacks in the context of fog-cloud architecture. The detection model employed ensemble learning techniques like decision trees, naive Bayes, and random forest as the first stage, with XGBoost serving for classification in the second stage. The obtained results showed that the developed technique is capable to detect cyberattacks.

Rahman et al. [30] developed an anomaly detection system to detect cyberattacks in IoT networks. The proposed approach used a Federated Learning based scheme for ioT intrusion detection that retains data privacy by performing local training and inference of detection models. The obtained performance shows the efficiency of the suggested IDS against IoT cyberattacks.

The literature has revealed several limitations. For example, the use of an inappropriate dataset for healthcare systems, or the use of an out-of-date dataset that is incompatible with designing and implementing IDS for cyberattack detection in IoMT networks. Although some approaches yielded promising performance results, the used detection algorithms suffer from computational complexity, making the deployment at edge devices critical.

To solve these challenges, a lightweight and cost-effective IDS is designed to protect IoMT systems from cyberattacks. The proposed work selects important features from the reduced feature vector using data-driven techniques such as KPCA and KPLS. Such techniques have demonstrated their effectiveness in sensor fault detection [31–33], and have a high potential for use and integration as an integral part of the detection operations within the IDS. In the classification phase, the KELM technique is used to effectively identify malicious activities from benign traffic. The proposed approach achieved a higher accuracy, sensitivity, specificity, and f1-score compared to other proposed techniques. Additionally, the learning and testing time of the proposed method was low, making it more suitable for use in an online detection system deployed in the context of edge computing

3 Proposed Methodology

In this section, the suggested intelligent IDS is detailed in Fig. 1.

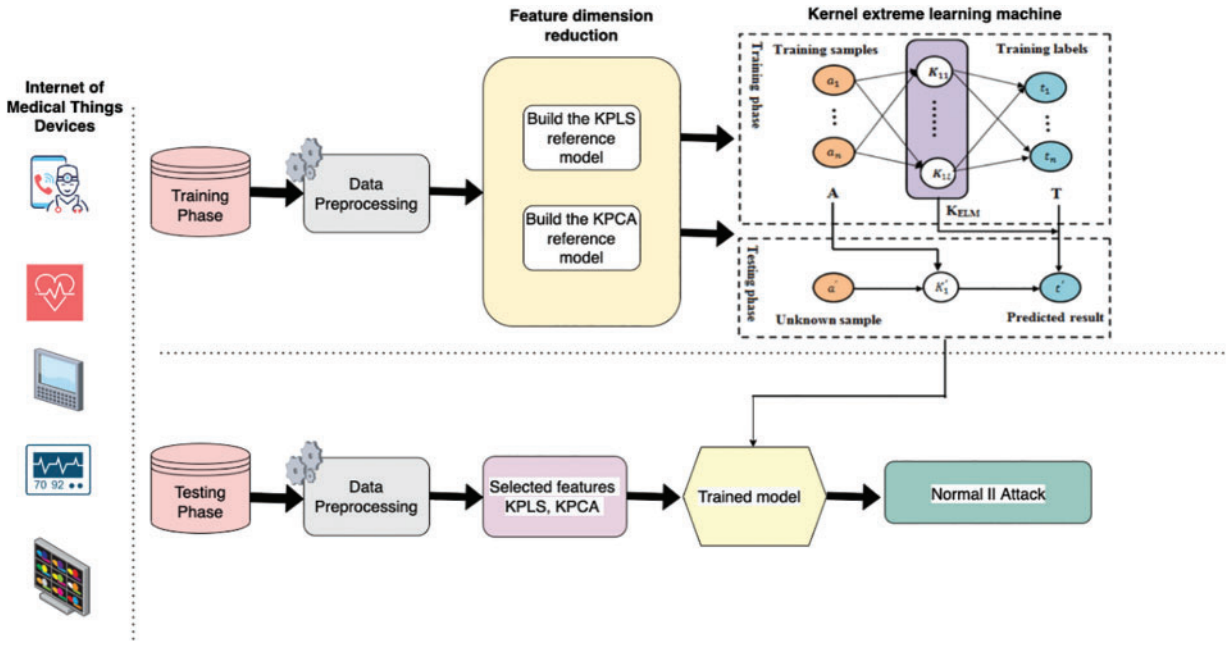

Figure 1: Flowchart of intelligent intrusion detection system for internet of medical things

To begin, the symbols and notations used in the paper are summarized in Table 1. Then the two data-driven kernel techniques, KPCA and KPLS, are detailed which are used in the feature extraction phase to reduce the dimensionality of data features. Also the KELM classifier model that is used in our intelligent detection system for cyberattack classification is discussed.

Table 1: Symbols and notations

Symbols and notations	Name
N	Number of instances
A	Data input matrix
C_{ψ}	Covariance matrix
F	Feature space
μ_j	Eigenvector
λ_j	Eigenvalue
$\alpha_{i,j}$	Parameters
K	Kernel matrix
K_t	Kernel matrix of the test samples.
$\hat{Y}_{Train}$	Prediction output of the learning set.
$\hat{Y}_{Test}$	Prediction outputs of the validation set.
Λ	Diagonal matrix
σ	Parameter of RBF kernel
P	Degree of the polynomial kernel
T	Score vectors
Z	Training observations

(Continued)

Table 1 (continued)

Symbols and notations	Name
m	Output nodes number
w_j	Weight vector
L	Number of hidden neurons in the hidden layer

3.1 Features Extraction Phase

3.1.1 Kernel Principal Component Analysis

In the literature, there are different dimension reduction techniques such as the locality preserving projections (LPP) [34] and principal component analysis (PCA) [35]. The PCA is a widely used technique for analyzing large datasets with a high number of features. It is a statistical method for reducing the dimensionality of the input information vector, to increase the interpretability while preserving the maximum information. Despite the proven advantages of the PCA technique, its foundation lies in the linearity of the process. To overcome this problem, the Kernel PCA approach is used in the literature [32]. KPCA technique is illustrated in two phases: the first one is to transform the input observations onto the feature space with large dimensional, then the PCA is carried out in the considered feature space.

In the suggested article, the Kernel PCA method is used to reduce the dimensionality of features and is given by:

$A = [z_1, z_2, \ldots, z_n]^T$ is the data input matrix. The transformation into high dimensional feature space is given by: as:

$$\psi : \mathrm{E} \subset \mathbb{R}^{\mathrm{d}} \rightarrow \mathrm{F} \subset \mathbb{R}^{\mathrm{r}};\ \mathrm{z} \rightarrow \psi\,(\mathrm{z}) \tag{1}$$

$$\mathrm{C}_{\psi} = \frac{1}{\mathrm{N}} \sum\nolimits_{\mathrm{i}=1}^{\mathrm{N}} \psi \psi_{\mathrm{i}}{}^{\mathrm{T}} = \frac{1}{\mathrm{n}-1} \mathrm{A}^{\mathrm{T}}\mathrm{A} \tag{2}$$

where $z \in E \subset \mathbb{R}^d$ is a observation vector and $\boldsymbol{C}_{\psi}$ is the covariance matrix in F with:

$$\psi_i = \psi\,(a_i) \in \mathbb{R}^N \text{ and } A = [\psi_1, \psi_2, \ldots, \psi_N]^T \in \mathbb{R}^{N*h} \tag{3}$$

The KPCA reference model is determined by solving the following equation.

$$\lambda_j \mu_j = C_{\psi} \mu_j \quad with \qquad j = 1, \ldots, h \tag{4}$$

where μ_j is the j^{th} eigenvector of C_{ψ} corresponding to eigenvalue λ_j.

For $\lambda_j \neq 0$, there exist parameters $\alpha_{i,j}\ \ i = 1, \ldots, N$ such all eigenvectors μ_j can be designed a linear combination of $[\psi(z_1), \psi(z_2), \ldots, \psi(z_N)]$ and can be expressed by:

$$\mu_j = \sum\nolimits_{i=1}^{N} \alpha_{i,j}\, \psi\,(z_i) \tag{5}$$

Eq. (4) can be rewritten as:

$$\begin{cases} \mu_j \langle \psi(z_i), \mu_j \rangle = \langle \psi(z_i), C_{\psi} \mu_j \rangle \\ j = 1, \ldots, h \end{cases} \tag{6}$$

Combining Eqs. (2) and (4) in Eq. (5) to obtain:

$$\lambda_j \sum_{k=1}^{N} \alpha_k^j \langle \psi(z_i), \psi(z_k) \rangle = \langle \psi(z_i), \frac{1}{N} \sum_{l=1}^{N} \psi(z_l) \psi(z_l)^T \sum_{k=1}^{N} \alpha_k^j \psi(z_k) \rangle \tag{7}$$

$$\lambda_j \sum_{k=1}^{N} \alpha_k^j \langle \psi(z_i), \psi(z_k) \rangle = \psi \langle (z_i), \frac{1}{N} \sum_{l=1}^{N} \sum_{k=1}^{N} \alpha_k^j \psi(z_l) \langle \psi(z_l) \psi(z_k) \rangle \rangle \tag{8}$$

$$\lambda_j \sum_{k=1}^{N} \alpha_k^j \langle \psi(z_i), \psi(z_k) \rangle = \frac{1}{N} \sum_{l=1}^{N} \sum_{k=1}^{N} \alpha_k^j \langle \psi(z_l) \psi(z_l) \langle \psi(z_l) \psi(z_l) \rangle \rangle \tag{9}$$

Using the kernel trick. The inner product given in Eq. (2) may be computed using a kernel function $k(.,.)$ and is written:

$$\langle \psi(z_i), \psi(z_j) \rangle_H = k(z_i, z_j) \tag{10}$$

Considering a Gram matrix $K \in \mathbb{R}^{N \times N}$ associated with a kernel function k:

$$K = \begin{bmatrix} k(z_1, z_1) & \cdots & k(z_1, z) \\ \vdots & \ddots & \vdots \\ k(z_N, z_1) & \cdots & k(z_N, z_N) \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{11}$$

Using the kernel matrix may reduce the problem of the eigenvalue decomposition of C_ψ. Hence, the eigen decomposition of the transformed kernel matrix K is identical to applying PCA in a feature space $\mathbb{R}^H$, so that:

$$N \Lambda \Upsilon = K \Upsilon \tag{12}$$

where:

$\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_j, \ldots \lambda_N)$ represents the diagonal matrix with eigenvalues λ_j sorted in in downward order and

$\Upsilon = [\alpha_1, \ldots, \alpha_j, \ldots \alpha_N]$ represents the matrix of their corresponding eigenvectors.

Since the principal components vectors are orthonormal, it is necessary to guarantee the normality of μ_j in Eq. (5):

$$\langle \mu_j, \mu_j \rangle_H = 1 \tag{13}$$

$$\begin{aligned} \langle \mu_j, \mu_j \rangle_H &= \sum_{i,k}^{N} \alpha_{i,j}, \alpha_{k,j} \langle \psi(z_i), \psi(z_k^T) \rangle_H \\ : \quad &= \sum_{i,k}^{N} \alpha_{i,j}, \alpha_{k,j} K_{i,k} = \langle \alpha_j, K \alpha_j \rangle_H \\ &= \langle \alpha_j, K \alpha_j \rangle_H = \lambda_j \langle \alpha_j, \alpha_j \rangle_H \end{aligned} \tag{14}$$

where N represents the number of the first eigenvalues with values different of zero. With

$K_{i,k} = k(a_1, a_N)$. The corresponding eigenvectors α_j is written as:

$$\langle \alpha_j, \alpha_j \rangle_H = ||\alpha_j||^2 = \frac{1}{\lambda_j} \tag{15}$$

Many kernel functions have been used in literature, see Table 2 below.

Table 2: Kernels function

Name	Formula and parameters
RBF-kernel	$K_\sigma(x,y) = exp\left(-\frac{1}{2}\frac{\|\|x-y\|\|^2}{\sigma^2}\right)$ σ: RBF kernel parameter
Linear-kernel	$K_w(x,y) = <x,y>$
Polynomial-kernel	$k_P(x,y) = (1+<x,y>)^P$ P: Polynomial kernel degree

The radial basis function (RBF) is given by:

$$k(x,y) = \exp\left(-\frac{\|\|x-y\|\|^2}{2\sigma^2}\right) \tag{16}$$

where $\sigma \in \mathbb{R}^+$, the centered gram matrix $\boldsymbol{K}$ is computed as:

$$\overline{K} = FKF \quad \text{With} \quad F = (I_n - E) \tag{17}$$

where $1_n = \frac{1}{n}[1,\ldots,1]^T \in \mathbb{R}^n$ and E is an $n \times n$ matrix with elements $\frac{1}{n}$.

3.1.2 Kernel Partial Least Square Method (KPLS)

The are many approaches, such as principal component analysis (PCA) [36] and partial least squares (PLS) [37] have been used to analyze gene expression data. However, these traditional techniques still have some drawbacks. The researchers proposed many versions of the methods to overcome these disadvantages. One of these methods is the Kernel partial least squares regression (KPLS) which is an approach widely used to generate a predictive models. The calculation time, chosen for the KPLS method, may be higher than that chosen for the PLS method during the learning phase since the number of selected latent variables for KPLS may be higher than that of the linear PLS. In the proposed work, a new classification method is proposed. This proposed method is an extension of the basic PLS.

Using the input data $= \{x_1,\ldots,x_N\} \in R^{N\times M}$, the PLS technique extracts the latent variables LVs where it contains N observations and their corresponding output observations $Y = \{y_1,\ldots,y_N\} \in R^N$ where y_i are the categories, classes or labels of the corresponding x_i to construct a linear equation.

In the next step, the input and output observations will be transformed into space that is generated by some latent variables [30] and decomposed as follows:

$$\begin{cases} X = TL^T + G \\ Y = UR^T + H \end{cases} \tag{18}$$

where $T = [t_1,\ldots,t_l]$ and $L = [l_1,\ldots,l_l]$ are the score vectors and $U = [u_1,\ldots,u_l]$, $R = [r_1,\ldots,r_l]$ are the charging for X and Y, jointly.

The PLS residues of X and Y are defined by the two matrices G and H respectively. PLS is a linear approach. If a linear model is not adequate, a transformation function [38] can be applied to map data into a higher-dimensional space (called feature space), where linear model is applied.

Mathematically, the transformation of observation in the feature space is given:

$$\psi : E \subset \mathbb{R}^N \to F \subset \mathbb{R}^H \; ; \; x \to \psi(x) \tag{19}$$

The kernel function is computed as:

$$\langle \psi(x_i), \psi(x_j) \rangle_H = k(x_i, x_j) \;\; \forall \, x_i, \; x_j \in \mathbb{R}^N \tag{20}$$

The Gram matrix $K \in \mathbb{R}^{N \times N}$ corresponding to a kernel function k is given by:

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & \ddots & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix} \in \mathbb{R}^{N \times N} \tag{21}$$

where $\sigma \in \mathbb{R}^+$.

Algorithm 1: Kernel partial least square method

Input: X: input data matrix of size N×M
Y: output data matrix of size N × 1
Output: input score matrix T
output score matrix V

1. Compute and center the Gram matrix
2. $i = 1, \; K1 = K \, , \; Y1 = Y$
3. Initialize u_i to any column vector of Y_i randomly
4. $t_i = K_i^T u_i t_i = t_i / ||t_i||$
5. $c_i = Y_i^T t_i$
6. $v_i = Y_i c_i, c_i = c_i / ||c_i||$
7. If t_i converges, go to Eq. (7); else go to Eq. (3)
8. Exhausted K and Y
9. Extract more latent variables by repeating steps three through six
10. Collect the T and U matrices.

The rank-one reduction of K and Y [39] gave us the deflation step where the K and Y matrices are deflated based on a new T-score vector as follows:

$$K = K - tt^T - Ktt^T + tt^T Ktt^T \tag{22}$$

$$Y = Y - tt^T Y \tag{23}$$

Then, the model generated by the KPLS technique is defined as:

$$\begin{cases} \widehat{Y}_{Train} = KU \; \left(T^T \; KU\right)^{-1} T^{\,T} Y \\ \widehat{\mathrm{Y}}_{\mathrm{Test}} = \mathrm{K_t} U \; \left(\mathrm{T^T KU}\right)^{-1} \mathrm{T^T} \; Y \end{cases} \tag{24}$$

- K_t presents the kernel matrix of the test samples.
- $\widehat{Y}_{Train}$ denotes the prediction output of the learning set.
- $\widehat{Y}_{Test}$ denotes the prediction outputs of the validation set.

3.2 Classification Phase

3.2.1 Extreme Learning Machine

An extreme learning machine (ELM) scheme represents a feedforward neural network with a single layer (SLFN). The neural structure is crucial to the transformed representation of data and the final performance and zero-shot learning [40]. ELM was first developed for learning SLFNs and was subsequently extended for learning the generalized SLFNs. In ELM, the structure contains an input layer, a hidden layer, and an output layer. Each neuron is attached by a weight (w) and is given in Fig. 2.

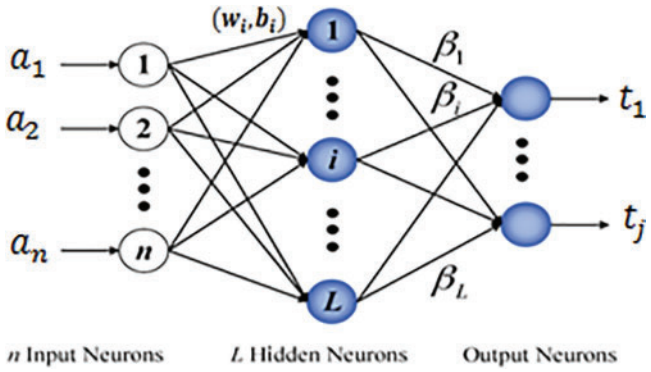

Figure 2: The structure of extreme learning machine

where: b is the bias, the activation function (h), which computes the model output (t).

The network is described based on the triplet (w, b, h) as follows:

Given learning observations $Z = \{A_i, T_i\}_{i=1,\ldots N}$ where a_i, $t_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in R^m\}_{i=1,\ldots,N}$ where N is the number of instances and m is the output nodes number.

The output function of ELM is given by:

$$f(a_i) = t_i = \sum_{j=1}^{L} \beta_j \, h\left(w_j . \, a_i + b_j\right) \quad i = 1, \ldots, N \tag{25}$$

where $w_j = \left[w_{j1}, w_{j2}, \ldots, w_{jn}\right]$ is the weight vector, $\beta_j = \left[\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}\right]^T$ is the weight vector relating the output neurons to the hidden neurons, L is the number of hidden neurons.

The matrix format of Eq. (25) is written as

$$H\beta = T\mathrm{W} \tag{26}$$

where H is given by:

$$\boldsymbol{H} = \begin{bmatrix} h(w_1.a_1 + b_1) & \cdots & h(w_k.a_1 + b_L) \\ \vdots & \ddots & \vdots \\ h(w_1.a_N + b_1) & \cdots & h(w_k.a_N + b_L) \end{bmatrix}_{N \times L} \tag{27}$$

$$\text{With } \beta = \begin{pmatrix} \beta_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \beta_L^T \end{pmatrix}_{L\times m} \text{ and } T = \begin{pmatrix} t_1^T \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ t_L^T \end{pmatrix}_{N\times m}$$

Using the least square principle, the solution of Eq. (26), can be mathematically modeled as

$$\beta = H^{+}T$$

With H^{+} is written as [41]:

$$H^{+} = \left(H^{T}.H\right)^{-1}.H^{T} \tag{29}$$

3.2.2 Kernel Extreme Learning Machine (KELM)

The kernel ELM is an extension of ELM using a kernel function. The architecture is given by Fig. 3. The kernel matrix for ELM that is given by:

$$K_{ELM} = HH^{T}: \; K_{ELM}(i,j) = h(a_i).h\left(a_j\right) = K\left(a_i, a_j\right) \tag{30}$$

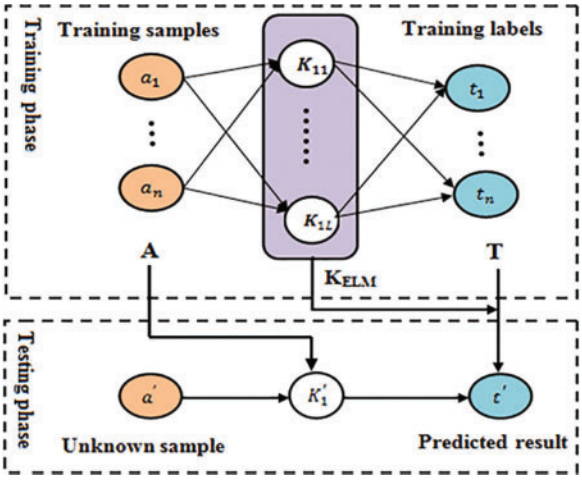

Figure 3: The structure of kernel extreme learning machine

The output function of the ELM classifier (Eq. (18)) can be presented compactly as:

$$f(a) = \left(\begin{bmatrix} K(a,a_1) \\ \vdots \\ K(a,a_N) \end{bmatrix}^{T} \left(\frac{I}{C} + \mathrm{K}_{ELM} \right)^{-1} T \right) \tag{31}$$

In the suggested methodology, an RBF-kernel function is used.

4 Experimental Results

This section evaluates and analyzes the performance results of the developed intelligent detection model. Also, this section presents the used dataset to validate and evaluate the proposed techniques.

Additionally, a comparison with the existing state-of-the-art is provided to attest the effectiveness of the suggested techniques. The experiments were conducted using windows 10 with an Intel (R) Core (TM) processor i7-7700 CPU @ 3.60 GHz 3.60 GHz.

4.1 Dataset Description

To validate the performance of the developed methods, a healthcare dataset named WUSTL-EHMS-2020 is used [28]. The WUSTL-EHMS-2020 dataset contains both the network flow and patient biometric data. This dataset consists of different types of cyberattacks such as MitM attacks, data injection, and spoofing. Table 3 displays statistical data from the WUSTL-EHMS-2020 dataset. This dataset contains 44 features, 35 of which are network flow data, eight biometric features from patients' data, and one label feature.

Table 3: Dataset statistical information

Measurement	Value
Dataset size	4.4 MB
Number of normal samples	14272 (87.5%)
Number of attack data	2046 (12.5%)
Total number of data	16318

4.2 Performance Metrics

The efficiency of the developed anomaly detection method is tested according to the following different performances. These different performances are computed using the entities TP, TN, FP, and FN, see Table 4 below.

$$Sensitivity\ \ (Se) = \frac{TP}{TP + FN} \tag{32}$$

$$Specificity\ \ (Sp) = \frac{TN}{TN + FP} \tag{33}$$

$$\text{Accuracy}\ \ (\text{Acc}) = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{34}$$

$$F_{Score} = 2 \times \frac{TP}{D} \times \frac{TP}{J} \div \left(\frac{TP}{D} + \frac{TP}{J} \right) \tag{35}$$

$$\text{With: } \begin{cases} D = TP + FP \\ J = TP + FN \end{cases}$$

Table 4: Confusion matrix

Total observations		Predicted labels	
		Attack	Normal
True labels	Attack	TP	FN
	Normal	FP	TN

4.3 Results and Discussion

Table 5 presents the performance results of our proposed methods. As shown, the KPLS-KELM and KPCA-KELM models achieved higher accuracy rates with 99.95% and 99.9%, respectively. The accuracy rate of the KPLS model was 71.8%, the lowest performance result of all models. The specificity rates for the KPLS-KELM and KPCA-KELM models were 99.9% and 99.8%, respectively. The KPLS, on the other hand, had the lowest specificity rate of all models (44.30%). The KPLS-KELM and KPCA-KELM models both obtained 100% in terms of sensitivity, while the KPLS model obtained 99.30%. The prediction times for the KPLS-KELM and KPCA-KELM models were 0.0468 and 0.0408, respectively. The KPLS took 6.40 s longer than the other models.

Table 5: Performance results for proposed detection methods

Methods	Accuracy	Specificity	Sensitivity	F-score	Prediction time
KPLS	71.8	44.30	99.30	77.88	6.40
KPLS-KELM	99.95	99.9	100	99.95	0.0468
KPCA-KELM	99.9	99.8	100	99.9	0.0408

Fig. 4 shows, the KPLS-KELM and KPCA-KELM models performed better than the KPLS model. As Fig. 5 demonstrates the performance result of KPLS, KPLS-KELM, and KPCA-KELM models in terms of receiver operating characteristics (ROC) to analyze the false positive rate of all models. The ROC curve has an x-axis that reflects the false positive rate and a y-axis that depicts the true positive rate.

Fig. 6 shows the learning and testing time of the proposed approach. The KPCA-KELM model operates better than the KPLS-KELM model in terms of learning time. For prediction time, the KPCA-KELM model outperforms the KPLS-KELM model.

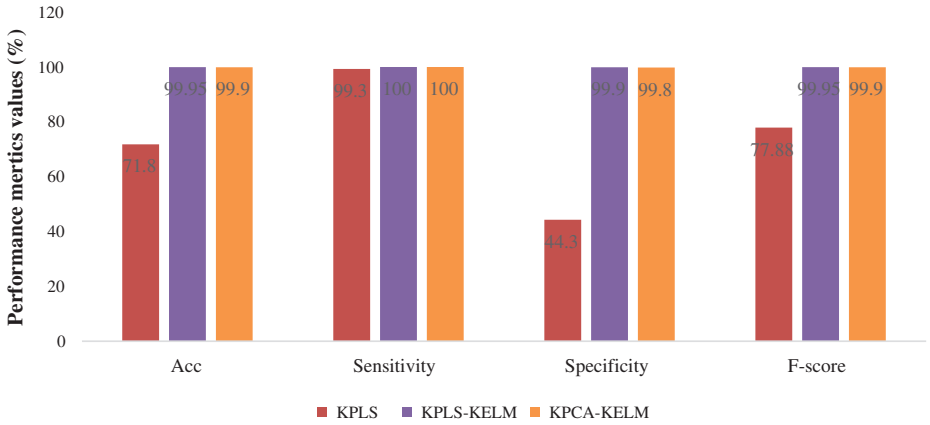

Figure 4: Performance results of proposed detection techniques

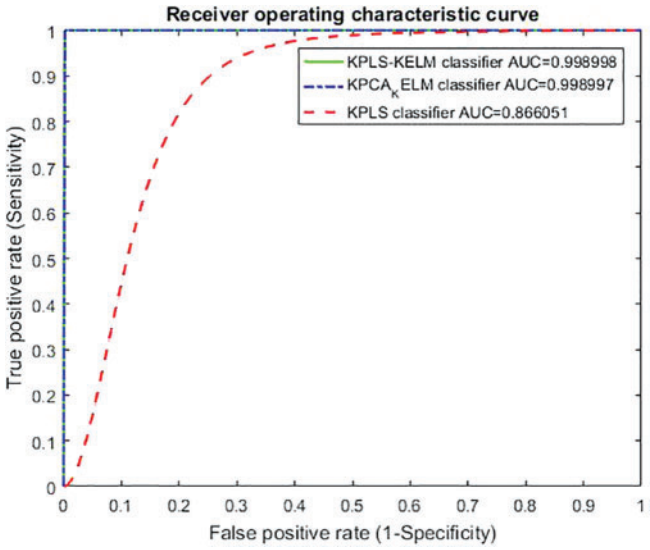

Figure 5: Receiver operating characteristic curve (ROC) of the proposed methods

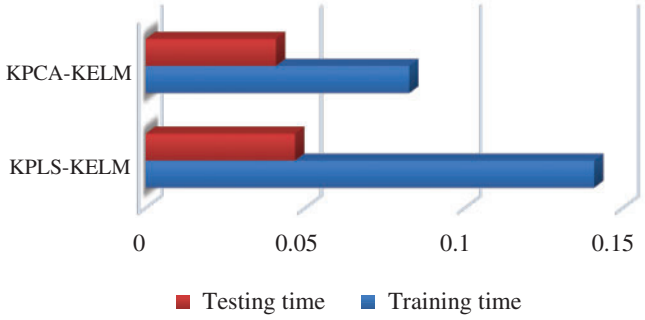

Figure 6: The training and testing time of the proposed techniques

5 Comparative Study

Table 6 compares the suggested methods to the existing state-of-the-art approaches. Our detection techniques outperformed the suggested method in [21], who used sample-selected extreme learning machine technique; the proposed KPLS-KELM and KPCA-KELM methods enhanced the accuracy rate by 0.88% and 0.83%, respectively. In addition, the training time of the proposed KPLS-KELM and KPCA-KELM techniques was reduced by 4.38 and 4.44. In comparison to the suggested model in [25], which utilized particle swarm optimization with a random forest classifier, the KPLS-KELM and KPCA-KELM methods improved the accuracy rate by 0.16% and 0.11%, respectively. In comparison to Hady et al. [28], who used the K-nearest neighbor and support vector machine, the proposed KPLS-KELM and KPCA-KELM methods improved the accuracy by 7.89% and 7.84% respectively, when compared to the K-nearest neighbor model and 7.5% and 7.45% respectively, when compared to support vector machine.

Table 6: Comparison with the existing state-of-the-art methods

Ref	Method used	Accuracy (%)	Training time
An et al. [21]	Sample selected extreme learning machine	99.07	4.52
An et al. [21]	Extreme learning machine	96.09	4.15
Saheed et al. [25]	Particle swarm optimization-random forest	99.79	0.11
Hady et al. [28]	K-nearest neighbor	92.06	0.21
Hady et al. [28]	Support vector machine	92.45	55.23
Proposed work	Kernel partial least square-kernel extreme learning machine	99.95	0.14
Proposed work	KPCA-KELM	99.9	0.08

6 Conclusions

This paper proposed intelligent intrusion detection based machine learning to overcome cyberattacks in IoMT networks. The proposed approach uses data-driven techniques to select the important data features and the KELM classifier to effectively identify cyberattacks in IoMT networks. To validate the proposed techniques, a modern healthcare dataset named WUSTL-EHMS-2020 is used. The proposed approaches achieved a higher performance result in contrast with the known approaches in terms of accuracy and specially in training time due of using a kernel extreme machine classifier characterized by one hidden layer. In the future, others data driven techniques and deep learning approach to detect intrusion in IoMT will be investigate. Additionally, more feature selection techniques will investigate and compare with our proposed work.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at University of Tabuk for funding this work through Research no. S-1443-0111.

Funding Statement: This work was supported by the Deanship of Scientific Research at the University of Tabuk through Research No. S-1443-0111.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Alanazi and A. Aljuhani, "Anomaly detection for industrial internet of things cyberattacks," *Computer Systems Science and Engineering*, vol. 44, no. 3, pp. 2361–2378, 2022.

[2] T. H. H. Aldhyani and H. Alkahtani, "Cyber security for detecting distributed denial of service attacks in agriculture 4.0: Deep learning model," *Mathematics*, vol. 11, pp. 1–19, 2023.

[3] S. Li, Y. Li, W. Han, X. Du, M. Guizani *et al.,* "Malicious mining code detection based on ensemble learning in cloud computing environment," *Simulation Modelling Practice and Theory*, vol. 113, pp. 1–16, 2021.

[4] M. E. Karar, O. Reyad and H. I. Shehata, "Deep forest-based fall detection in internet of medical things environment," *Computer Systems Science and Engineering*, vol. 45, no. 3, pp. 2377–2389, 2023.

[5] H. Mrabet, S. Belguith, A. Alhomoud and A. Jemai, "A survey of IoT security based on a layered architecture of sensing and data analysis," *Sensors*, vol. 20, no. 13, pp. 1–19, 2020.

[6] T. Ghazal, "Positioning of uav base stations using 5G and beyond networks for IoMT applications," *Arabian Journal for Science and Engineering*, vol. 46, pp. 1–12, 2021.

[7] R. B. Arslan and Ç. Candan, "A wearable system implementation for the internet of medical things (IoMT)," in *Proc. of the 44th Annual Int. Conf. of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Glasgow, United Kingdom, pp. 2447–2450, 2022.

[8] S. Ksibi, F. Jaidi and A. Bouhoula, "Cyber-risk management within IoMT: A context-aware agent-based framework for a reliable e-health system," in *Proc. of the 23rd Int. Conf. on Information Integration and Web Intelligence*, Linz, Austria, pp. 547–552, 2021.

[9] A. Aljuhani, P. Kumar, R. Kumar, A. Jolfaei and A. K. M. N. Islam, "Fog intelligence for secure smart villages: Architecture, and future challenges," *IEEE Consumer Electronics Magazine*, vol. 11, pp. 1–10, 2022.

[10] J. Qadir, B. S. Abajo, A. Khan, B. G. Zapirain, I. D. Díez *et al.,* "Towards mobile edge computing taxonomy, challenges, applications and future realms," *IEEE Access*, vol. 8, pp. 189129–189162, 2020.

[11] I. V. Pustokhina, D. A. Pustokhin, D. Gupta, A. Khanna, K. Shankar *et al.,* "An effective training scheme for deep neural network in edge computing enabled internet of medical things (IoMT) systems," *IEEE Access*, vol. 8, pp. 107112–107123, 2020.

[12] S. A. Parah, J. A. Kaw, P. Bellavista, N. A. Loan, G. M. Bhat *et al.,* "Efficient security and authentication for edge-based internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15652–15662, 2021.

[13] S. Tuli, N. Basumatary, S. S. Gill, M. Kahani, R. C. Arya *et al.,* "Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated IoT and fog computing environments," *Future Generation Computer Systems*, vol. 104, pp. 187–200, 2020.

[14] J. P. A. Yaacoub, M. Noura, H. N. Noura, O. Salman, E. Yaacoub *et al.,* "Securing internet of medical things systems: Limitations issues and recommendations," *Future Generation Computer Systems*, vol. 105, pp. 581–606, 2020.

[15] A. Aljuhani, "Machine learning approaches for combating distributed denial of service attacks in modern networking environments," *IEEE Access*, vol. 9, pp. 42236–42264, 2021.

[16] A. Almogren, I. Mohiuddin, I. U. Din, H. Almajed and N. Guizani, "FTM-IoMT: Fuzzy-based trust management for preventing sybil attacks in internet of medical things," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4485–4497, 2021.

[17] M. A. Allouzi and J. I. Khan, "Identifying and modeling security threats for IoMT edge network using markov chain and common vulnerability scoring system (cvss)," arXiv-CS-Cryptography and Security, 2021.

[18] P. Spadaccino and F. Cuomo, "Intrusion detection systems for IoT: Opportunities and challenges offered by edge computing," arXiv-CS-Cryptography and Security, 2020.

[19] A. Gatouillat, Y. Badr, B. Massot and E. Sejdić, "Internet of medical things: A review of recent contributions dealing with cyber-physical systems in medicine," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3810–3822, 2018.

[20] A. A. Toor, M. Usman, F. Younas, A. C. M. Fong, S. A. Khan *et al.,* "Mining massive e-health data streams for IoMT enabled healthcare systems," *Sensors*, vol. 20, no. 7, pp. 1–24, 2020.

[21] X. An, X. Zhou, X. Lü, F. Lin and L. Yang, "Sample selected extreme learning machine based intrusion detection in fog computing and MEC," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–10, 2018.

[22] T. Alatawi and A. Aljuhani, "Anomaly detection framework in fog-to-things communication for industrial internet of things," *Computers, Materials & Continua*, vol. 73, no. 1, pp. 1067–1086, 2022.

[23] P. R. Grammatikis, P. Sarigiannidis, G. Efstathopoulos, T. Lagkas, G. Fragulis *et al.,* "A Self-learning approach for detecting intrusions in healthcare systems," in *Proc. of IEEE Int. Conf. on Communications*, Montreal, QC, Canada, 2021.

[24] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. taouali, N. Liouane *et al.,* "Anomaly-based intrusion detection system in IoT using kernel extreme learning machine," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1–12, 2022.

[25] Y. K. Saheed and M. O. Arowolo, "Efficient cyber-attack detection on the internet of medical things-smart environment based on deep recurrent neural network and machine learning algorithms," *IEEE Access*, vol. 9, pp. 161546–161554, 2021.

[26] S. Ketu and P. K. Mishra, "Scalable kernel-based svm classification algorithm on imbalance air quality data for proficient healthcare," *Complex & Intelligent Systems*, vol. 7, pp. 2597–2615, 2021.

[27] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy *et al.,* "AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning," in *Proc. of 9th Annual Computing and Communication Workshop and Conf. (CCWC)*, University of Nevada, Las Vegas, USA, pp. 0305–0310, 2019.

[28] A. A. Hady, A. Ghubaish, T. Salman, D. Unal and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: A comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[29] P. Kumar, G. P. Gupta and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," *Computer Communications*, vol. 166, pp. 110–124, 2021.

[30] S. A. Rahman, H. Tout, C. Talhi and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Network*, vol. 34, pp. 310–317, 2020.

[31] H. Lahdhiri, A. Aljuhani, K. B. Abdellafou and O. Taouali, "An improved fault diagnosis strategy for process monitoring using reconstruction based contributions," *IEEE Access*, vol. 9, pp. 79520–79533, 2021.

[32] H. Lahdhiri and O. Taouali, "Reduced rank kpca based on glrt chart for sensor fault detection in nonlinear chemical process," *Measurement*, vol. 169, pp. 1–11, 2021.

[33] M. Said, K. Ben Abdellafou and O. Taouali, "Machine learning technique for data-driven fault detection of nonlinear processes," *Journal of Intelligent Manufacturing*, vol. 31, pp. 865–884, 2020.

[34] R. Wang, F. Nie, R. Hong, X. Chang, X. Yang *et al.,* "Fast and orthogonal locality preserving projections for dimensionality reduction," *IEEE Transactions on Image Processing*, vol. 26, pp. 5019–5030, 2017.

[35] S. Naung, Y. Feng, P. Santosa, R. Hartanto and K. Sakurai, "Machine learning-based IoT-botnet attack detection with sequential architecture," *Sensors*, vol. 20, pp. 1–120, 2020.

[36] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. of Military Communications and Information Systems Conf. (MilCIS)*, Canberra, Australia, pp. 1–6, 2015.

[37] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai *et al.,* "N-baiot-network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[38] O. Taouali, I. Jaffel, H. Lahdhiri, M. F. Harkat and H. Messaoud, "New fault detection method based on reduced kernel principal component analysis (rkpca)," *International Journal of Advanced Manufacturing Technology*, vol. 85, pp. 1547–1552, 2016.

[39] C. Yan, X. Chang, M. Luo, Q. Zheng, X. Zhang *et al.,* "Self-weighted robust lda for multiclass classification with edge classes," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, pp. 1–19, 2020.

[40] C. Yan, X. Chang, Z. Li, W. Guan, Z. Ge *et al.,* "Zeronas: Differentiable generative adversarial networks search for zero-shot learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 9733–9740, 2021.

[41] B. Schölkopf, A. Smola and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.