Official version: A. Abusitta, G. H. S. de Carvalho, O. Adel Wahab, T. Halabi, B. C. M. Fung, and S. Al Mamoori. Deep learning-enabled anomaly detection for IoT systems. Internet of Things (IoT), 21(100656):1-13, April 2023. Elsevier.

Deep Learning-Enabled Anomaly Detection for IoT Systems

Adel Abusitta[a,*], Glaucio H.S. de Carvalho[b], Omar Abdel Wahab[c], Talal Halabi[d], Benjamin C. M. Fung[e], Saja Al Mamoori[a]

[a]*Department of Computer Science, University of Windsor, Windsor, ON, Canada*
[b]*Department of Computer Science & Engineering, Brock University, St. Catharines, ON, Canada*
[c]*Department of Computer Engineering and Software Engineering, Polytechnique Montreal, QC, Canada*
[d]*Department of Computer Science, Laval University, QC, Canada*
[e]*School of Information Studies, McGill University, Montreal, QC, Canada*

Abstract

Internet of Things (IoT) systems have become an intrinsic technology in various industries and government services. Unfortunately, IoT devices and networks are known to be highly vulnerable to security attacks that target data integrity and service availability. Moreover, the heterogeneity of the data collected from various IoT devices, together with the disturbances incurred within the IoT system, render the detection of anomalous behavior and compromised nodes more challenging compared to traditional Information Technology (IT) networks. As a result, there is a pressing need for effective and reliable anomaly detection to identify malicious data to guarantee that they will not be used in IoT-driven decision support systems. In this paper, we propose a deep learning-powered anomaly detection for IoT that can learn and capture robust and useful features, which cannot be significantly affected by unstable environments. These features are then used by the classifier to enhance the accuracy of detecting malicious IoT data. More specifically, the proposed deep learning model is designed based on a denoising autoencoder, which is adopted to obtain features that are robust against the heterogeneous environment of IoT. Experimental results based on real-life IoT datasets show the effectiveness of the proposed framework in terms of enhancing the accuracy

*Corresponding author
Email addresses: adel.abusitta@uwindsor.ca (Adel Abusitta), gdecarvalho@brocku.ca (Glaucio H.S. de Carvalho), omar.abdul-wahab@polymtl.ca (Omar Abdel Wahab), talal.halabi@ift.ulaval.ca (Talal Halabi), ben.fung@mcgill.ca (Benjamin C. M. Fung), sajak@uwindsor.ca (Saja Al Mamoori)

of detecting malicious data compared to the state-of-the-art IoT-based anomaly detection models.

Keywords: IoT Security, Deep Learning, IoT Anomaly Detection, Artificial Intelligence for Security.

---

1. Introduction

The Internet of Things (IoT) is becoming an essential component of many Information and Communications Technology (ICT) systems. It has brought new service paradigms across different sectors such as wearable health devices, autonomous transportation, and various smart city applications [1]. IoT relies on devices embedded with sensors that can transmit data to the cloud to perform data analytics and generate control decisions related to cyber-physical systems. According to the recent statistics, there are currently more than 26 billion connected and active IoT devices worldwide [2]. The number of IoT devices is expected to increase and reach around 75 billion in 2025 [2]. Many IoT systems are used by organizations to enhance safety and productivity. For example, manufacturers can use IoT-based solutions to analyze large amounts of data captured by sensor devices integrated into manufacturers' equipment, to enable data scientists and analysts to prevent and forecast critical and real-time problems such as engine breakdowns and other incidents. This, in turn, allows manufacturers to significantly boost productivity and safety [3] [4].

IoT is split into four different abstraction layers, i.e., physical layer collecting data using IoT sensors; network layer (cloud/edge communications) used for transferring the data among devices for processing; processing layer responsible for performing some processes and computational tasks with the help of the Cloud Computing; and the application layer delivered by the devices of end users. All these layers are subject to security threats [5, 6, 7, 8]. In recent years, the security of IoT devices have attracted tremendous research efforts [9]. IoT devices generate, gather, and process data that - in most cases - consist of sensitive information, making them largely vulnerable to serious security threats that can be exploited by attackers. Therefore, the integrity of the data gathered by IoT devices should be protected in real time, making the design of

effective anomaly detection systems extremely important [10].

Machine learning can play an important role in detecting anomalous and malicious data through training detection models on both malicious and benign IoT data. In the literature, there are several machine learning techniques (e.g., [11] [12]) that are used to detect malicious data. However, these techniques are fundamentally based on the assumption that the data used for the training are homogeneous (extracted from similar sources) and belong to the same data types (e.g., pixels). In other words, they are not designed to work with the inherent characteristics of data used by most practical IoT systems today, which are largely heterogeneous, consisting of various and mixed data types (e.g., images, text, graph data, stream data, time series data).

Deep learning is a category of machine learning that can be used to learn a good representation of data through layers of abstractions. This enables deep learning to outperform traditional machine learning techniques as the scale and the heterogeneity of data increases [13]. Recently, anomaly detection algorithms powered by deep learning have become increasingly useful in various domains. This paper addresses the challenges brought by the heterogeneous IoT systems by devising a deep learning-enabled framework for robust anomaly detection in IoT.

The proposed framework is based on a modified version of autoencoder, a denoising autoencoder [14] [15], which we adopt as a building block for the training of IoT-based Deep Neural Networks (DNN) (Figure 1). This architecture enables us to extract robust features from the IoT data by pre-training the DNN on large-scale heterogeneous unlabeled data. This step allows us to obtain "good" representation of IoT data that leads to better detection or classification accuracy when training a machine learning classifier (e.g., SVM) on labeled data. In other words, the pre-training process enables us to obtain features that are robust against the heterogeneous environment of IoT systems. The proposed framework consists of two different layers of denoising autoencoder (Figure 1): neutral and decision layers. The neutral layer is used to isolate unnecessary features (or neutral features), which can be seen in both anomalous and benign IoT data. Isolating these features boosts unbiased detection and minimizes the chances of degrading the performance of the classifier [16]. The decision layer is used to capture useful and robust IoT-related features that are mainly used by the classifier

to distinguish between anomalous and benign IoT data.

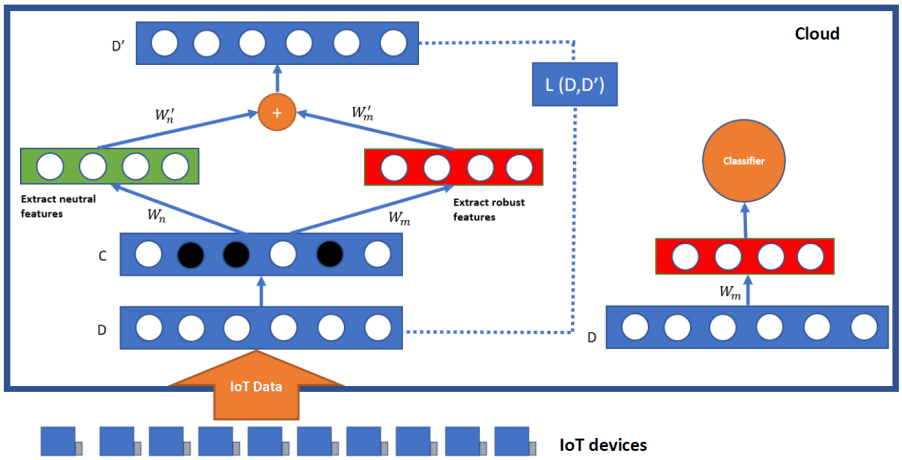


Figure 1: Extracting robust and useful features for IoT anomaly detection. The proposed IoT-based DNN for anomaly detection is trained not only to extract robust and useful features, but also is trained to isolate unnecessary features (neutral features). This is turn enhances the detection accuracy when using a classifier, compared to the traditional DNNs

Our contributions in this paper are summarized as follows:

- Designing a robust anomaly detection framework for the IoT, which facilitates and improves the detection of malicious data produced by compromised and/or attacked heterogeneous IoT devices.

- Proposing an efficient model to extract robust IoT-based features and isolate unnecessary features. This, in turn, enables us to improve the performance of anomaly detection in the IoT.

- Studying and evaluating the effectiveness of the proposed model using real-world IoT datasets. We compare our model with existing deep learning models used for anomaly detection in IoT.

The remainder of this paper is organised as follows. In Section 2, we discuss the related work. Section 3 presents the proposed framework. In Section 4, we present our empirical results. Finally, Section 5 concludes the paper.

2. Background and Related Work

This section discusses the recent machine learning and deep learning approaches used for anomaly detection in computer systems and IoT. In the literature, ML-powered for anomaly detection have been proposed in many works (e.g., [17, 18]). The machine learning methods for anomaly detection are divided into three approaches: supervised, unsupervised, or semi-supervised. In supervised machine learning-based anomaly detection [19, 20], the machine learning model is trained on labeled datasets, while in unsupervised machine learning-based anomaly detection [21, 22], the machine learning model works on learning patterns and features using unlabeled dataset. Finally, the semi-supervised method [23, 24] adopts both labeled and unlabeled datasets in the training and learning process [25]. Below we discuss recent machine learning-based anomaly detection.

Lopez-Martin et al. [26] proposed a conditional variational autoencoder for anomaly detection in IoT. They make a few changes on the traditional architecture of a conditional variational autoencoder by integrating the intrusion labels into the layers used for decoding. Their method can perform feature reconstruction, which is useful to address missing information in the training IoT datasets. Diro and Chilamkurti [27] proposed a distributed deep learning approach for anomaly detection in IoT, where the concept of Fog Computing has been adopted to identify attacks in IoT systems. The goal of their study is to show the strength of deep learning models in attack detection compared to traditional machine learning models.

Moustafa et al. [28] designed a unified intrusion detection framework that integrates Naive Bayes, decision trees, and artificial neural networks. The framework can be used as a classifier to identify botnet attacks against IoT-related protocols: MQTT, DNS, and HTTP. To this end, the researchers generated new statistical flow features from these protocols by analysing their potential properties [28]. More recently, Aversano et al. [29] proposed a deep learning approach for anomaly detection in IoT. They used an autoencoder as a building block to train a deep neural network. The use of an autoencoder enables them to apply feature reduction, which is used to enhance the detection accuracy. Similarly, Sarma [30] proposed a two-stage framework for attack

detection in IoT: feature extraction and classification. The feature extraction phase is considered as the preliminary phase, where the features are created by every application. Thereafter, a deep convolutional neural network model is used to classify and identify attacks.

Saxe and Berlin [31] also presented a DNN for intrusion detection. They designed multi-layer perceptron (MLP)-powered IDS. MLP is considered as a type of feed-forward NN. In [31], they adopt a rectified linear units (ReLU) as activation functions, which allows the DNN to significantly enhance the anomaly detection performance compared to other functions [32] [33]. They also used a Bayesian model (BM) to evaluate to which extent a suspicious activity is considered real intrusion. Similarly, Dahl et al. [34] adopt a deep learning method to enhance the anomaly detection process. In particular, they apply a restricted Boltzmann machine (RBM), which can be used as a building block for designing DNN. This can be considered as a pre-training process that can be used to obtain useful features to enhance the accuracy.

Huang and Stokes [35] proposed a neural network (NN) for multi-task training. At the beginning, the NN is used to decide whether a given binary or malicious activity is an attack or not. Then, they applied NN to classify the family of an intrusion. Kolosnjaji et al. [36] designed a convolutional neural network (CNN) combined with LSTM (or Long Short-Term Memory) networks to explore the intrusion families. The convolution layer is applied to understand the correlation of the features related to the intrusion. The output of the CNN is then exploited to train the LSTM to find dependencies of features.

More recently, Ullah et al. [37] proposed a deep learning model approach for the detection of anomalous behaviour in IoT networks. To this end, they adopted a recurrent neural network powered by Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) to implement anomaly detection system in IoT. They also used a Convolutional Neural Network (CNN) to analyze IoT-related input features with the aim of keeping important information. Such a hybrid deep learning method allowed them to build a lightweight deep learning model used for binary classification. Similarly, Zhou et al. [38] designed DB-CGAN, an integrated deep learning model powered by Generative adversarial networks (GANs) [39]. The adversarial training allowed them to build

robust data used to improve the classification and detection accuracy. Kale et al. [40] proposed a deep learning model for anomaly detection, which can be called as three-stage deep learning anomaly detection for IoT. They designed a unified framework that integrated three techniques: K-means clustering, GANomaly, and CNN.

Recently, Abusitta et al. [41] designed a deep learning-powered intrusion detection system in the Cloud. They use a modified version Autoencoder (Denoising Autoencoder) to train the DNN efficiently. The Denoising Autoencoder allows them to improve the detection accuracy under incomplete information. Thus, it allows the IDS to proactively decide about suspicious activity even in the presence of incomplete feedback. Abusitta et al. [42] also designed a proactive collaborative malware detection system. They adopted the Denoising Autoencoder to obtain all nodes' decisions from incomplete decisions. As a result, they enhanced the classification and detection accuracy in real time [42].

More recently, a new approach for identifying malware in dynamic (or non-stationary) environments was proposed in [43]. The researchers use deep learning methods to extract high-level features used to enhance the detection of malware in non-stationary environments. Specifically, their framework is based on a denoising autoencoder, which is used as a building block to train a deep neural network. Overall, for an IoT network, a framework for anomaly detection under a largely heterogeneous and unstable environment is still missing. Most of the proposed deep learning models are generally applied on high-quality data [44]. They hence do not work effectively on corrupted IoT data or under highly noisy environments.

In this paper, we propose a deep learning-powered anomaly detection for IoT systems that can learn and capture robust features, which cannot be significantly affected by heterogeneous environments (i.e., IoT systems). These features are then used by the machine learning classifier such as SVM to enhance the accuracy of detecting malicious IoT data. Unlike other DNN-based anomaly detection approaches for IoT systems, the proposed framework is powered by a new layer used to isolate unnecessary features (or neutral features), which are found in both anomalous and benign IoT data. Isolating these features allows us to boost unbiased detection and hence enhances the detection accuracy.

3. The Proposed Anomaly Detection Framework

This section introduces the proposed framework for anomaly detection in IoT networks with heterogeneous devices. The framework is based on a denoising autoencoder, which is used as a building block in the DNN. We first present the basic concepts of a traditional autoencoder, then we describe the proposed deep learning approach for anomaly detection in IoT.

The purpose of an autoencoder is to abstract data and try to learn "good" representations of data by applying unsupervised training (i.e., unsupervised learning) [45]. It can be used as a building block in the DNN. Having such a building block allows us to pre-train the DNN and obtain initial weights that are able to accelerate the training process [46], and hence improve the performance of the prediction and/or classification accuracy when attached with a classifier (e.g., SVM, Logistic Regression).

As can be seen in Figure 2, the training of an autoencoder is done by mapping an input $d \in [0,1]^{dim}$ to a hidden layer h, where dim represents the input's dimension. The following function shows the procedure of mapping:

$$\begin{aligned} h &= f_\theta(d) \\ f_\theta(d) &= Sig(W * x + b) \end{aligned} \tag{1}$$

Sig is the sigmoid function. $\theta = \{W, b\}$, where W and b are the neural weights and bias, respectively. The output hidden layer h is then used to obtain the input d'as follows:

$$\begin{aligned} d' &= f_{\theta'}(h) \\ g_{\theta'}(h) &= Sig(W' * h + b') \end{aligned} \tag{2}$$

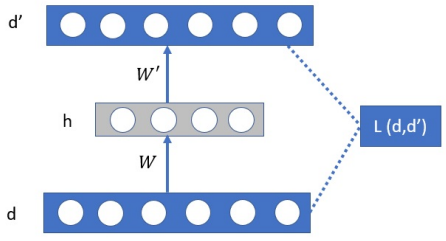


Figure 2: The traditional autoencoder architecture.

The purpose of training the autoencoder is to optimize its parameters in such a way that the reconstruction error between the output (d') and input (d) is very small (minimization problem). To achieve this, the optimization problem is formulated as follows:

$$\begin{aligned} \theta^*, \theta'^* &= arg \ \ min_{\theta,\theta'} \ \frac{1}{n}\sum_{i=1}^{n} L(d^{(i)}, d'^{(i)}) \\ &= \frac{1}{n}\sum_{i=1}^{n} L(d^{(i)}, g_{\theta'}(f_\theta(d^{(i)}))) \end{aligned} \tag{3}$$

where L is a loss function, e.g., squared error $L(d, d') = \|d - d'\|^2$. There is also another loss that we adopt in this paper called cross-entropy error: $L_H(d, d')$= $-\sum_{i=1}^{d}[d_i log d'_i + (1 - d_i) log(1 - d'_i)]$. The cross-entropy error is shown to be better than the squared error in classification and detection problems [47].

3.1. The Proposed Denoising Autoencoder for Anomaly Detection in IoT

While the traditional autoencoder can enhance the detection of most AI-powered applications by abstracting and mapping an input to an intermediate representation, the traditional autoencoder is not able to obtain robust and useful features in heterogeneous environments [14] [15]. The denoising autoencoder can be used to address these problems. Unlike the traditional autoencoder, the denoising autoencoder is trained to reconstruct input data after applying some noises on the input data. This architecture allows us to obtain useful and robust features in the presence of unstable environments such as IoT systems, in which edge devices may succumb to physical disturbances at the control and communication layers.

The training of a denoising autoencoder is done as follows. First, we apply some noises to the IoT sensor data d before mapping it to the hidden layer h. Then, we try to reconstruct d from the hidden layer [14]. The corrupted version of d (after applying noise) is denoted by c. The noise function used in this process is a mask function [14]. The corrupted version c is then used for mapping to the hidden layer h. Thereafter, we try to obtain the value d' (the difference between d and d' should be very small) . The following equation shows how to map d to the hidden layer h. Note that unlike a

traditional autoencoder, we map c to h instead of d.

$$\begin{aligned} h &= f_\theta(c) \\ f_\theta(c) &= Sig(W * c + b) \end{aligned} \tag{4}$$

The representation h can then be used to reconstruct the data d'. The following equation shows the procedure:

$$\begin{aligned} d' &= g_{\theta'}(h) \\ g_{\theta'}(h) &= Sig(W * h + b) \end{aligned} \tag{5}$$

Figure 3 shows the architecture of the proposed IoT-based denoising autoencoder. Like in the traditional autoencoder, the following equation shows the optimization function used in the training process:

$$\begin{aligned} \theta^*, \theta'^* &= arg \ \ min_{\theta,\theta'} \frac{\sum_{j=1}^{n} L(d^{(j)}, d'^{(j)})}{n} \\ &= \frac{\sum_{j=1}^{n} L(d^{(j)}, g_{\theta'}(f_\theta(c^{(j)})))}{n} \end{aligned} \tag{6}$$

Equation (6) represents the optimization function used for training the denoising autoencoder. The denoising autoencoder is trained to reconstruct the original input data after applying noises on it.

The proposed IoT-based denoised autoencoder can be combined with two layers as shown in Figure 4: neutral and decision layers. The neutral layer is adopted to isolate neutral features, which are available in each IoT data (malicious/incorrect or benign

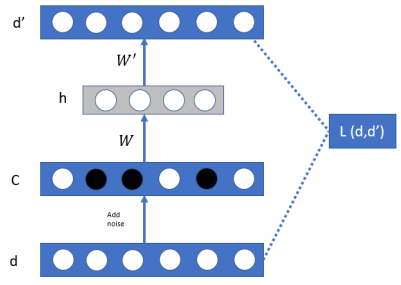


Figure 3: An IoT-based denoising autoencoder.

IoT data). This is useful to avoid biased detection [16]. The decision layer is then used to capture useful and necessary features that can be used to distinguish between anomalous and benign IoT data.

The two-layer denoising autoencoder is considered as a modified-version denoising autoencoder [16] and can be trained as follows. First, we train the parameters related to the neutral layer (neutral hidden representation) on the benign IoT data. To learn these parameters, a denoising autoencoder is used as we will explain in Algorithm 1. Then, the corrupted IoT data (anomalous or benign data) is mapped to two hidden layers (neutral and decision hidden layers). Mapping IoT data to the neutral hidden layer allows us to capture and isolate neutral features [16].

Algorithm 1 shows the training process of the proposed IoT-based denoising autoencoder. In this algorithm, we first apply noise to input d in order to obtain c (the noisy version of d). The value c is then used by the mapping function to produce h (the hidden layer). The hidden layer is then used to obtain (reconstruct) d' with the aim of making d and d' very close. Note that we use lev in Algorithm 1, which represents the percentage of noises applied on the input data.

The outputs of Algorithm 1 are the initial parameters, which are used to obtain the intermediate representation that can be used as an input to the classifier, as shown in Figure 5. In our experiments, we adopt two types of classifiers: SVM and binary logistic regression classifiers.

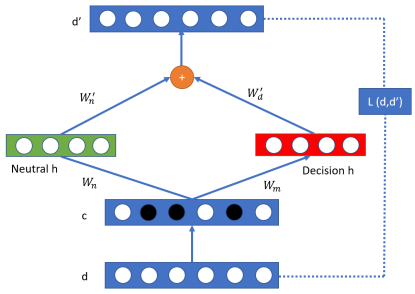


Figure 4: An IoT-based denoising autoencoder using two layers.

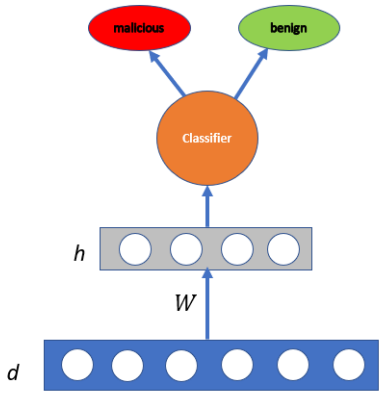


Figure 5: The proposed IoT-based Stacked denoising autoencoder.

Algorithm 1 IoT-based Denoising Autoencoder Algorithm for Anomaly Detection

procedure IOT_DA_TRAINING($d, lr, epoch, bt, \theta$)

Require: d= $[d_1 - d_n]$ ▷ Input IoT training data

Require: lr ▷ learning rate

Require: $epoch$ ▷ the number of epochs to be iterated

Require: bt ▷ the number of batches

Require: θ=$\{W, b, b_h\}$ ▷ parameters of a DA

$j = 0$

while $j \leq epoch$ **do**

$k = 0$

while $k \leq bt$ **do**

$c = GetCorruptedData(d, lev)$ ▷ lev is corruption level

$h = SigmoidF(c * W + b)$

$d' = SigmoidF(h * W^T + b_h)$

$L_H(d, d')$= $-\sum_{j=1}^{d}[d_j log d'_j + (1 - d_j) log(1 - d'_j)]$

$CostValue = mean(L(d, d'))$

g=determining the gradients with respect to θ

while θ_j,$g_j \in (\theta$,g) **do**

θ_j = θ_j - $lr * g_j$

end while

end while

$k = k + 1$

end while

end procedure

It is worth mentioning that the denoising autoencoders can be stacked before applying the classifier [15]. This can be done by making the hidden layer produced from the first denoising autoencoder as an input to the second denoising autoencoder, and the output layer produced from the second autoencoder as an input to the third denoising autoencoder and so on. The stacked denoising autoencoder allows us to obtain a much better representation that can then be used as an input to the classifier [15]. Algorithm 2 shows the training algorithm of the stacked denoising autoencoder. For each denoising authoencoder added to the network, the procedure $IoT\_DA\_Training$ is called (from Algorithm 1) to learn the corresponding parameters related to this layer.

Algorithm 2 IoT-based Stacked Denoising Autoencoder Algorithm

procedure IOT_STACKED_DA(d,lr,$epoch$,b,h,Θ)
Require: d= $d_1, d_2, ..., d_n$
Require: $h = h_1, h_2, ..., h_z$
Require: $\Theta = \theta_1, \theta_2, ..., \theta_z$ ▷ θ_j = $\{W_j, b_j, b_{hj}\}$
Require: O= $O_1, O_2, ...O_l$ ▷ hidden layer's output
θ_1 = IoT_DA_Training($x, lr, epoch, b, \theta_1$)
j=0
while $j \leq n$ **do**
$o_{1,j}$=Sig(x_j*W_1+b_j)
j=j+1
r=2
while $r \leq lr$ **do**
$\theta_r = IoT\_DA\_Training(O_{r-1}, l, epoch, b, \theta_r)$
$r = r + 1$
k=0
while $k \leq n$ **do**
$o_{r,k}$=σ(o_{r-1}*W_r+b_r)
$k = k + 1$
end while
end while
end while
end procedure

After training the last layer using Algorithm 2, the network becomes ready to be used by any supervised machine learning or classifier to distinguish between malicious

and benign IoT data. For example, we may add a binary logistic regression as a new layer to the network as illustrated in Figure 6, to generate a deep neural network. After adding the binary logistic regression, the parameters of all layers are fine-tuned in order to reduce the error (i.e., minimization) between the predicted decision and the actual decision (obtained from the training data). To this end, the back-propagation algorithm [48] [49] [46] [15][14] is used as shown in Algorithm 3.

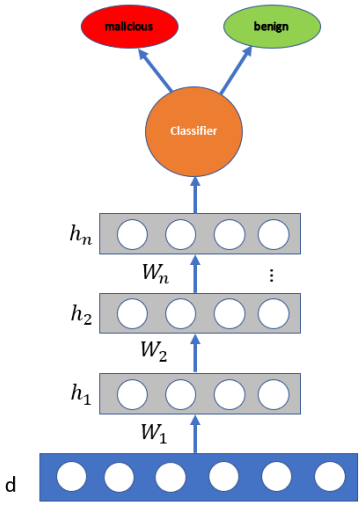


Figure 6: The Proposed Deep Neural Networks for Anomaly Detection in IoT.

Algorithm 3 IoT-based Fine Tuning Algorithm

procedure FINETUNING($d, lr, epoch, b, h, \Theta$)

Require: d= $d_1, d_2, ..., d_n$

Require: $h = h_1, h_2, ..., h_z$

Require: Θ = [$\theta_1 - \theta_z$], θ_1 = IoT_DA_Training (x, lr, e, b, θ_1)

Require: O= $O_1, O_2, ...O_l$ ▷ hidden layer's output

while $e \leq epoch$ **do**

CostValue = $\frac{1}{|D|} = L(\theta = \{W, b\}, D)$

grt= determine the gradient with respect to θ

while θ_i,$grt_i \in (\theta$,$grt)$ **do**

θ_i = θ_i - $l * grt_i$

end while

if validLoss $<$ optimalvalidLoss **then**

$optimalEpoch = epoch$

$optimalParams = \theta$

$optimalvalidLoss = validLoss$

end if

end while

return $optimalParams$

end procedure

4. Experimentation and Results

In this section, we evaluate the proposed framework for anomaly detection in IoT. We used the DS2OS traffic traces dataset [50][51] for evaluation, which consists of data captured from different IoT devices to reflect the heterogeneity aspect. The dataset contains communication traces between several IoT devices. These devices are of eight types, i.e., light controllers, thermometers, motion sensors, washing machines, batteries, thermostats, smart doors and smartphones. Seven types of attacks are considered in the dataset, i.e., Denial of Service, probing, malicious control, malicious operation, network scan, spying, and wrong setup. The dataset stores $357,000$ observations with 13 dimensions. The dataset has been used in many state-of-the-art papers such as [5, 52].

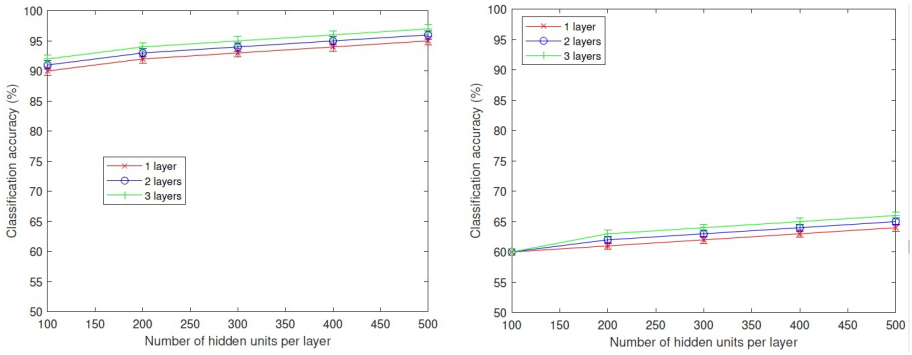
The proposed IoT-based DNN model was trained on the DS2OS traffic traces dataset. Table 1 shows the hyperparameters used in our experiments.

Table 1: Evaluation parameters.

Parameter	Values
Epochs Number	200
Number of layers in the NN	$\{1-3\}$
Number of units in the NN	$\{100-500\}$
Percentage of noise applied c	25%
Learning rate	0.005
Classifier (output layer)	SVM

Figure 7 shows the detection accuracy of our model compared to the multilayer perceptron (MLP)-based anomaly detection [53]. The MLP-based anomaly detection trains the deep neural network on IoT data without applying the pre-training process. In the conducted study, we use a different number of layers (ranging between 1 to 3) and Hidden Units (HUs) (from 100 to 500). Figure 7a shows that the classification accuracy (average) reported by our model at various numbers of HUs (from 100 to 500) is 94.6%, which is better than the result obtained using MLP-based anomaly detection (63.5%) as shown in Figure 7b. This can be justified due to the fact that the pre-training process applied in our model allows us to capture robust and useful features that lead to better classification accuracy. In other words, the pre-training process adopted by our model allows the DNN to have better initialization of the parameters to be used during applying the backprobagation algorithm and fine tuning processes [14] [15].

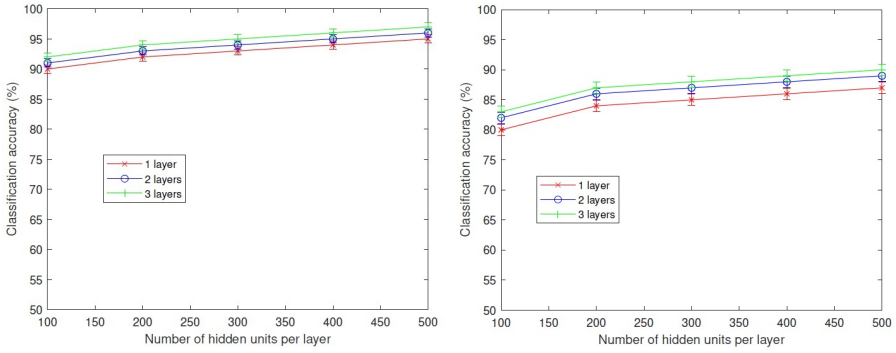
In Figure 8, we compare our method with a Stacked Autoencoder (SAE)-based anomaly detection [54]. The SAE adopts a traditional autoencoder in the pre-training process. In other words, it uses a traditional autoencoder as a building block for the DNN. As can be seen in Figure 8, the classification accuracy obtained by our model is better than SAE-based anomaly detection. Specifically, the detection accuracy (average) reported by the SAE-based anomaly detection at different numbers of HUs (100 - 500) is 84.7%, which is less than the result obtained by our model (94.6%). This can be interpreted due to the fact that the traditional autoencoder does not work properly



(a) our model (b) MLP

Figure 7: Classification accuracy. Our model is compared against MLP-based anomaly detection

when applied in heterogeneous and noisy environments such as IoT systems [14] [15].

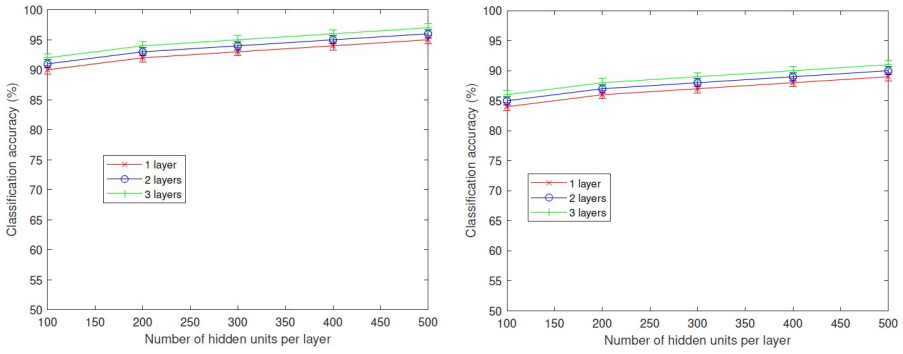


(a) Our model (b) SAE-based Anomaly Detection

Figure 8: Classification accuracy. Our model is compared against SAE-based anomaly detection

The proposed model was also compared with the Restricted Boltzmann machine (RBM)-based anomaly detection [55] [56]. An RBM can be used as a building block in deep-belief networks [57] [58]. As shown in Figure 9, our model outperforms the RBM-based anomaly detection for IoT. Numerically, the detection accuracy (average) reported by the RBM-based anomaly detection at different numbers of HUs (100 - 500) is 85.7%, which is less than the result obtained by our model (94.6%). While RBMs can be used to learn "good" representation of data, their performance can be degraded in the presence of noisy inputs [14] [15]. This is mainly due to the fact that the RBM

does not take into consideration noisy inputs during the pre-training process [14].



(a) Our model

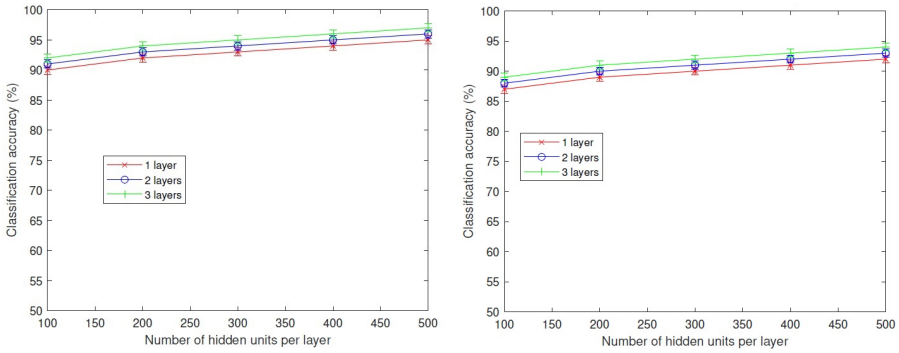
(b) RBM-based Anomaly Detection

Figure 9: Classification accuracy. Our model is compared against RBM-based anomaly detection

Finally, we compare our model with the Stacked Denoising Autoencoder (SDAE)-based anomaly detection in IoT [15] [26]. The SDAE can be used as a building block to train DNNs. As can be seen in Figure 10, our method yields increased accuracy compared to SDAE-based anomaly detection. The experiment was conducted using three hidden layers. As for the one-hidden-layer network, Figure 10 shows that the detection accuracy (in average) reported in our model is 93.5%, which is better than the one reported using one hidden layer in SDAE-based anomaly detection (85.50%) as shown in Figure 10b. Our model also yields improved detection accuracy compared to SDAE-based anomaly detection using two-hidden-layer and three-hidden-layer networks. The reason why our model yields a better accuracy than SDAE-based anomaly detection is that our model adopts a modified-version denoising autoencoder, which allows us to extract robust features and isolate unnecessary features. As a result, we obtained features that lead to a better classification.

4.1. Using logistic regression classifier as the output layer

In the previous section, we evaluated the proposed framework using the SVM as the output layer. In this section, we evaluate the proposed framework using the logistic regression classifier, which is widely used for binary classification in the literature. We



(a) Our model

(b) SDAE-based Anomaly Detection

Figure 10: Classification accuracy. Our model is compared against SDAE-based anomaly detection

used the DS2OS traffic traces dataset [59] for evaluation. Table 2 shows the hyperparameters used in our experiments.

Table 2: Evaluation parameters.

Parameter	Values
Epochs Number	200
Number of layers in the NN	100
Number of HUs per layer	$\{200 - 1000\}$
Percentage of noise applied c	30%
Learning rate	0.005
Classifier (output layer)	logistic regression

We compare our model with other state-of-the-art deep learning architectures for anomaly detection: RBM, SAE, and SDAE. As can be seen in Figure 11, our model yields increased anomaly detection accuracy compared to RBM, SAE, and SDAE. Specifically, the average accuracy reported by the proposed framework at different number of hidden nodes (the number of hidden units are from 200 to 1000) is 94.7%. This results is better than the results reported by applying RBM (87.4%, SAE (75.4%), and SDAE (87.6%).

We also evaluate the performance of the proposed framework based on the BoTNeTIoT-L01 dataset [60] [61], which is an IoT dataset for Intrusion Detection Systems (IDS).

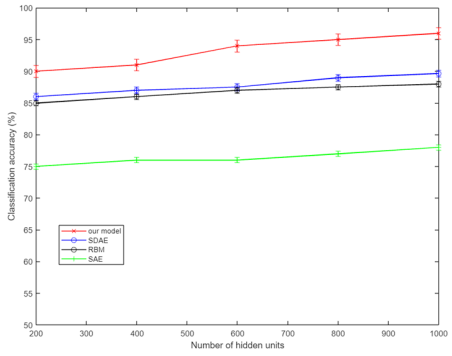


Figure 11: Classification accuracy. Our model is compared against RBM, SAE, SDAE-based anomaly detection (DS2OS traffic traces dataset)

Our model also improved the accuracy compared to RBM, SAE, and SDAE. More specifically, the average accuracy reported by the proposed framework at different number of hidden nodes (the number of hidden units are from 200 to 1000) is 94.9%. These results are better than the results reported by applying RBM (92.3%, SAE (84.3%), and SDAE (89.5%).

The reason why our method yields a better classification accuracy than other methods is that it adopts a modified-version denoising autoencoders [16] as a building block for training the DNN. While the traditional denoising autoencoders enable the DNN to extract robust features that somehow enhance accuracy, they are not able to isolate unnecessary features (or neutral features) that lead to degrading the classifier performance.

Acknowledgement

The financial support of the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

5. Conclusion and Future Work

In this paper, we proposed a deep learning-enabled anomaly detection framework for IoT systems. The proposed framework is based on a denoising autoencoder, which

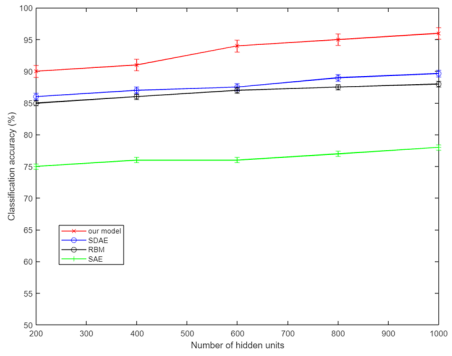


Figure 12: Classification accuracy. Our model is compared against RBM, SAE, SDAE-based anomaly detection (BoTNeTIoT-L01 dataset)

has been adopted as a building block in the Deep Neural Networks. The denoising autoencoder allows us to efficiently extract features that are robust against heterogeneous and noisy environments that characterize IoT systems. These features are then used by the classifier to distinguish between malicious and benign IoT data. Our framework also enables us to isolate features that lead to degrading the classification performance. Our results based on real-world datasets show the effectiveness of the proposed method in terms of enhancing the accuracy of identifying anomalous IoT data compared to other state-of-the-art methods.

In the future, we plan to build deep learning-enabled anomaly detection that is robust against adversarial attacks. Indeed, machine learning models including deep neural networks are shown to be highly vulnerable to adversarial attacks. Attackers can create "adversarial inputs" to fool a machine learning model by simply manipulating its inputs. Existing techniques for mitigating these attacks basically work on the assumption that the data used for training machine learning models are homogeneous - extracted from the same source - and belong to the same data type (e.g., image pixels). Hence, they are not designed to work with the inherent characteristics of data used by most practical AI-powered applications (e.g., securing IoT systems), which are largely heterogeneous and consist of various and mixed data types.

Additionally, most of the existing mitigation methods look at the challenge of addressing adversarial attacks as an inevitable continual arms race between defenders

and attackers (e.g., adversarial training methods). This assumption makes the defence strategies against adversarial attacks unstable and unable to guarantee good performance. In other words, there is still a possibility that sophisticated attackers launch catastrophic attacks targeting safety-critical applications (e.g., autonomous vehicles). Therefore, building more robust and resilient intrusion detection systems will continue to be a fundamental security problem.

References

[1] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, W. Ni, Anatomy of threats to the internet of things, IEEE Communications Surveys & Tutorials 21 (2) (2018) 1636–1675.

[2] I. Cvitić, D. Peraković, M. Periša, M. Botica, Novel approach for detection of iot generated ddos traffic, Wireless Networks 27 (3) (2021) 1573–1586.

[3] Y.-Q. Chen, B. Zhou, M. Zhang, C.-M. Chen, Using iot technology for computer-integrated manufacturing systems in the semiconductor industry, Applied Soft Computing 89 (2020) 106065.

[4] Y. Tan, W. Yang, K. Yoshida, S. Takakuwa, Application of iot-aided simulation to manufacturing systems in cyber-physical system, Machines 7 (1) (2019) 2.

[5] O. A. Wahab, Intrusion detection in the iot under data and concept drifts: Online deep learning approach, IEEE Internet of Things Journal.

[6] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, R. Candell, A survey of physics-based attack detection in cyber-physical systems, ACM Computing Surveys (CSUR) 51 (4) (2018) 1–36.

[7] J. E. Rubio, C. Alcaraz, J. Lopez, Preventing advanced persistent threats in complex control networks, in: European Symposium on Research in Computer Security, Springer, 2017, pp. 402–418.

[8] O. A. Wahab, J. Bentahar, H. Otrok, A. Mourad, How to distribute the detection load among virtual machines to maximize the detection of distributed attacks

in the cloud?, in: 2016 IEEE International Conference on Services Computing (SCC), IEEE, 2016, pp. 316–323.

[9] A.-R. Sadeghi, C. Wachsmann, M. Waidner, Security and privacy challenges in industrial internet of things, in: 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), IEEE, 2015, pp. 1–6.

[10] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, M. Guizani, A survey of machine and deep learning methods for internet of things (iot) security, IEEE Communications Surveys & Tutorials 22 (3) (2020) 1646–1685.

[11] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, H. Ming, Ad-iot: Anomaly detection of iot cyberattacks in smart city using machine learning, in: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2019, pp. 0305–0310.

[12] M. Hasan, M. M. Islam, M. I. I. Zarif, M. Hashem, Attack and anomaly detection in iot sensors in iot sites using machine learning approaches, Internet of Things 7 (2019) 100059.

[13] R. Chalapathy, S. Chawla, Deep learning for anomaly detection: A survey, arXiv preprint arXiv:1901.03407.

[14] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 1096–1103.

[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, Journal of machine learning research 11 (Dec) (2010) 3371–3408.

[16] R. Xia, Y. Liu, Using denoising autoencoder for emotion recognition., in: Interspeech, 2013, pp. 2886–2889.

[17] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, M. Guizani, Corrauc: a malicious bot-iot traffic detection method in iot network using machine-learning techniques, IEEE Internet of Things Journal 8 (5) (2020) 3242–3254.

[18] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, M. Guizani, Iot malicious traffic identification using wrapper-based feature selection mechanisms, Computers & Security 94 (2020) 101863.

[19] E. Raff, C. Nicholas, An alternative to ncd for large sequences, lempel-ziv jaccard distance, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 1007–1015.

[20] M. Shafiq, Z. Tian, Y. Sun, X. Du, M. Guizani, Selection of effective machine learning algorithm and bot-iot attacks traffic identification for internet of things in smart city, Future Generation Computer Systems 107 (2020) 433–442.

[21] A. Mohaisen, O. Alrawi, M. Mohaisen, Amal: High-fidelity, behavior-based automated malware analysis and classification, computers & security 52 (2015) 251–266.

[22] M. Polino, A. Scorti, F. Maggi, S. Zanero, Jackdaw: Towards automatic reverse engineering of large datasets of binaries, in: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2015, pp. 121–143.

[23] A. Tamersoy, K. Roundy, D. H. Chau, Guilt by association: large scale malware detection by mining file-relation graphs, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 1524–1533.

[24] L. Chen, T. Li, M. Abdulhayoglu, Y. Ye, Intelligent malware detection based on file relation graphs, in: Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015), IEEE, 2015, pp. 85–92.

[25] A. Abusitta, M. Q. Li, B. C. Fung, Malware classification and composition analysis: A survey of recent developments, Journal of Information Security and Applications 59 (2021) 102828.

[26] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in iot, Sensors 17 (9) (2017) 1967.

[27] A. A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for internet of things, Future Generation Computer Systems 82 (2018) 761–768.

[28] N. Moustafa, B. Turnbull, K.-K. R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, IEEE Internet of Things Journal 6 (3) (2018) 4815–4830.

[29] L. Aversano, M. L. Bernardi, M. Cimitile, R. Pecori, L. Veltri, Effective anomaly detection using deep learning in iot systems, Wireless Communications and Mobile Computing 2021.

[30] S. K. Sarma, Optimally configured deep convolutional neural network for attack detection in internet of things: impact of algorithm of the innovative gunner, Wireless Personal Communications 118 (1) (2021) 239–260.

[31] J. Saxe, K. Berlin, Deep neural network based malware detection using two dimensional binary program features, in: Malicious and Unwanted Software (MALWARE), 2015 10th International Conference on, IEEE, 2015, pp. 11–20.

[32] A. F. Agarap, Deep learning using rectified linear units (relu), arXiv preprint arXiv:1803.08375.

[33] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Icml, 2010.

[34] G. E. Dahl, J. W. Stokes, L. Deng, D. Yu, Large-scale malware classification using random projections and neural networks, in: Acoustics, Speech and Signal

Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 3422–3426.

[35] W. Huang, J. W. Stokes, Mtnet: a multi-task neural network for dynamic malware classification, in: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer, 2016, pp. 399–418.

[36] B. Kolosnjaji, A. Zarras, G. Webster, C. Eckert, Deep learning for classification of malware system call sequences, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2016, pp. 137–149.

[37] I. Ullah, Q. H. Mahmoud, Design and development of rnn anomaly detection model for iot networks, IEEE Access 10 (2022) 62722–62750.

[38] X. Zhou, Y. Hu, J. Wu, W. Liang, J. Ma, Q. Jin, Distribution bias aware collaborative generative adversarial network for imbalanced deep learning in industrial iot, IEEE Transactions on Industrial Informatics.

[39] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (11) (2020) 139–144.

[40] R. Kale, Z. Lu, K. W. Fok, V. L. Thing, A hybrid deep learning anomaly detection framework for intrusion detection, in: 2022 IEEE 8th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), IEEE, 2022, pp. 137–142.

[41] A. Abusitta, M. Bellaiche, M. Dagenais, T. Halabi, A deep learning approach for proactive multi-cloud cooperative intrusion detection system, Future Generation Computer Systems 98 (2019) 308–318.

[42] A. Abusitta, O. A. Wahab, T. Halabi, Deep learning for proactive cooperative malware detection system, in: Edge Intelligence Workshop, Vol. 711, 2020, p. 7.

[43] A. Abusitta, T. Halabi, O. A. Wahab, Robust: Deep learning for malware detection under changing environments, in: AIofAI'21: 1st Workshop on Adverse Impacts and Collateral Effects of Artificial Intelligence Technologies, CEUR Workshop Proceedings, 2021, pp. 1–13.

[44] Q. Zhang, L. T. Yang, Z. Chen, P. Li, A survey on deep learning for big data, Information Fusion 42 (2018) 146–157.

[45] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, D.-R. Liou, Autoencoder for words, Neurocomputing 139 (2014) 84–96.

[46] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in neural information processing systems, 2007, pp. 153–160.

[47] D. M. Kline, V. L. Berardi, Revisiting squared-error and cross-entropy functions for training neural network classifiers, Neural Computing & Applications 14 (4) (2005) 310–318.

[48] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, science 313 (5786) (2006) 504–507.

[49] G. E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural computation 18 (7) (2006) 1527–1554.

[50] F. Aubet, M. Pahl, DS2OS traffic traces, `https://www.kaggle.com/datasets/francoisxa/ds2ostraffictraces` (2018).

[51] M.-O. Pahl, F.-X. Aubet, All eyes on you: Distributed multi-dimensional iot microservice anomaly detection, in: 2018 14th International Conference on Network and Service Management (CNSM), IEEE, 2018, pp. 72–80.

[52] B. Weinger, J. Kim, A. Sim, M. Nakashima, N. Moustafa, K. J. Wu, Enhancing iot anomaly detection performance for federated learning, Digital Communications and Networks.

[53] H. Taud, J. Mas, Multilayer perceptron (mlp), in: Geomatic approaches for modeling land change scenarios, Springer, 2018, pp. 451–455.

[54] H. Yi, S. Shiyu, D. Xiusheng, C. Zhigang, A study on deep neural networks framework, in: 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), IEEE, 2016, pp. 1519–1522.

[55] K. Demertzis, L. Iliadis, E. Pimenidis, P. Kikiras, Variational restricted boltzmann machines to automated anomaly detection, Neural Computing and Applications (2022) 1–14.

[56] A. Dawoud, S. Shahristani, C. Raun, Deep learning and software-defined networks: Towards secure iot architecture, Internet of Things 3 (2018) 82–89.

[57] H. Larochelle, M. Mandel, R. Pascanu, Y. Bengio, Learning algorithms for the classification restricted boltzmann machine, The Journal of Machine Learning Research 13 (1) (2012) 643–669.

[58] G. E. Hinton, Deep belief networks, Scholarpedia 4 (5) (2009) 5947.

[59] F. Aubet, M. Pahl, Ds2os traffic traces (2018).

[60] A. Alhowaide, I. Alsmadi, J. Tang, Towards the design of real-time autonomous iot nids, Cluster Computing (2021) 1–14.

[61] A. Alhowaide, I. Alsmadi, J. Tang, Features quality impact on cyber physical security systems, in: 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, 2019, pp. 0332–0339.