A Cost-Sensitive Machine Learning Model With Multitask Learning for Intrusion Detection in IoT

Telikani, Akbar; Rudbardeh, Nima Esmi; Soleymanpour, Shiva; Shahbahrami, Asadollah; Shen, Jun; Gaydadjiev, Georgi; Hassanpour, Reza

DOI
10.1109/TII.2023.3314208

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Industrial Informatics

Citation (APA)
Telikani, A., Rudbardeh, N. E., Soleymanpour, S., Shahbahrami, A., Shen, J., Gaydadjiev, G., & Hassanpour, R. (2023). A Cost-Sensitive Machine Learning Model With Multitask Learning for Intrusion Detection in IoT. *IEEE Transactions on Industrial Informatics*, *20*(3), 3880-3890. https://doi.org/10.1109/TII.2023.3314208

A Cost-Sensitive Machine Learning Model With Multitask Learning for Intrusion Detection in IoT

Akbar Telikani, *Graduate Student Member, IEEE*, Nima Esmi Rudbardeh, *Graduate Student Member, IEEE*, Shiva Soleymanpour, Asadollah Shahbahrami, Jun Shen, *Senior Member, IEEE*, Georgi Gaydadjiev, and Reza Hassanpour

***Abstract*—A problem with machine learning (ML) techniques for detecting intrusions in the Internet of Things (IoT) is that they are ineffective in the detection of low-frequency intrusions. In addition, as ML models are trained using specific attack categories, they cannot recognize unknown attacks. This article integrates strategies of cost-sensitive learning and multitask learning into a hybrid ML model to address these two challenges. The hybrid model consists of an autoencoder for feature extraction and a support vector machine (SVM) for detecting intrusions. In the cost-sensitive learning phase for the class imbalance problem, the hinge loss layer is enhanced to make a classifier strong against low-distributed intrusions. Moreover, to detect unknown attacks, we formulate the SVM as a multitask problem. Experiments on the UNSW-NB15 and BoT-IoT datasets demonstrate the superiority of our model in terms of recall, precision, and F1-score averagely 92.2%, 96.2%, and 94.3%, respectively, over other approaches.**

***Index Terms*—Deep learning (DL), Internet of things (IoT), intrusion detection, multitask learning, support vector machine (SVM).**

NOMENCLATURE

AE	Autoencoder.
CNN	Convolutional neural networks.
CSHL	Cost-sensitive hinge loss.
DBN	Deep belief network.
DFR	Deep-full-range.
DL	Deep learning.
DNN	Deep neural network.
FN	False negative.
FNN	Feedforward neural network.
FP	False positive.
GAN	Generative adversarial network.
IDS	Intrusion detection system.
IoT	Internet of Things.
LSTM	Long short-term memory.
ML	Machine learning.
ReLU	Rectified linear unit.
RNN	Recurrent neural network.
SAE	Stacked autoencoder.
SMOTE	Synthetic minority oversampling technique.
SNN	Self-normalizing neural network.
SVM	Support vector machine.
x	Value in the dataset.
x_{norm}	Normalized value.
x_{min}	Minimum value for a variable.
x_{max}	Maximum value for a variable.
α	Distribution of a class.
γ	Cost matrix.
C	Number of classes in the dataset.
y_{train}	Set of target label.
X	Input vector in autoencoder.
k	Number of samples in a batch.
h	Number of layers in autoencoder.
$\vec{X}$	Reconstructed vector in autoencoder.
Y	Vector in the encoding layer of autoencoder.
Z	Vector in the decoding layer of autoencoder.
W	Weights in the encoding layer of autoencoder.
$\vec{W}$	Weights in the decoding layer of autoencoder.
b	Bias in the encoding layer of autoencoder.
$\vec{b}$	Bias in the decoding layer of autoencoder.
$L(.)$	Loss function in autoencoder.
$K(.)$	Radial basis function in SVM.
m	Number of features in a dataset.
$\varnothing\,(.)$	Nonlinear mapping function.
$\vec{b}$	Bias value in SVM.
$\vec{w}$	Weight vector in SVM.
$\vec{X}$	Feature vector in SVM.

Manuscript received 10 January 2023; revised 16 April 2023 and 2 July 2023; accepted 31 August 2023. Date of publication 2 October 2023; date of current version 23 February 2024. Paper no. TII-23-0105. *(Corresponding author: Akbar Telikani.)*

Akbar Telikani and Jun Shen are with the School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: at952@uowmail.edu.au; jshen@uow.edu.au).

Shiva Soleymanpour and Asadollah Shahbahrami are with the Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht 4199613776, Iran (e-mail: Soleymanpour.sh@gmail.com; shahbahrami@guilan.ac.ir).

Nima Esmi Rudbardeh and Georgi Gaydadjiev are with the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Faculty of Science and Engineering, University of Groningen, 9712 CP Groningen, The Netherlands (e-mail: nimaesmi.rudbardeh.2014@ieee.org; g.gaydadjiev@rug.nl).

Reza Hassanpour is with the Department of Computer Engineering, Gidatarim University Konya-Turkey, 42080 Konya, Turkey, and also with the Department of Computer Science, Rotterdam University, 3000 DR Rotterdam, The Netherlands (e-mail: zarer@hr.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3314208.

Digital Object Identifier 10.1109/TII.2023.3314208

$\vec{\xi}$	Slack variable.
η	Regularization parameter.
ρ	Penalty factor.
$D(\vec{X})$	Decision for $\vec{X}$.
sign(.)	Sign function.
$\mathcal{L}$	Loss value.
$\bar{d}$	Predicted output.
d	Desired output.
O	Output layer.
∂	Partial derivative.
$l(\cdot)$	Misclassification error.
exp(.)	Exponential function.
λ	Control parameter for new features in SVM.
n	Number of data points.
q	Scalar quantity in SVM.
n	Number of data points in the dataset.
$f(.)$	Decision function for two classes.

I. Introduction

INTRUSIONS are indeed a significant concern in the IoT environments, which are carried out for different reasons, such as obtaining private or sensitive data, disrupting data integrity, and damaging IoT devices and services [1]. Recognizing intrusion attacks is a critical component of a secure IoT network against cyberattacks . The components of an IDS in the IoT can be placed in the hierarchical architecture so that the analyzed data are moved up through the layers.

In recent years, studies on intrusion detection in the IoT have increasingly turned to ML and DL techniques and have been developed in different applications, such as traffic management [2] and autonomous vehicles [3]. However, IoT-related datasets are imbalanced in nature, in which some classes include far more instances (i.e., majority classes) compared to other classes (i.e., minority classes). In this situation, intrusion classifiers are biased toward the high-frequency classes so that low-frequency classes are wrongly detected as the high-frequency classes. These misclassifications can be harmful in critical IoT-related applications because the wrongful defense mechanisms lead to the waste of time, effort, and resources. For example, the occurrence of equipment failures is relatively rare compared to normal operations in manufacturing plants. If the system wrongly detects this low-frequency class as normal or benign, it may overlook the actual machine failure, leading to prolonged operation in a faulty state. As a result, the malfunctioning machine could cause further damage, production delays, and safety hazards or even compromise the integrity of the entire manufacturing process [4]. On the other hand, attackers can exploit vulnerabilities that have not yet been discovered or patched in security-critical applications to develop and deploy unknown attacks, posing a significant threat to privacy and security. ML models are built on datasets including only specific attacks, leading to the failure of the model in diagnosing new and unknown attacks.

Most papers on intrusion detection for the IoT used deep neural network models without the consideration of the class imbalance problem and unknown attack detection. Some recent studies have focused on handling these two problems separately. However, no paper has ever touched on both the issues of unknown attack detection and class imbalance problems simultaneously.

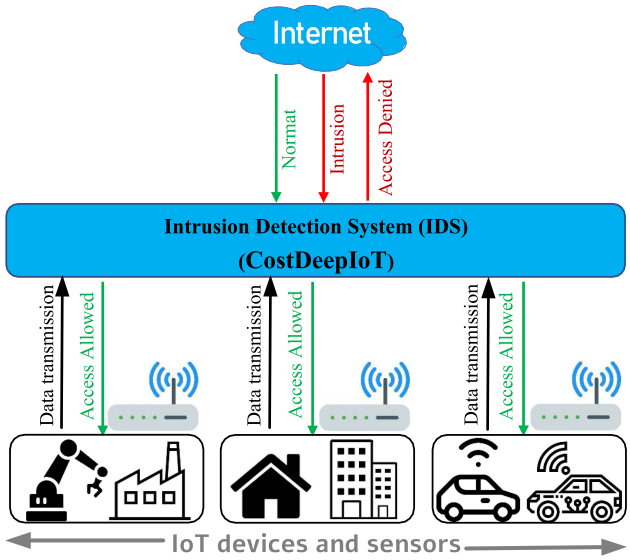


Fig. 1. Block diagram of the application of the proposed method in IoT network intrusion detection.

Our main motivation with this article is to develop an intrusion detection framework, namely, *CostDeepIoT*, for IoT security to address both the class imbalance problem and the unknown attack detection simultaneously using cost-sensitive learning and multitask learning, respectively. Our main contributions, for such a motivation, are as follows.

1) We develop a framework consisting of three main components: a) a hybrid ML model; b) cost-sensitive learning; and c) multitask learning. The hybrid ML model integrates an SVM classifier into a stacked autoencoder (SAE) network. In this model, the SAE extracts essential features from raw network data and feeds them into the SVM-based shallow classifier.
2) A multitask learning schema is designed by formulating a multiclass problem as multiple independent tasks to mine the shared information among tasks. Using multitask learning, features are learnt concurrently for all the tasks. For each task, the hinge loss function, which is a loss function for the SVM, is improved through the cost-sensitive learning strategy, in which costs related to misclassifications are considered when computing the loss value. With these two strategies, we can fully exploit their advantages to significantly improve the detection accuracy of both the low-frequency and unknown attacks.
3) We perform a comparative analysis to evaluate the effectiveness of our proposed framework in comparison to the existing state-of-the-art DL methods. These evaluations are conducted on two datasets, namely, UNSW-NB15 and BoT-IoT, using various metrics to assess performance and effectiveness.

Fig. 1 depicts how our proposed model can contribute to securing IoT networks against intrusion attacks. The model is

employed at the gateway of IoT different applications, where input data traffic is monitored. This can optimize bandwidth, latency, and response time. Network requests are intercepted and processed and fed into the proposed model to detect anomalies and reject threatening requests. Based on the corresponding information, a server offers guidelines to IoT devices for enhancing service quality and denies suspicious access from the Internet. Our proposed approach can be employed in all the classification-related IoT applications, such as smart fire detection, homes, transportation, and health, with the class imbalance problem. In addition, our framework is suitable for security-critical applications, where the probability of generating new attacks is high [5].

The rest of this article is organized as follows. Section II presents related studies on IDSs for the IoT. Section III introduces our framework for addressing two problems of class imbalance and unknown intrusions detection in the IoT using cost-sensitive learning and multitask learning. Section IV compares and evaluates the proposed framework with a series of comprehensive experiments. Finally, Section V concludes this article. In addition, a list of abbreviations and notations is depicted in the Nomenclature.

II. Related Work

A considerable number of papers have subjected intrusion detection for the IoT using DL techniques. We have classified these studies into three categories: 1) papers that provide generic DL models for intrusion detection (see Section II-A); 2) papers that consider the class imbalance problem (see Section II-B); and 3) papers that have focused on unknown attack detection (see Section II-C).

A. General Solutions for Intrusion Detection in the IoT

Intrusion detection is often performed using supervised ML algorithms [6], [7]. Particularly, DL techniques are broadly employed to effectively resolve the intrusion detection problem in IoT platforms, such as AEs [8], [9], FNN [10], DBN [11], and dense random neural network [12]. Alkadi et al. [13] employed a bidirectional LSTM in a deep blockchain framework to securely exchange and migrate data between multicloud IoT services. A two-layer architecture for handling both the flow- and packet-based features was developed in [14]. A pipeline of classification models was trained with the packet-based training subset and the other with the flow-based training subset. Lu et al. [15] integrated a memory module into an AE model to store and locate the latent space feature representations of normal data. Li et al. [16] combined federated learning (FL) and fog/edge computing for distributed denial-of-service (DDoS) traffic detection. A distributed policy optimization model was trained to reduce the impact of resource constraints of IoT devices on the IDS. A privacy-preserving FL scheme on a decentralized platform was proposed in [17]. These models are trained on devices, and their learning is federated.

The integration of DL models to empower the IDS has been introduced by some publications. A framework combining a gated recurrent unit (GRU), a multihead self-attention mechanism (MHSA), and the feedforward layer was proposed in [18]. The GRU model extracts local representations from raw traffic and sends them to the MHSA for capturing long-term information to facilitate the distributed and paralleled execution of the model. The feedforward layer performs a nonlinear feature transformation.

An appropriate adjustment of hyperparameters for the DL algorithm can effectively affect the performance of the IDS. Evolutionary techniques can be employed to improve detection models for intrusion attacks in IoT environments. The genetic algorithm for the DBN [19], the particle swarm optimization for the CNN [20], and spider monkey optimization for the stacked deep polynomial network [21] are some of these efforts in the literature. Chen et al. [22] modified a multiobjective evolutionary algorithm to perform the parameter tuning process of the CNN. Their model was executed on fog nodes for detecting intrusions in the IoT. Detection performance and model complexity of the CNN model were considered as two conflicting objectives.

Owing to the imbalanced nature of data in IoT environments, classifiers are more biased toward the majority classes than minority classes. This results in DL models that have poor predictive performance for the minority class. Moreover, all the abovementioned studies in IoT intrusion detection expect classes captured by the classifier in the training phase are what the classifier handles in the testing step. This assumption cannot be applicable in real-world applications because adversaries and hackers try to implement novel attacks differently from traditional intrusions to fool an intelligent IDS [23]. Therefore, an unknown attack is classified as one of the classes the model knows. A few papers have focused on the class imbalance problem and unknown intrusion detection separately, which are described in the following subsections.

B. Solutions for the Class Imbalance Problem in Intrusion Detection in the IoT

Gupta et al. [24] employed random oversampling, borderline-SMOTE, and SVM-SMOTE techniques to address the class imbalance problem. A two-layer model was then employed that applied LSTM in the first layer to separate benign and malicious network traffic, and it used random forest and bagging ensembles in the second layer to classify attacks. Telikani and Gandomi [1] integrated the cost-sensitive learning strategy into an SAE model, where the class-dependent costs are determined using a heuristic. They then improved the cost adjustment through an evolutionary algorithm during the model training [25]. Moreover, they designed a fog-computing-enabled framework to accelerate the IDS for big IoT data. In [26], few-shot learning with variational feature representation was employed to handle the out-of-distribution problem in imbalanced data.

C. Solutions for Unknown Attack Detection in Intrusion Detection in the IoT

Lu et al. [15] integrated a memory module into an AE between encoder and decoder layers to store and locate the latent space feature representations of normal data, resulting in the detection of unknown attacks. D-Sign [27] is a DL-based system for the intrusion detection and signature generation of unknown web attacks. First, the incoming traffic is analyzed via a *Misuse*

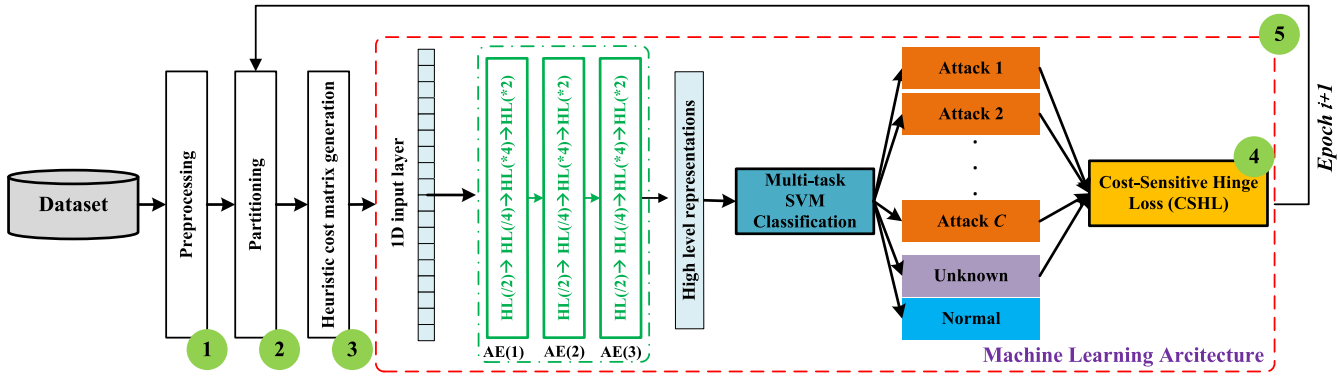


Fig. 2. Proposed framework based on cost-sensitive learning, multitask learning, and ML for intrusion detection in the IoT platform.

Detection Engine to detect known attacks. Then, an *Anomaly Detection Engine* processes the packet stream to recognize unknown attacks and sends them to a *Signature Generation Engine* to generate a signature for the malicious packet.

III. *CostDeepIoT:* An IDS in the IoT

ML models for IoT intrusion detection apply multiclass models for attack detection and can only learn a single task at a time. Unfortunately, traffic flow contains different types of attacks simultaneously. Owing to the particular features of each attack type, the identification of each attack can be considered as an independent task. Moreover, these models cannot handle the class imbalance problem and unknown attack detection simultaneously.

In this section, a novel ML-based framework to mitigate the impacts of the class imbalance problem and unknown attacks on the IDS in IoT environments is proposed. This framework is developed using three concepts: 1) ML techniques; 2) cost-sensitive learning strategy; and 3) multitask learning. Fig. 2 shows our developed framework including five main components: 1) preprocessing; 2) partitioning; 3) heuristic cost matrix generation; 4) ML architecture; and 5) CSHL. First, the *preprocessing* step is employed to provide a clean dataset for our DL model by deleting irrelevant attributes, normalizing the data, and dividing the dataset into training and testing sets. Then, in the *partitioning* step, the training dataset is partitioned into different parts; after that, in the *heuristic cost matrix generation* phase, a cost matrix is generated for each partition using a heuristic; in the *ML architecture* phase, by stacking an SAE and a multitask SVM, a hybrid ML model is built, in which the SAE reconstructs raw data and provides new representations with which to perform classification tasks. In this model, the multitask SVM is used for detecting unknown attacks. Finally, to calculate the weighted loss value, a *cost-sensitive hinge loss function* takes into account the cost values obtained in the third step. In this way, an ML model becomes sensitive to low-frequency attacks.

A. Preprocessing

To preprocess the data, irrelevant features, such as source/destination IP and source/destination port number, are first removed from the dataset. Then, the traffic features are converted into numeric features using a data conversion technique. This is because DL models require features in numerical values. In the datasets selected in this study, there are some categorical features, such as "proto," "state," and "service," which are changed to numerical features using the ordinal encoding method. Convergence speed and model learning are negatively influenced by the large variations in data values. Therefore, the min–max technique is used to normalize the dataset and convert all the values between 0 and 1 using the following equation:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{1}$$

In this equation, the terms x_{norm} and x refer to the normalized and original values, respectively. In addition, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of an attribute, respectively.

B. Partitioning

Defining appropriate class-dependent costs positively influences the performance of an ML model, particularly in training it to be sensitive to minority classes. A single predefined cost matrix is often used in cost-sensitive learning. In this study, we introduce a diversity concept in determining class-dependent costs. We accomplish this by designing a dynamic partitioning strategy that generates diverse cost matrices in each epoch of the training process. This strategy allows the learning of our model with different class imbalance conditions. In this strategy, at each epoch, the dataset is first split into a number of equal parts, and then, a cost matrix is created for each partition. Instances in each partition are selected randomly to provide dynamic data distribution of each class. The cost matrix associated with each partition is used for the DL model training.

C. Heuristic Cost Matrix Generation

After the partitioning step, a heuristic mechanism is employed to acquire a class-dependent cost matrix (notated by γ) from each partition, based on the distribution of classes in that partition. In a cost matrix, each cell refers to the probability of misclassifying a class to another class. This probability value is specified according to the class distribution. The cost of class i in class j,

Algorithm 1: Cost Matrix Generation.

Input: y_{train}, C
Output: cost-matrix γ
1: **Begin**
2: $\gamma \leftarrow Initialize\ with\ zeros$
3: $\alpha \leftarrow Compute\ frequency\ of\ classes$
4: **For each** $i \in labels$
5: **For each** $j \in labels$
6: **if** $i \neq j$
7: $\gamma_{i,j} = P(j|X) = 1 - \frac{\alpha_i}{\alpha_j}$

notated as $\gamma_{i,j}$, is computed as follows:

$$\begin{cases} \gamma_{i,j} = P(j|X) = 1 - \frac{\alpha_i}{\alpha_j}, \quad i,j = 1,2,\ldots,C \\ \\ \text{subject to } i \neq j \text{ and } \alpha_i, \alpha_j > 0 \end{cases}. \tag{2}$$

The terms α_i and α_j are the distribution of classes i and j, respectively, and $P(j|X)$ is the probability over a class given a feature vector X. If there is no sample for a class i in a partition (i.e., α_i), the costs related to the class are determined to be zero. Indeed, all the cells in the corresponding row and column are set to be zero. Algorithm 1 presents the pseudocode of our cost matrix generation algorithm.

D. Hybrid ML Architecture

In an unsupervised learning setting, AEs are capable of learning compact representations [28]. Furthermore, recent studies have shown that self-supervised learning is more resilient to imbalances in datasets [29]. In our framework, we integrate a multitask SVM classifier into an SAE. In the first phase, a stack of three AEs with a 1-D input layer is designed to extract high-level representations from the raw data (see Fig. 3). In the second step, the features are sent to a multitask SVM model to detect intrusion classes. Owing to the higher generalization capability of SVM than that of SAE, this integration can minimize generalization errors with global optimization.

1) Designing the SAE Model for Feature Learning: We design an SAE model with three AE models, and each of them has two encoder and decoder layers. In this structure, at the encoding stage, the output of the $(h-1)$th layer is used for learning kth-order features in the hth hidden layer of the SAE. In this way, raw input is considered for learning first-order features in the first hidden layer. These features are used in the second hidden layer for learning second-order features. Conversely, at the decoding stage, the output of the kth layer is utilized to reconstruct $(h-1)$th-order features in the $(h-1)$th layer. This process continues until the input is reconstructed.

An AE tries to minimize the difference between the original input (i.e., X) and the reconstructed input (i.e., $\bar{X}$) using the following equation as the mean squared error:

$$L(X, \bar{X}) = \frac{1}{k} \sum_{i=1}^{k} \left\| X_i - \bar{X}_i \right\|_2^2. \tag{3}$$

where k is the number of samples in a batch.

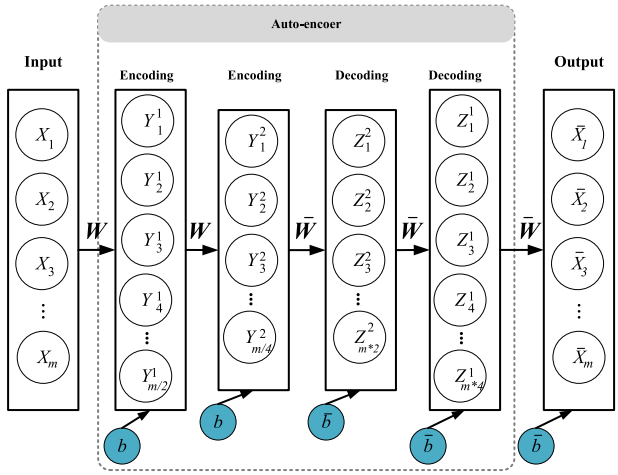


Fig. 3. Structure of AEs in the hybrid model.

2) Multitask SVM Classification: An SVM model separates data classes by generating a hyperplane from a high-dimensional feature space mapped from training examples. Radial basis function is commonly used as a kernel function for the mapping between two objectives x_i and x_j, as follows:

$$K(x_i, x_j) = \exp\left(-\frac{\|(x_i, y_j)\|^2}{2\lambda^2}\right) \tag{4}$$

where λ is a parameter to control the influence of new features on the decision boundary. The main motivation is to maximize the margin between the data classes. For this reason, an optimization problem is formulated in an SVM to construct a maximal margin classifier

$$\begin{cases} \text{minimize} & \frac{1}{2}\|\vec{w}\|^2 + \eta \sum_{i=1}^{n} \zeta_i \\ \text{subject to} & y_i(\vec{w}_i x_i + q) > 1 - \zeta_i \end{cases} \tag{5}$$

where η is the regularization constant (i.e., slack factor) for classification errors, and ζ_i represents the slack variable associated with the ith sample. The terms n, $\vec{w}$, and q are the number of data points, weight vector, and a scalar quantity, respectively.

SVM models are used to solve many real-world industrial problems, such as fault detection in the production line, car/road detection in autonomous vehicles, and intrusion detection in industrial control systems [30]. In our problem, we employ the SVM for detecting IoT-related intrusions. We are dealing with a multiclass intrusion detection task, where multiple types of attacks may occur simultaneously within the traffic flow. However, a learning model can only handle a single task at a time. This limitation arises from the fact that each attack type possesses unique features, making the identification of each attack an independent task in itself.

As an SVM is by itself a two-class classifier, a multiclass SVM is implemented by combining multiple two-class SVM. *One against rest* and *one against one* are two common approaches to build a multiclassifier. The first one generates C number of SVM models from C classes. The second approach constructs one SVM for each pair of classes, resulting in $C(C-1)/2$ SVM models. It has been proven that the one-against-one approach is

substantially fast to train and is preferable for problems with a large number of classes [31]. Thus, we use the one-against-one method to detect different types of intrusions in IoT platforms. The decision function for two classes i and j for a feature vector X ($f_{ij}(X)$) is formulated as follows:

$$f_{ij}(\vec{X}) = \left\langle \varnothing(\vec{X}).\vec{w}_{ij} \right\rangle .\vec{b}_{ij} \tag{6}$$

where $\varnothing(\vec{X})$ is a nonlinear function for mapping $\vec{X}$ into a high-dimensional space feature space, m is the number of features in a dataset, and $\vec{b}$ is a constant bias value.

We use $L1$ regularization as the optimization problem for the decision function is formulated as follows:

$$\min \frac{1}{2}\left(\vec{w}_{ij}\right)^2 + \rho \sum_n \left(\vec{\xi}_{ij}\right)_n \tag{7}$$

$$\begin{cases} \left\langle \varnothing(\vec{X}).\vec{w}_{ij} \right\rangle .\vec{b}_{ij} \geq 1 - \vec{\xi}_{ij}, & \bar{d} = i \\ \left\langle \varnothing(\vec{X}).\vec{w}_{ij} \right\rangle .\vec{b}_{ij} \leq -1 + \vec{\xi}_{ij}, & \bar{d} = j \end{cases} \tag{8}$$

where the parameters of $\vec{w}_{ij}$, $\vec{b}_{ij}$, and $\vec{\xi}_{ij}$ are optimized in the SVM classifier, and (8) is the constraint. $\vec{\xi}_{ij}$ is a slack variable, ρ is a penalty factor, and $\bar{d}$ is the predicted class.

In our study, each binary SVM model is considered as a task, and the feature weights for each model (task) are learnt simultaneously by the relationship across all the tasks, and the specific type for attacks is predicted. The "Max-Wins" decision function is used to determine the output of all $C(C-1)/2$ SVM models [see (9)]. In "Max-Wins" decision, each class is fed to all the SVM models to produce different outputs. The final decision for $\vec{X}$ is determined based on a voting for all classes, and the class with the most votes is the final class

$$D\left(\vec{X}\right) = \text{argmax} \sum_{i \neq j, j=1}^{C} \text{sign}(f_{ij}(\vec{X})) \tag{9}$$

where sign(.) is a symbolic function and sign(.) $\in [-1, 1]$. If sign(.) $= -1$, samples are judged as negative samples; if sign(.) $= 1$, the samples are judged as positive samples.

In the case of multiple classes with the same number of votes, the closest class is determined via a real-valued decision function, as described in the following equation:

$$D\left(\vec{X}\right) = \text{argmax} \sum_{i \neq j, j=1}^{C} f_{ij}(\vec{X}). \tag{10}$$

E. Cost-Sensitive Hinge Loss

Using a cost-sensitive learning strategy, the hinge loss function is enhanced to update the parameters of the hybrid model based on class-specific costs in a way that the model prioritizes the correct classification of minority attacks. The costs are considered in the hinge loss function after the softmax layer. The hinge loss function is selected due to its faster convergence and better performance than that of other loss functions in most cases. In addition, it can avoid the learning slowing down, which is a typical problem of the mean squared error loss function [32]. By considering the costs assigned in the cost matrix generation phase, a punishment value is adjusted for each misclassification. Based on this objective, a maximum margin should be maintained between each pair of classes using the following equation:

$$\mathcal{L}\left(d, \bar{d}\right) = \max(0, 1 - d.\bar{d}) \tag{11}$$

where $\mathcal{L}$ is the loss value and d is the desired output. As a result of incorporating the corresponding class-dependent cost, the predicted class probability changes, as expressed in the following equation:

$$\bar{d} = \frac{1}{1 + \exp(-O_n \gamma_{d,\bar{d}})} \tag{12}$$

where $\gamma_{d,\bar{d}}$ is the class-sensitive penalty, which depends on the desired class of a particular training sample, and $-O_n$ is the output of ith neuron in the output layer.

At each neuron in the SAE model, the directional derivative can be computed using the following equation:

$$\frac{\partial l(O, \bar{d})}{\partial O_n} = -\left(2d_n - 1\right) \frac{\partial \bar{d}_n}{\partial O_n} f\left(1 > \bar{d}_n(2d_n - 1)\right) \tag{13}$$

where $l(\cdot)$ is the misclassification error. Based on $\partial \bar{d} / \partial O_n = \gamma_{d,\bar{d}}$, the partial derivative of the softmax output with respect to the penultimate layer output is obtained. As a result of combining the two expressions mentioned above, we can express the following derivative:

$$\frac{\partial l(O, \bar{d})}{\partial O_n} = -\left(2d_n - 1\right) \gamma_{d,\bar{d}} f\left(1 > \bar{d}_n(2d_n - 1)\right) \tag{14}$$

where $f(.)$ denotes an indicator function.

IV. Experiments and Discussion

This section compares the performance of *CostDeepIoT* in comparison with its baseline version, Deep-IFS [18], EvolCost-Deep [25], and D-Sign [27] on the UNSW-NB15 and BoT-IoT datasets. The Baseline is a version of the *CostDeepIoT* without both the cost-sensitive learning and the task for the unknown attack detection task.

Our model has been implemented using Python and YAFS [33]. The machine used for all the experiments has an Intel CPU with 32 cores at 2.2 GHz and 13-GB RAM on CentOS 7 Linux. The environment development is based on Python-3.9.7 and Tensorflow-2.9.1. We have carried out diverse experiments by setting the number of epochs and partitions to 100 and 8 (based on our analysis shown in the next section), respectively. The batch size and dropout ratio are chosen as 512 and 0.05, respectively. Besides, ReLU was used as an activation function in the SAE model. In order to eliminate the overfitting problem, we used the early stopping technique with $patience = 3$. In other words, the training process ends after three epochs of no improvement in the validation loss value. The optimizer algorithm was the Adam, and the loss function was our improved hinge loss (i.e., CSHL). To adjust SVM parameters, i.e., λ and η, we have employed the grid search technique on the training data. The results showed that the best parameter combination is $\lambda = 0.3$ and $\eta = 1.0$. We have considered the training, validation, and testing sets to have 80%, 10%, and 10% ratios, respectively.

TABLE I
STATISTICS OF UNSW-NB15 AND IOT-BOT DATASETS

Dataset	Class type	|Samples|	Ratio
UNSW-NB15	Normal	1 958 467	0.879
	Exploits	33 422	0.015
	DoS	11 716	0.005
	Backdoor	1959	0.0008
	Analysis	2069	0.0009
	Fuzzers	19 578	0.00879
	Generic	187 598	0.084
	Recon.	10 871	0.004
	Shellcode	1187	0.0005
	Worms	134	0.00006
	Total	**2 227 000**	
BoT-IoT	DoS	33 005 194	0.341894
	DDoS	38 532 480	0.39915
	Recon.	1 821 639	0.01887
	Theft	1587	0.000016
	Normal	23 175 520	0.24007
	Total	**96 536 420**	

The bold values represent the sum of frequency of classes.

A. Dataset Characteristics

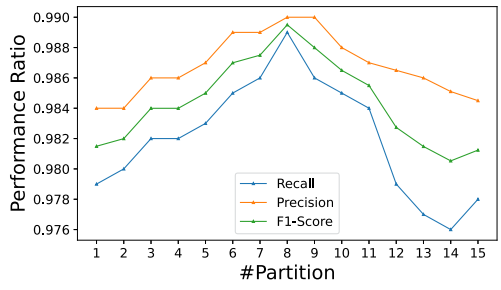
In this study, two imbalanced datasets of UNSW-NB15 [34] and BoT-IoT [35], which are widely used for the evaluation of IoT-related intrusion systems, are utilized. Moreover, evaluations of these two datasets can faithfully reflect the robustness of the models to different level data imbalance. The UNSW-NB15 dataset has been collected during the period of January and February 2015. Both the attack and normal traffic against servers were created using an automatic attack generation tool (IXIA PerfectStorm). This dataset has nine types of attacks, where the "Normal" is the majority class with around 88%. Other attack classes have less distributions. The BoT-IoT dataset is a collection of traffic data related to the simulation of five IoT devices considering the traffic of various attacks. The dataset comprises five categories, three classes of "DoS," "DDoS," and "Normal" have high distribution. On the other hand, the classes "Recon." and "Theft" represent attack categories with a minor frequency. These two datasets are compared statistically in Table I.

As one of our framework's objectives is to detect unknown attacks, we have considered "Exploits" in the UNSW-NB15 dataset and "DDoS" in the BoT-IoT dataset as unknown attacks and removed them from the training set.

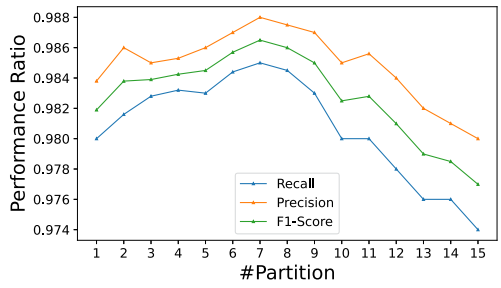
B. Experimental Results

This section evaluates the results obtained from the intrusion detection methods in terms of three measures of recall, precision, and F1-score. Defining an appropriate number of partitions effectively influences the performance of the *CostDeepIoT*. Fig. 4 shows how the performance is influenced by varying the number of partitions. The results demonstrated that a low number of partitions cannot train a model to be strong against imbalanced data, while a high number of partitions result in categories with zero samples in some partitions that make the model unreliable for the detection of these attack categories. According to these results, we have set the number of partitions in our experiments to eight for both the UNSW-NB15 and BoT-IoT datasets.

Table II presents the confusion matrix obtained from our *CostDeepIoT* model on the UNSW-NB15 dataset. The results



(a)



(b)

Fig. 4. Performance of *CostDeepIoT* with different partitions. (a) UNSW-NB15. (b) BoT-IoT.

demonstrate that the "Normal," which is the majority class, noticeably influences the classification accuracy of the low-frequency intrusions (e.g., "Worms" and "Backdoor"), so that the highest misclassification ratio is for these intrusion classes, 15.3% and 30.7% for "Backdoor" and "Worms," respectively. The average detection error of our proposed method for the UNSW-NB15 dataset was 7%.

Table III compares the performance of the models compared with the other methods on the UNSW-NB15 dataset. Cost-sensitive learning resulted in a higher recall ratio for our model compared with the other methods. The results have demonstrated that the *CostDeepIoT* performed better than the D-Sign [27], EvolCostDeep [25], and Deep-IFS [18], with a recall ratio of 87.4% against 81.2%, 71.9%, and 75.1%, respectively. The reason for this tendency is that our model using cost-sensitive learning reduces the number of false negatives for minority data. Moreover, the ability of our model in detecting unknown attacks results in higher performance compared to the other methods that cannot recognize these attacks. An average precision ratio of 97.5% was obtained for our proposed model in comparison with the baseline, EvolCostDeep [25], and Deep-IFS [18], which are 85.2%, 85%, and 85.6%, respectively. According to the F1-score criterion, our model achieved the highest F1-score compared with the other models, with an average of 92.8%. Meanwhile, this value is 75.5%, 77.3%, and 79.6% for baseline, EvolCostDeep [25], and Deep-IFS [18], respectively. Except of our model, the other classifiers' performance for unknown attacks was zero because they could not identify unknown attacks, and these attacks were wrongly classified as other classes. Overall, the developed model could increase recall, precision, and F1-score by 14.2%, 12.2%, and 15.3%, respectively.

To have a better illustration of our results on the UNSW-NB15 dataset, Fig. 5 depicts the error bar plot of the methods in terms

TABLE II
CONFUSION MATRIX OF THE *COSTDEEPIOT* FOR THE UNSW-NB15 DATASET

	Normal	Unknown	DoS	Backdoor	Analysis	Fuzzers	Generic	Recon.	Shellcode	Worms	Error	Rec.
Normal	193,901	153	15	0	0	235	1,554	11	1	1	0.01	0.989
Unknown	307	3,007	7	0	0	6	15	0	0	0	0.1	0.899
DoS	141	3	1,018	0	0	0	9	0	0	0	0.13	0.869
Backdoor	21	1	2	165	0	1	5	0	0	0	0.153	0.846
Analysis	20	2	2	0	175	2	5	0	0	0	0.15	0.849
Fuzzers	231	3	2	0	0	1,713	8	0	0	0	0.124	0.875
Generic	1,018	8	3	0	0	1	17,727	2	0	0	0.055	0.944
Recon	82	2	2	0	0	0	6	995	0	0	0.084	0.915
Shellcode	15	0	0	0	0	0	0	0	97	0	0.133	0.866
Worms	4	0	0	0	0	0	0	0	0	9	0.307	0.692
Error	0.009	0.057	0.03	0	0	0	0.082	0	0	0	**AE**: 0.071	**AR**: 0.87
Pre.	0.99	0.945	0.968	1.0	1.0	0.874	0.917	0.987	0.989	0.9	**AP**: 0.957	

AE: average error; AP: average precision; AR: average recall; Pre.: precision; and Rec.: recall.

TABLE III
COMPARISON BETWEEN THE PERFORMANCE OF INTRUSION DETECTION METHODS ON THE UNSW-NB15 DATASET

	CostDeepIoT			**Baseline**			**DEEP-IFS [18]**			**EvolCostDeep [25]**			**D-Sign [27]**		
Class	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1
Normal	0.989	0.99	0.99	0.982	0.975	0.978	0.986	0.978	0.981	0.987	0.98	0.983	0.982	0.976	0.978
Unknown	0.899	0.976	0.935	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.847	0.952	0.896
DoS	0.869	0.969	0.916	0.806	0.932	0.864	0.784	0.911	0.842	0.836	0.936	0.883	0.829	0.971	0.894
Backdoor	0.846	1.0	0.916	0.762	0.982	0.858	0.754	0.988	0.855	0.812	0.988	0.891	0.807	0.974	0.882
Analysis	0.849	1.0	0.918	0.77	0.972	0.859	0.773	0.983	0.865	0.817	0.988	0.894	0.816	0.976	0.887
Fuzzers	0.875	0.925	0.899	0.832	0.829	0.83	0.812	0.817	0.814	0.823	0.833	0.827	0.818	0.836	0.826
Generic	0.944	0.947	0.945	0.934	0.911	0.922	0.925	0.906	0.915	0.914	0.915	0.914	0.906	0.921	0.913
Recon.	0.915	0.987	0.949	0.854	0.954	0.901	0.843	0.943	0.890	0.877	0.963	0.918	0.866	0.957	0.909
Shellcode	0.866	1.0	0.928	0.78	0.97	0.864	0.771	0.979	0.862	0.792	0.96	0.867	0.635	0.951	0.761
Worms	0.692	1.0	0.817	0.538	1.0	0.7	0.551	1.0	0.711	0.654	1.0	0.791	0.614	1.0	0.76
Average	**0.874**	**0.979**	**0.921**	**0.725**	**0.852**	**0.755**	**0.719**	**0.85**	**0.773**	**0.751**	**0.856**	**0.796**	**0.812**	**0.951**	**0.871**

Pre.: precision; Rec.: recall; and F1: F1-score.
The bold values represent the average values of all measurements for different classes.

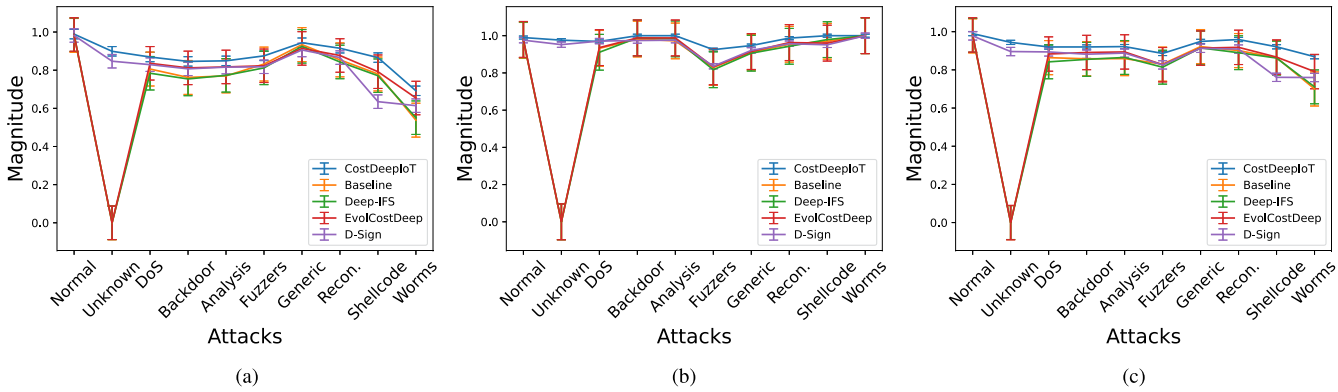


Fig. 5. Performance difference between the values of three measures of (a) recall, (b) precision, and (c) F1-score obtained from different methods on the UNSW-NB15 dataset.

of three performance criteria of recall [see Fig. 5(a)], precision [see Fig. 5(b)], and F1-score [see Fig. 5(c)]. This figure shows how much the performance of the models for each class is closer to their variance. The lower variance demonstrates a better performance for a model than other models. Based on the results, it is obvious that the *CostDeepIoT* model outperformed the other three methods and had lower variances. This is more outstanding for recall and F1-score measures that are better indicators to show the superiority of a classification model on the imbalanced data and the existence of unknown attacks. The figure explains that because of its capability in handling unknown and low-frequency attack detection, the lowest error variance and the highest performance have been obtained for our model.

The confusion matrix for the *CostDeepIoT* on the BoT-IoT is shown in Table IV. The table demonstrates that the highest error ratio is for low-frequency attacks. For instance, "Theft" attack, which has the lowest data distribution, led to the highest classification error (13.8%) for our model. In contrast, there is a low proportion of detection errors attributed to the "Unknown" attack (0.16%), which is highly distributed. On the BoT-IoT dataset, Table V presents the results for the *CostDeepIoT* in comparison with the other intrusion classifiers. Based on

TABLE IV
ANALYSIS OF THE CONFUSION MATRIX DERIVED FROM THE BOT-IOT DATASET USING THE PROPOSED MODEL

	DoS	Unknown	Recon.	Theft	Normal	Error	Rec.
DoS	323,450	3,039	736	3	2,824	0.02	0.979
Unknown	2,924	379,135	587	4	2,674	0.016	0.983
Recon.	268	215	17,551	0	182	0.036	0.963
Theft	8	7	1	137	6	0.138	0.861
Normal	3,245	2,974	2,641	2	222,663	0.038	0.961
Error	0.019	0.016	0.225	0.065	0.025	**AE**: 0.06	**AR**: 0.95
Pre.	0.98	0.983	0.815	0.938	0.975	**AP**: 0.938	

AE: average error; AP: average precision; AR: average recall; Rec.: recall; and Pre.: precision.

TABLE V
COMPARISON BETWEEN PERFORMANCE OF INTRUSION DETECTION METHODS ON THE BOT-IOT DATASET

	CostDeepIoT			Baseline			DEEP-IFS [18]			EvolCostDeep [25]			D-Sign [27]		
Class	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1	Rec.	Pre.	F1
DoS	0.979	0.98	0.979	0.967	0.97	0.968	0.965	0.972	0.968	0.97	0.976	0.972	0.973	0.957	0.964
Unknown	0.973	0.985	0.978	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.936	0.941	0.938
Recon.	0.963	0.815	0.882	0.916	0.796	0.851	0.882	0.804	0.841	0.915	0.792	0.849	0.903	0.782	0.838
Theft	0.861	0.938	0.898	0.751	0.884	0.812	0.724	0.892	0.799	0.776	0.913	0.838	0.732	0.911	0.811
Normal	0.961	0.975	0.967	0.952	0.964	0.957	0.948	0.957	0.952	0.955	0.96	0.957	0.934	0.952	0.942
Average	**0.947**	**0.938**	**0.94**	**0.717**	**0.722**	**0.717**	**0.703**	**0.725**	**0.712**	**0.723**	**0.728**	**0.723**	**0.895**	**0.908**	**0.902**

Pre.: precision; Rec.: recall; and F1: F1-score.
The bold values represent the average values of all measurements for different classes.

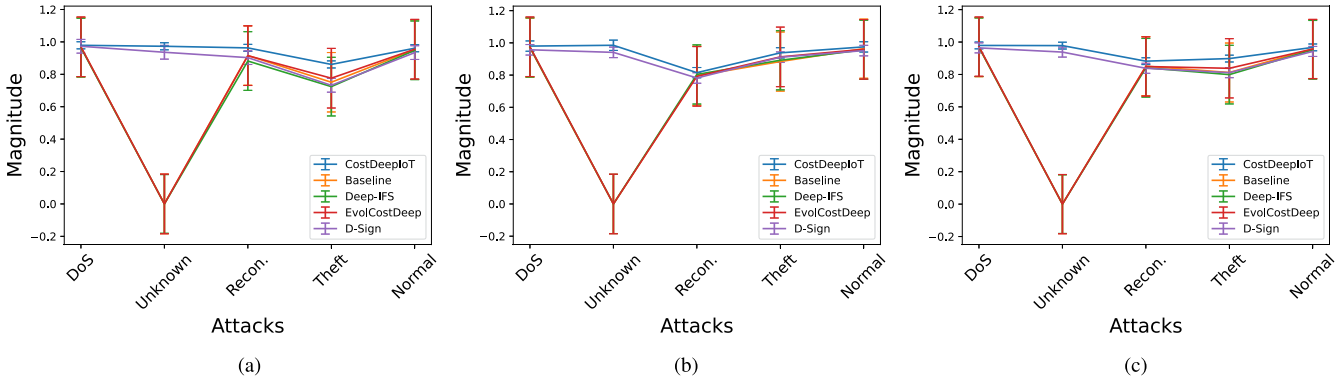


Fig. 6. Performance difference between the values of three measures of (a) recall, (b) precision, and (c) F1-score obtained from different methods on the BoT-IoT dataset.

our findings, the *CostDeepIoT* method has been observed that outperforms the other models, especially for the minority classes (e.g., "Recon." and "Theft"). For instance, the *CostDeepIoT* model could increase the classification ratio of "Theft" class around 11%, 13.8%, and 8.6% than that of the baseline model, EvolCostDeep [25], and Deep-IFS [18], respectively. In overall, the *CostDeepIoT* could obtain the highest detection ratio with recall, precision, and F1-score around 23.2%, 21.3%, and 22.2%, respectively, higher than that of the other methods.

Fig. 6 illustrates the error bar plot comparison of the performance between the *CostDeepIoT* and other DL methods in terms of different evaluation metrics on the BoT-IoT dataset. The experimental results showed that the *CostDeepIoT* outperformed the EvolCostDeep [25] and the Deep-IFS [18] for all three measures of recall, precision, and F1-score. The standard deviation of the results for the low-frequency attacks, such as "Recon." and "Theft," was smaller for the *CostDeepIoT* than the other three methods. Especially on F1-score, *CostDeepIoT* obtained a significant performance improvement over the others.

V. CONCLUSION

This article integrated multitask learning and cost-sensitive learning into a hybrid ML model for handling unknown attack detection and class imbalance problem in the IoT. The hybrid model included a combination of stacked AEs as a feature extractor and an SVM as an intrusion classifier. In this model, multiclass classification was formulated as multitask learning so that each binary SVM classifier was considered an independent subtask. Each task employed a weighted loss function, which was an enhancement of the hinge loss function. To have diverse costs and train the model with datasets with different characteristics, the costs were determined at each epoch using a formulated heuristic based on data statistics. The results on the UNSW-NB15 and BoT-IoT datasets proved that our model could attain high levels of performance, achieving (88.8%, 95.6%), (96.8%, 95.7%), and (92.8%, 95.6%) for recall, precision, and F1-score on the UNSW-NB15 dataset, respectively. This article can help different IoT-aided industrial applications, such as health care, industry, and transportation, to detect attacks

regarding obtaining financial gains, damaging equipment, and threatening human lives through the obstruction of normal industrial operations. In the future, we aim to adapt our framework for other applications with imbalanced nature, such as network traffic classification. Moreover, ensemble learning can be used by employing different DL models as the classifier. Owing to the difficulty of using our technique for real-time systems, we will focus on the decentralized implementation of the proposed system using FL in the edge/fog layer to offload the training and processing tasks from IoT devices in the future. The inability of fog computing in supporting mobility can be handled by using reinforcement learning techniques, such as Q-learning. Owing to the lack of application programming interface (API) authentication in edge devices, blockchain technology will be employed in the future.

References

[1] A. Telikani and A. H. Gandomi, "Cost-sensitive stacked auto-encoders for intrusion detection in the Internet of Things," *Internet Things*, vol. 14, 2021, Art. no. 100122.

[2] U. Ahmed, J. C.-W. Lin, G. Srivastava, U. Yun, and A. K. Singh, "Deep active learning intrusion detection and load balancing in software-defined vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 953–961, Jan. 2023.

[3] U. Ahmed, J. C.-W. Lin, and G. Srivastava, "A resource allocation deep active learning based on load balancer for network intrusion detection in SDN sensors," *Comput. Commun.*, vol. 184, no. 1, pp. 56–63, 2022.

[4] M. N. Syed, M. R. Hassan, I. Ahmad, M. M. Hassan, and V. H. C. De Albuquerque, "A novel linear classifier for class imbalance data arising in failure-prone air pressure systems," *IEEE Access*, vol. 9, pp. 4211–4222, 2020.

[5] A. Hussain et al., "Security framework for IoT based real-time health applications," *Electronics*, vol. 10, no. 6, pp. 719–734, 2021.

[6] A. Mishra, B. B. Gupta, D. Peraković, F. J. G. Peñalvo, and C.-H. Hsu, "Classification based machine learning for detection of DDoS attack in cloud computing," in *Proc. EEE Int. Conf. Consum. Electron.*, 2021, pp. 1–4.

[7] A. Mishra and N. Gupta, "Supervised machine learning algorithms based on classification for detection of distributed denial of service attacks in SDN-enabled cloud computing," in *Cyber Security, Privacy and Networking*. New York, NY, USA: Springer, 2022, pp. 165–174.

[8] K. Yang, Y. Shi, Z. Yu, Q. Yang, A. K. Sangaiah, and H. Zeng, "Stacked one-class broad learning system for intrusion detection in industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 251–260, Jan. 2023.

[9] A. Basati and M. M. Faghih, "PDAE: Efficient network intrusion detection in IoT using parallel deep auto-encoders," *Inf. Sci.*, vol. 598, pp. 57–74, 2022.

[10] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Comput. Netw.*, vol. 186, pp. 107784–107785, 2021.

[11] S. Manimurugan, S. Al-Mutairi, M. M. Aborokbah, N. Chilamkurti, S. Ganesan, and R. Patan, "Effective attack detection in internet of medical things smart environment using a deep belief neural network," *IEEE Access*, vol. 8, pp. 77396–77404, 2020.

[12] S. Latif et al., "Intrusion detection framework for the Internet of Things using a dense random neural network," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6435–6444, Sep. 2022.

[13] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K. R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021.

[14] M. M. Alani and A. I. Awad, "An intelligent two-layer intrusion detection system for the Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 683–692, Jan. 2023.

[15] H. Lu, T. Wang, X. Xu, and T. Wang, "Cognitive memory-guided autoencoder for effective intrusion detection in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3358–3366, May 2022.

[16] J. Li, L. Lyu, X. Liu, X. Zhang, and X. Lyu, "FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 6, pp. 4059–4068, Jun. 2022.

[17] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of things intrusion detection: Centralized, on-device, or federated learning?," *IEEE Netw.*, vol. 34, no. 6, pp. 310–317, Nov./Dec. 2020.

[18] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-IFS: Intrusion detection approach for industrial Internet of Things traffic in fog environment," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7704–7715, Nov. 2021.

[19] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.

[20] X. Kan et al., "A novel IoT network intrusion detection approach based on adaptive particle swarm optimization convolutional neural network," *Inf. Sci.*, vol. 568, pp. 147–162, 2021.

[21] Y. Otoum, D. Liu, and A. Nayak, "DL-IDS: A deep learning–based intrusion detection framework for securing IoT," *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 3, 2019, Art. no. e3803.

[22] Y. Chen, Q. Lin, W. Wei, J. Ji, K.-C. Wong, and C. A. C. Coello, "Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in fog computing," *Knowl.-Based Syst.*, vol. 244, 2022, Art. no. 108505.

[23] V. A. Memos, K. E. Psannis, and Z. Lv, "A secure network model against bot attacks in edge-enabled industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 7998–8006, Nov. 2022.

[24] N. Gupta, V. Jindal, and P. Bedi, "LIO-IDS: Handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system," *Comput. Netw.*, vol. 192, 2021, Art. no. 108076.

[25] A. Telikani, J. Shen, J. Yang, and P. Wang, "Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 23260–23271, Nov. 2022.

[26] W. Liang, Y. Hu, X. Zhou, Y. Pan, I. Kevin, and K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5087–5095, Aug. 2022.

[27] S. Kaur and M. Singh, "Hybrid intrusion detection and signature generation using deep recurrent neural networks," *Neural Comput. Appl.*, vol. 32, no. 12, pp. 7859–7877, 2020.

[28] C. Zhou, Y. Jia, and M. Motani, "Optimizing autoencoders for learning deep representations from health data," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 1, pp. 103–111, Jan. 2019.

[29] Y. Li et al., "Efficient and effective training of COVID-19 classification networks with self-supervised dual-track learning to rank," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 10, pp. 2787–2797, Oct. 2020.

[30] A. Telikani, A. Tahmassebi, W. Banzhaf, and A. H. Gandomi, "Evolutionary machine learning: A survey," *ACM Comput. Surv.*, vol. 54, no. 8, pp. 1–35, 2021.

[31] Q. Yang et al., "Beyond one-against-all (OAA) and one-against-one (OAO): An exhaustive and parallel half-against-half (HAH) strategy for multi-class classification and applications to metabolomics," *Chemometrics Intell. Lab. Syst.*, vol. 204, pp. 104107–104118, 2020.

[32] B. M. Ozyildirim and M. Kiran, "Levenberg–Marquardt multi-classification using hinge loss function," *Neural Netw.*, vol. 143, no. 1, pp. 564–571, 2021.

[33] I. Lera, C. Guerrero, and C. Juiz, "YAFS: A simulator for IoT scenarios in fog computing," *IEEE Access*, vol. 7, no. 1, pp. 91745–91758, 2019.

[34] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. IEEE Conf. Mil. Commun. Inf. Syst.*, 2015, pp. 1–6.

[35] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: BoT-IoT dataset," *Comput. Syst.*, vol. 100, no. 1, pp. 779–796, 2019.

Akbar Telikani (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and information technology from University of Guilan, Rasht, Iran, in 2012 and 2015, respectively. He is currently working toward the Ph.D. degree in computer science with the University of Wollongong, Wollongong, NSW, Australia.

His main research interests include intelligent transportation systems, privacy and security, machine learning, the Internet of Things (IoT), and evolutionary computation. His main focus is to develop unmanned-aerial-vehicle-aided intelligent transportation systems (ITS) and security protection mechanisms for the IoT.

Nima Esmi Rudbardeh (Graduate Student Member, IEEE) received the B.Sc. degree in computer software engineering from Islamic Azad University of Lahijan, Lahijan, Iran, in 2013, and M.Sc. degree in computer software engineering from Islamic Azad University Science and Research Branch, Iran, in 2016. He is currently working toward the Ph.D. degree in computer engineering with the University of Groningen, Groningen, The Netherlands, under the supervision of Prof. Georgi Gaydadjiev.

His current research interests include data fusion, deep learning, the Internet of Things, and their applications in various industrial fields.

Shiva Soleymanpour received the B.Sc. and M.Sc. degrees in computer engineering from Ayandegan Higher Education Institution, Tonekabon, Iran, in 2015 and 2019, respectively.

She is currently with the Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran. Her research interests include deep learning, class imbalance mitigation, Internet of Things, and network traffic analysis. Her work addresses class imbalance challenges in machine learning models, particularly within network traffic classification.

Asadollah Shahbahrami received the B.S. degree in computer engineering (hardware and machine intelligence) from the Iran University of Science and Technology, Tehran, Iran, in 1993, the M.Sc. degree in computer engineering (hardware and machine intelligence) from Shiraz University, Shiraz, Iran, in 1996, and the Ph.D. degree in computer engineering from the Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, Delft, The Netherlands, in 2008.

He held a faculty position with the Department of Electrical Engineering, University of Guilan, Rasht, Iran, Since August 1996, he has been with the University of Guilan, where he is a Full Professor with the Department of Computer Engineering. He is also a Visiting Professor for his sabbatical with the Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, Faculty of Science and Engineering, University of Groningen, Groningen, The Netherlands. His research interests include advanced computer architecture, machine learning, machine vision, multimedia processing, information retrieval, parallel processing, and parallel programming.

Jun Shen (Senior Member, IEEE) received the Ph.D. degree in british journal of education technology from Southeast University, Nanjing, China, in 2001.

He is currently a Full Professor with the University of Wollongong, Wollongong, NSW, Australia, and has established expertise in multidomain computational intelligence applications, including education, transport, manufacturing and bioinformatics. He was a Visiting Professor with the Massachusetts Institute of Technology, Cambridge, MA, USA, and the Georgia Institute of Technology, Atlanta, Georgia. He has authored or coauthored more than 300 papers in relevant fields in journals and conferences, including IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, IEEE TRANSACTIONS ON SERVICES COMPUTING, *Briefings in Bioinformatics*, International Conference on Machine Learning, Conference on Knowledge Discovery and Data Mining, and British Journal of Education Technology (BJET). He is an Associate Editor for four Tier 1 international journals and editorial members of other eight journals. He was a Panel Member of ACM/AIS MSIS2016 and also a Program Committee Chair or Member for more than 400 international conferences.

Dr. Shen is a Senior Member of the Association for Computing Machinery.

Georgi Gaydadjiev received the M.Sc. degree in electrical engineering and the Ph.D. degree in computer engineering from the Delft University of Technology, Delft, The Netherlands, in 1996 and 2007, respectively.

He is a Computer Engineer with more than 30 years of experience in the Industry and Academia. He is currently the Chair in Innovative Computer Architectures with the University of Groningen, Groningen, The Netherlands. He is an Honorary Visiting Professor with the Department of Computing, Imperial College London, London, U.K., and a Visiting Professor of Computer Engineering with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. He was the Chair in Computer Systems Engineering with the Chalmers University of Technology, Gothenburg, Sweden, and was a Vice-President of Dataflow Software Engineering with Maxeler Technologies Ltd., London, and a Managing Director of Maxeler IoT-Labs BV, Delft.

Reza Hassanpour received the B.S. degree from Shiraz University, Shiraz, Iran, in 1995, the M.S. degree from Tehran Polytechnic University, Tehran, Iran, in 1998, and the Ph.D. degree from Middle East Technical University, Ankara, Türkiye, in 2003, all in computer engineering.

He is currently a Full Professor with Gidatarim University Konya-Turkey, Konya, Turkey. He is also with the Department of Computer Science, Rotterdam University, Rotterdam, The Netherlands. His main research interests include artificial intelligence, machine learning, and image processing.