IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 15, 2021, accepted March 23, 2021, date of publication April 9, 2021, date of current version April 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3072126

Improving Data Generalization With Variational Autoencoders for Network Traffic Anomaly Detection

MEHRNOOSH MONSHIZADEH[1,2], (Graduate Student Member, IEEE), VIKRAMAJEET KHATRI[3], MARAH GAMDOU[4], RAIMO KANTOLA[2], (Member, IEEE), AND ZHENG YAN[2,5]

[1]Nokia Bell Labs, 91620 Nozay, France
[2]Department of Comnet, Aalto University, 02150 Espoo, Finland
[3]Nokia Bell Labs, 02610 Espoo, Finland
[4]Centralesupélec Engineering School, Paris-Saclay University, 91190 Gif-sur-Yvette, France
[5]The State Key Lab of Integrated Services Network, Xidian University, Xi'an 710071, China

Corresponding author: Mehrnoosh Monshizadeh (mehrnoosh.monshizadeh@nokia-bell-labs.com)

ABSTRACT Deep generative models have increasingly become popular in different domains such as image processing, though, they hardly appear in the cybersecurity arena. While the main application of these models is dimensionality reduction, marginally they have been utilized for overcoming challenges such as data generalization and overfitting issues inherited from feature selection methods. To solve the mentioned challenges, we propose a combined architecture comprising a Conditional Variational AutoEncoder (CVAE) and a Random Forest (RF) classifier to automatically learn similarity among input features, provide data distribution in order to extract discriminative features from original features, and finally classify various types of attacks. CVAE introduces the labels of traffic packets into a latent space in order to better learn the changes of input samples and distinguish the data characteristics of each class. It avoids the confusion between classes while learning the whole data distribution. Compared with feature selection mechanisms such as Support Vector Machine Online (SVMo) by considering various evaluation metrics, the proposed architecture demonstrates considerable improvement in terms of performance. To verify the versatility of the proposed architecture, two publicly available datasets have been used in experiments.

INDEX TERMS Anomaly detection, data mining, feature selection, machine learning, security.

I. INTRODUCTION

In the field of machine learning, feature selection is one of the well-known challenges. Many studies have been conducted with different techniques to solve the feature selection problem in overfitting contexts that have disastrous effects on anomaly detection performance.

Former techniques like Principal Component Analysis (PCA) or Autoencoders yield a framework to automate this process in an unsupervised manner respectively for linear and non-linear data representations. However, they reveal drawbacks since on the one hand PCA linear representations poorly represent data in most cases, and on the other hand, latent spaces derived in autoencoder often lack required regularities for model generalization.

In the recent years and in particular for imaging applications, the dual structure of Variational Autoencoders (VAE) show promising results on data compression or reconstruction. Furthermore, efficiency of VAE techniques, can be improved by data labelling adaptation, in their conditional version. These techniques mitigate overfitting and have nice potential for data model generalization. As counterparts, they are essentially used in a black-box way, their dimensioning lack deep understanding, they are not widely used outside the imaging domain and hardly appear in network cybersecurity applications.

Therefore, generalizing and assessing autoencoders' properties in a statistical framework would be a breakthrough in cybersecurity applications, where false alarm rates, detection probabilities, and classification error guaranties are still missing when classically using machine learning or deep learning tools.

The associate editor coordinating the review of this manuscript and approving it for publication was Michele Magno.

In a prior study [1], the authors applied various feature selection methods to achieve the highest efficiency for attack detection. However, in the earlier study, various challenges such as data generalization and overfitting had been discovered and in the current paper, the authors propose an architecture to overcome the addressed issue.

Feature selection techniques have been widely used as an initial stage of ML-based intrusion detection techniques. However, due to the lack of labelled datasets, these methods suffer from data generalization which may considerably degrade the accuracy.

While, there are manual techniques such as cross-validation to solve to some extent the overfitting problem, yet they will not be efficient for real-time intrusion detection. On the other hand, deep generative models can provide a feature representation by estimating of latent space of data. Following this characteristic and to improve detection accuracy, this paper proposes an effective deep learning method, namely CVAEwRF (Conditional Variational AutoEncoder with Random Forest). CVAE automatically learns similarity among input features, provides data distribution in order to extract discriminative features from original features, and finally RF efficiently classifies various types of attacks. The efficiency of the proposed model is evaluated against the well-known feature selection method Support Vector Machine online (SVMo). To verify the versatility of the proposed architecture, two publicly available datasets have been used in the experiments.

To improve the data generalization and overcome the overfitting challenge, this paper introduces a model by combining a classifier and a feature extraction method that significantly avoids overfitting normal data, accurately labels network traffic, and efficiently detects cyber-attacks.

While the majority of state-of-the-art methods have limited focus on the discussed challenge, our paper has a comprehensive review on the mentioned issue and with an optimized architecture overcoming the mentioned challenge. The major differences of this paper against related works are as follow:

- Most of the studies do not evaluate performance robustness. Their presented solution may improve detection rate in overall and only for a specific dataset while the performance varies considerably for another dataset and per each type of attack.
- Majority of studies presented limited evaluation metrics and do not present a detailed analysis.

Overall, the contribution of this paper is in introducing an efficient attack classifier with several characteristics:

- We propose an efficient architecture: Conditional Variational AutoEncoder with Random Forest (CVAEwRF) that applies a conditional VAE to extract the best features from an original dataset and utilizes a random forest algorithm to classify data into different categories (normal, unknown, and attack categories). This model achieves an effective representation and reduces dimensionality, which provides high detection rates. The achieved detection rates are mostly above 99.9% (overall and per attack class). The proposed architecture solves the overfitting issue.
- We evaluate the architecture efficiency per packet and based on uniform metrics including computation time, precision, recall, F1-score, Area Under Curve (AUC) and log loss, as well as Receiver Operator Characteristic (ROC) curves.
- We evaluate our system reliability against two different datasets that contain various types of attacks.

The rest of the paper is organized as follows. Section 2 provides a brief background in applying the feature selection method vs the feature extraction method. Section 3 gives an introduction about the architecture and applied algorithms. In Section 4, a random forest algorithm is implemented with feature selection SVMo and feature extraction CVAE method. Section 5 discusses the experimental results for scenarios described in the previous section. Finally, in Section 6, we draw a conclusion along with the scope of future research.

II. RELATED WORK

Wang *et al.* [2] propose a self-adversarial variational autoencoder and gaussian transformer machine to detect anomalies. In this study, the authors apply a regularization mechanism to add discrimination to anomalous classes during the training phase and on biased data in order to solve overfitting. The robustness of the mentioned model is tested against five different datasets. Though the applied mechanism sounds novel, the architecture is not utilized for network traffic dataset and intrusion detection application.

Yousefi-Azar *et al.* [3] apply an auto encoder-based feature selection model in order to generate more discriminative features and to reduce the dimensionality of features. The analysis is flow-based and two public datasets are used in the study, though the result of one dataset is presented. The applied dataset is categorized into two classes: normal and attack. The chosen proportion of each class in each fold (five-fold cross-validation) is equal. Furthermore, in this study, five different classifiers are applied in order to test model robustness. Authors claim to apply feature sets from both payload and header, though there are no details (e.g., on payload analysis) in the presented results. On the other hand, different evaluation metrics are introduced in the paper, while only accuracy and log loss are presented in the experimental results.

Yang *et al.* [4] introduce an improved conditional variational autoencoder combined with a deep neural network in order to solve the imbalance issue by generating new attack samples in the training phase. Authors also claim the proposed model can detect unknown attacks, however, in the results only normal and attack classes are presented and unknown class is missing from the analysis. For the experiments, the authors have applied three subsets of two public datasets rather than a full dataset. Furthermore, the authors compare the performance of their approach against five other oversampling techniques. Though the result shows improved

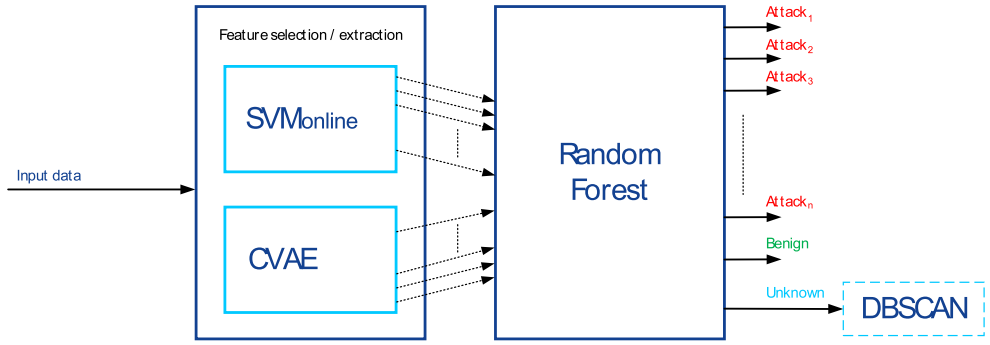


FIGURE 1. CVAEwRF architecture.

precision in comparison to the prior art, still the detection rate per each attack class is not considerably high.

Yang *et al.* [5] propose an intrusion detection model comprising of a supervised adversarial variational autoencoder with regularization, and a deep neural network. The architecture benefits from VAE data generation capability in order to synthesize samples of less frequent attacks and therefore solving the class imbalance issue in the training phase. In addition, GAN learning ability trains the adversarial learning model and a deep neural network classifies various types of attack. The model is tested with two public datasets containing 21 known and 14 unknown attack types, and its performance is compared against various classifiers and oversampling techniques. Even if, the proposed model solves issues such as class imbalance and improves the detection rate (precision), in comparison to the other discussed techniques, still the overall detection rate (highest achieved 91.94%) and especially per each attack class is not high (highest achieved 74.27%). Furthermore, model robustness is uncertain since the performance considerably varies for the two datasets used in this study.

Sun *et al.* [6] introduce a generative dictionary learning model for dimensionality reduction and in order to learn a normal dictionary on latent space of VAE for anomaly detection. Model is tested with three datasets, in which one is related to network traffic and two other datasets contain image and video. The evaluation is presented based on the F1 score and AUC metrics, though a detailed analysis per attack is lacking in this study.

Wei *et al.* [7] apply an unsupervised deep learning framework together with an unsupervised multi-autoencoder to detect insider threats. For this purpose, the authors analyze system logs. The model performance is compared against other machine learning algorithms and based on parameters including recall and AUC. However, no further information on attack classes is presented in this study.

Bedi *et al.* [8] present a two-layered hierarchical filtration solution to tackle the class imbalance issue. Two flow-based public datasets are used in this study and seven machine learning algorithms are applied for binary classification, in which three of them are implemented in the first layer. The study compares m-eXtreme Gradient Boost (m-XGBoost) and Siamese Neural Network (NN), where m-XGBoost is chosen for the 2nd layer. Recall, Precision and F1-scores show improvement in the score metrics of minority classes, while keeping those of majority class acceptable compared to other classification algorithms. Similar to other studies, the model may present an improved detection rate in comparison to state-of-the-art methods, but the achieved detection rate is not robust, specifically per attack class.

III. ARCHITECTURE

The proposed architecture in the current paper is a continuation of ongoing research that has been published previously as Hybrid Anomaly Detection Model (HADM) [1]. The architecture comprises a random forest classifier and a feature selection/extraction algorithm as shown in Fig. 1.

The feature selection (SVMonline) / extraction (CVAE) algorithm extracts the best features from the incoming packets and provides these features to the classifier algorithm (Random Forest) in order to classify data into different categories (normal, unknown and attack categories). DBSCAN algorithm that clusters unknown traffic is part of an ongoing project and will be published in the next paper.

A. APPLIED ALGORITHMS

For performance testing, the selected features are applied to an optimized random forest algorithm. The algorithm's internal architecture and parameters are explained below.

1) RANDOM FOREST

This algorithm comprises many decision trees. Each tree gives a classification, and we say the tree has a vote for that class. The forest chooses the classification having the most votes, over all other trees. Compared to a decision tree, a random forest is considered more stable and robust against overfitting. However, it is more difficult to interpret. To classify a new sample, it is placed in each of the trees. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature. For each decision tree, the node importance is calculated using Gini Importance (for binary tree in Scikit-learn) [9], [10]:

$$ni_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \quad (1)$$

where,

- ni_j is the importance of node j
- w_j is the weighted number of samples reaching node j
- C_j is the impurity value of node j
- $left(j)$ is child node from left split on node j
- $right(j)$ is child node from right split on node j

The importance for each feature on a decision tree is calculated as:

$$fi_i = \frac{\sum_{j:\ node\ j\ splits\ on\ feature\ i} ni_j}{\sum_{k\in\ all\ nodes} ni_k} \tag{2}$$

where,

- fi_i is the importance of feature i
- ni_j is the importance of node j

The final feature importance, at the random forest level is the average over all trees. The sum of feature's importance value on each tree is calculated and divided by total number of trees as seen in (3):

$$RFfi_i = \frac{\sum_{j\in\ all\ trees} normfi_{ij}}{T} \tag{3}$$

where,

- $RFfi_i$ is the importance of feature i calculated from all trees in the random forest model
- $normfi_{ij}$ is the normalized feature importance for i in tree j
- T refers to total number of trees

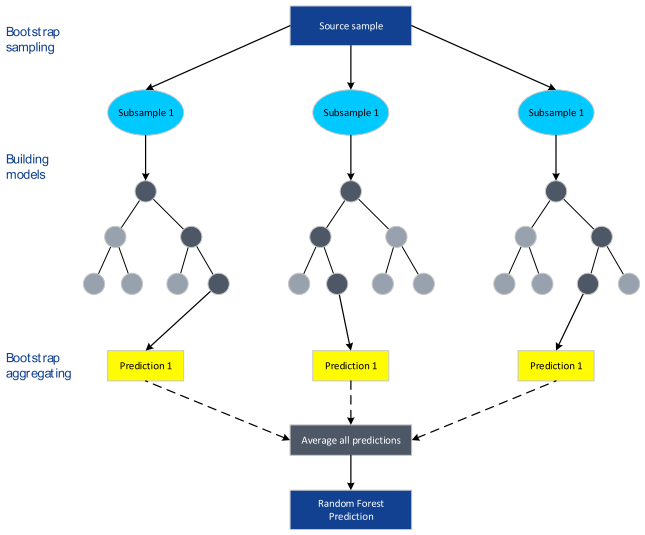


FIGURE 2. Random forest.

B. FEATURE SELECTION

Considered features in applied datasets are categorized as following [11]:

1) Flow features such as client-to-server or server-to-client.
2) Basic features representing protocols connections.
3) Content features encapsulating the attributes of Transport Control Protocol (TCP), Internet Protocol (IP) and Hypertext Transfer Protocol (HTTP) services.
4) Time features such as arrival time between packets, start or end packet time and round-trip time.
5) Labelled Features: this group represents the label of each record [12].
6) Additional features

It must be noted that network packets also carry a wide variety of irrelevant or redundant features. In this section, the feature characteristics of our datasets are examined to remove the unwanted features that affect the efficiency and detection rate of our algorithms. For this purpose, we apply the feature selection SVMonline method to find the best features from the datasets.

The latter method is compared to the feature extraction CVAE method that we applied, and which projects the input data in a new representation feature space called latent feature space. This low dimensional space is created based on new relevant features discovered by the CVAE. Though, the new features that are created by the CVAE and based on the original features are usually difficult to interpret.

The utilized algorithms are described below. It must be noted, though the current study, only applies CVAE for extracting features, still the VAE is explained in the following, since the applied CVAE utilizes the major structure of VAE.

1) SVMonline

Incremental SVM calculates the loss and retrains linear SVM in every batch using stochastic gradient descent. It assigns SVM weights to each feature and selects those with the highest absolute value as the best discriminative features. Although SVMonline relies on the linear dependency of features and labels as in F-Score, it is more robust than F-Score, since it splits the dataset into small batches and calculates the average of model coefficients that further increases the robustness [13].

2) VAE

This is an unsupervised Latent-variable-based deep generative model. VAE comprises two neural networks: an encoder network and a decoder network.

Encoder is a neural network that inputs a data point x and outputs a latent representation z. This latent variable z belongs to a latent space of lower dimension than the input space. The encoder has weights and biases φ. We denote the encoder as $q(z \mid x; \varphi)$, the distribution of the latent variable z.

Decoder is a neural network that receives the latent variable z as input and reconstructs $\hat{x}$ from the probability distribution $p(x \mid z; \theta)$. The decoder has weights and biases θ.

Loss function is a negative loglikelihood with a regularizer:

$$D_{KL}\left(q(z \mid x; \varphi) \| p(z; \theta)\right) - E_{z\sim q\varphi}\left[\log p(x \mid z; \theta)\right] \tag{4}$$

$p(z)$ is the expected distribution (the prior) of z which is specified as a standard normal distribution with mean zero and variance one.

An observation x is assumed to be distributed according to $p(x \mid z; \theta*)$, where the decoder takes as input z and

outputs $p(x \mid z; \theta)$. The choice of this distribution depends on the type of data. In this paper, we applied a multivariate gaussian distribution as it is usually used when the input data is continuous. In order to estimate θ to get the closest possible $p(x \mid \theta)$ to the true data distribution, the decoder can be fit by maximizing the marginal likelihood as seen in (5):

$$p(x; \theta) = \int p(x \mid z; \theta)\, p(z)\, dz \tag{5}$$

Unfortunately, this likelihood can't be evaluated or approximated as it is intractable. Even trying to use $p(z \mid x; \theta)$ will not solve this problem because $p(z \mid x; \theta)$ is intractable too.

Variational autoencoder model solves this problem by using variational inference which uses majorization-minimization principles to solve this optimization problem. The approach is to approximate $p(z \mid x; \theta)$ using an encoder network and to use this approximation to estimate a lower bound on the marginal log-likelihood. As a result, the model will learn its parameters by maximizing this lower bound (the Evidence Lower Bound).

We consider $q(z \mid x; \varphi)$ as the approximating distribution of $p(z \mid x; \theta)$ where $q(z \mid x)$ is a multivariate gaussian distribution). It is parametrized with the encoder that takes as input x and outputs $q(z \mid x; \varphi)$.

The marginal log-likelihood of an observation x and for any variational distribution $q(z \mid x; \varphi)$ over the latent variables z can be expressed as follows:

$$\log p(x; \theta) = L(x; \varphi, \theta) + D_{KL}(q(z|x; \varphi) \| p(z|x; \theta)) \tag{6}$$

where $L(x; \varphi, \theta)$ represents the Evidence Lower Bound (ELBO) as seen in (7):

$$L(x; \varphi, \theta) = E_{z \sim q\varphi}\left[\log p(x, z; \theta) - \log q(z \mid x; \varphi)\right] \tag{7}$$

As the Kullback-Leibler divergence is non-negative: $\log p(x; \theta) \geq L(x; \varphi, \theta)$ with equality only when $q(z \mid x; \varphi) = p(z \mid x; \theta)$. Therefore, the objective function maximized in variational inference is:

$$\begin{aligned} L(x; \varphi, \theta) &= E_{z \sim q\varphi}\left[\log p(x, z; \theta) - \log q(z \mid x; \varphi)\right] \\ &= -D_{KL}(q(z \mid x; \varphi) \| p(z; \theta)) \\ &\quad + E_{z \sim q\varphi}\left[\log p(x \mid z; \theta)\right] \end{aligned} \tag{8}$$

As it is shown in (8), the ELBO has two terms. The first is the *KL* divergence term which is a regularization term. It ensures that the encoder stays close to the prior. The second is the reconstruction term. Even if we don't always have an analytical expression of the ELBO, we can have an approximation of it using Monte Carlo estimate [14].

3) CVAE

It is a conditional version of VAE where the decoder network takes label y as an additional input in order to generate a sample that belongs to the class indicated by the label, i.e. label y is concatenated with latent vector z. Therefore, instead of having $p(x \mid z; \theta)$ as the likelihood that is parametrized by the decoder, we will have $p(x \mid z; \theta, y)$ which is a conditional probability that depends on input label y.

This CVAE helps to make classes of input data more distinguishable as it forces the VAE to take class labels into account in latent space. CVAE can be seen in Fig. 4 below.

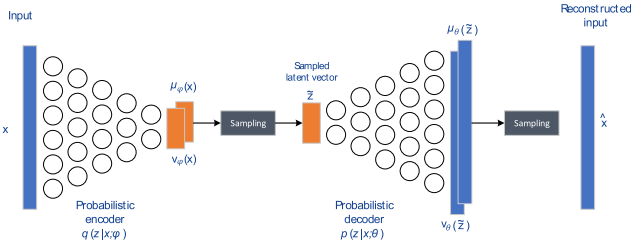


FIGURE 3. Variational autoencoder architecture.

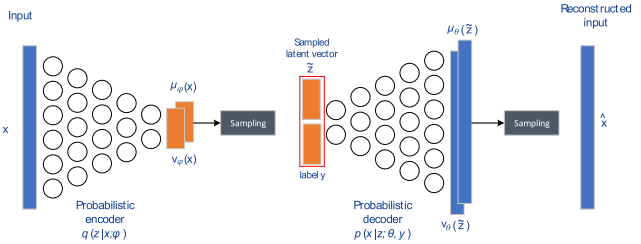


FIGURE 4. Conditional variational autoencoder architecture.

C. EVALUATION METRICS

To evaluate CVAEwRF detection rate, applied metrics such as accuracy score, precision, recall, F1 score, confusion matrix and AUC are briefly explained. We consider classes of normal (-1), unknown (0), attack $(1, \ldots, n)$.

1) **Accuracy score:** It computes the count of correct predictions:

$$Accuracy\left(y, y'\right) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} 1\left(y_i' = y_i\right) \tag{9}$$

In (9), $y_i^{'}$ refers to the predicted value of i^{th} sample, y_i refers to the corresponding true value and $1(x)$ is the indicator function.

2) **Precision:** It is the ability of a classifier not to wrongly label a negative sample as positive. In other words, how many of the selected objects were correct. Precision is calculated with:

$$Precision = \frac{TP_i}{TP_i + FP_i} \tag{10}$$

where,

- TP_i or True Positive: Is the number of instances with an actual class other than the *i-th,* and correctly predicted to belong in the *i-th* class. For binary classification, this metric represents the malicious traffic that is correctly identified as an attack.
- FP_i or False Positive: Is the number of instances with an actual class other than the *i-th*, but wrongly

predicted to belong in the *i-th* class. For binary classification, this metric represents the safe traffic that is incorrectly identified as an attack.

3) **Recall:** It refers to the ability of a classifier to find all positive samples. In other words, how many of the objects that should have been selected were actually selected. Recall is calculated with:

$$Recall = \frac{TP_i}{TP_i + FN_i} \tag{11}$$

where,

- FN_i or False Negative: Is the number of instances with *i-th* being the actual class, but falsely predicted to belong to another class. For binary classification, this metric represents the malicious traffic that is incorrectly identified as safe traffic.

4) **F1 score:** It is the weighted average of the precision and recall and is calculated with:

$$F1\ score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

5) **Confusion matrix:** An example can be seen in Fig. 5 and it can help in determining TP, FP, TN, and FN as well.

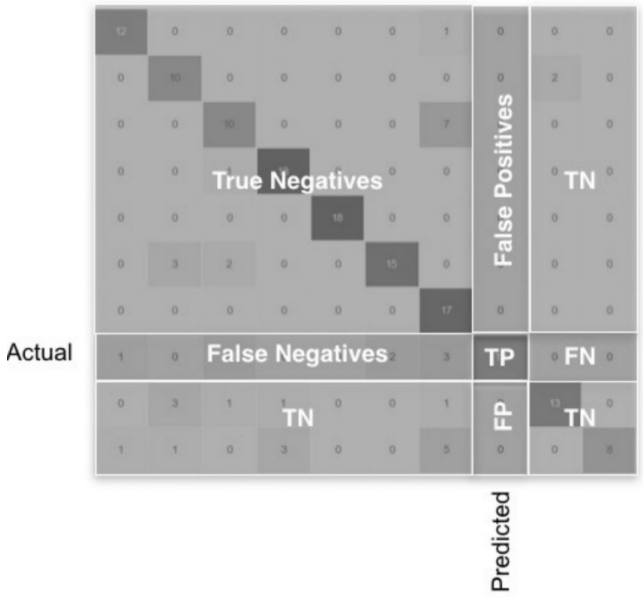


FIGURE 5. Confusion matrix [16].

It summarizes the performance of a classification algorithm and is represented in table format. The diagonal elements represent the number of points for which the predicted label and true label are equal; whereas non-diagonal elements represent mislabeled prediction by classifier [15].

6) **ROC:** This curve shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples and present a view of the discriminant ability of a classification model. The goal of a classifier is to be in the upper-left-hand corner in ROC space.
In the ROC graph, the x-axis represents the False Positive Rate (FPR) and the y-axis represents the True Positive Rate (TPR) where:

$$True\ Positive\ Rate = \frac{TP_i}{TP_i + FN_i} \tag{13}$$

True Positive Rate is the fraction of positive examples that are correctly classified.

$$False\ Positive\ Rate = \frac{FP_i}{FP_i + TN_i} \tag{14}$$

False Positive Rate is the fraction of negative examples that are misclassified as positive.
AUC stands for the area under the ROC curve is a measure between 0 and 1 that describes the discriminant ability of a classifier. It is the probability that a model ranks a randomly chosen positive sample higher than a randomly chosen negative one. The AUC value close to 1 means the detection results are credible [17], [18].

In order to evaluate the quality of the classifier and effectiveness of correctly identifying the intrusion. Below criterion is considered for AUC value:
AUC = 1, accurate results
AUC = [0.85, 0.95], good results
AUC = [0.7, 0.85], general result
AUC = [0.5, 0.7], less accurate results
AUC = 0.5, random prediction
AUC < 0.5, worse than random prediction

7) **Log loss:** It is also called cross-entropy loss or logistic loss and is defined as the negative log-likelihood of a classification model.
Let's consider a classification task with n classes. Suppose that $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\}$ is a training dataset of n samples, where y_i is the label of the *i-th* sample x_i. This label is represented as a one-hot vector. Suppose that p_i is a vector in which the *j-th* (with $j \in \{1, 2, \ldots, n\}$) element is the probability that sample x_i is assigned to the *j-th* class. Then, the log loss can be defined as follows [19], [20]:

$$Log\ loss = -\frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}} \left(y_i^T \log\left(p_i\right)\right) \tag{15}$$

IV. IMPLEMENTATION PHASES

The implementation of CVAEwRF involves data cleaning and processing of datasets, applying sampling techniques and feature selection/extraction algorithm to improve the detection performance, and finally utilizing a classification algorithm to categorize input traffic.

A. DATASETS

In order to evaluate our model performance, we applied two publicly available datasets that include diverse attacks and

TABLE 1. Comparison between different publicly available datasets.

Dataset	Year	Labelled	Protocol	Attacks
MAWILab	2018	Yes	HTTP, HTTPS, SSH[a], FTP[b], DNS[c], email, BGP[d]	DoS, Scan, Worm, Browser, Brute force, DNS, others (SMB[e], RPC[f], RST[g], Sasser, Netbios, SYN[h], FIN[i], Ping flood, FTP, SSH)
ISCX	2012	Yes	HTTP, SSH, FTP, email	DoS, Scan, Backdoor, Brute force, Browser and other attacks

[a]SSH Secure SHell
[b]FTP File Transfer Protocol
[c]DNS Domain Name System
[d]BGP Border Gateway Protocol
[e]SMB Server Message Block
[f]RPC Remote Procedure Call
[g]RST A TCP flag to reset the connection
[h]SYN A TCP flag to synchronize sequence numbers
[i]FIN A TCP flag to indicate no more data from sender

meet the real traffic criteria to some extent. Table 1 presents the datasets characteristics. Datasets are classified into normal, unknown and *n* classes of attacks.

Packets that do not have any label in the dataset are presented in an unknown class for further investigation by a clustering algorithm (DBSCAN).

- For ISCX-2012, there are packets that cannot be correlated to any of the provided labels by the dataset.
- For Mawilab-18, there is an unknown class from the beginning which is labelled as such.

The ISCX-2012 dataset was captured in 2012 over one week and in an emulated environment. Dataset includes normal and malicious traffic [21]–[23]. Table 1 provides detailed information about this dataset. For our experiments, the attacks that are listed in Table 1 are grouped into three categories: L2R, R2L and L2L.

The MAWILab-2018 dataset is captured at a link between USA and Japan, every day and over a long time. For the current paper, the traffic from 28th August 2018 is used [24], [1]. Furthermore, in order to check model resilience and robustness and to have a diversity of attacks, all the DoS attacks contained in ISCX-2017 dataset [23] were extracted and injected into Mawilab-2018 (DoS attack class). More information about this dataset is presented in Table 1.

While the network traffic payload may have different characteristics for every dataset, this study only analyzes the header of the network traffic datasets that consist of similar attributes and protocols. Therefore, mixing datasets has not been an issue as the data points were also close in the feature space during the experiments.

B. DATA PREPROCESSING

Data cleaning, converting the columns to the right types, handling missing values, splitting IP addresses into four fields, vectorizing categorical variables, normalizing the dataset, changing the labels of attack categories in order to differentiate different attack categories are the processes carried out in this phase. For the normalization, statistical and scaling normalization are used [25]. In order to improve the performance of the algorithms, numeric attributes are transformed into nominal attributes. In addition, the IP address and hexadecimal Medium Access Control (MAC) address of the applied datasets are transformed into separate numeric attributes. Each numeric attribute is normalized using batch mean and standard deviation unless there is an already defined range (e.g., IP address range) [1].

Distribution of packets in datasets is shown in Table 2; whereas distribution of packets for testing and training is shown in Table 3, about, the 2/3 of data is used for training the phase and 1/3 is used for testing.

V. EXPERIMENTAL RESULTS

All the experiments are carried out on a server with Intel®Xeon®16 x E5-2623 CPU @3.0GHz (4 cores in each processor), 128 GB RAM and 1.6 TB HDD. The scripts were developed in Python in a Linux environment (Ubuntu 20.04.1 LTS) and utilized Scikit-learn library [26]; for CVAE Tensorflow2 and Tensorflow-probability are used [26]. Random forest algorithm is trained once (with SVMo and VAE) and the trained model is saved for future tests.

The proposed approach is tested, and performance is evaluated with two architecture combined of the below algorithms.

i) Random forest classifier with SVMo feature selection
ii) Random forest classifier with CVAE feature extraction

The box plots are used to represent the distribution of data for each feature. The distribution is displayed based on minimum value, first quartile (Q1/25th percentile), median (Q2/50th percentile), third quartile (Q3/75th percentile) and maximum value as shown in Fig. 6.

Selected Features via SVMo: The distribution of the features selected using SVMo for ISCX-2012 and MAWILab-2018 datasets can be seen in Fig. 6 and Fig. 7 respectively.

Extracted Features via CVAE: The box plots in Fig. 8 and Fig. 9 show the distribution of input data for each feature that is created in the latent space for datasets ISCX-2012 and MAWILab-2018 respectively.

All of these figures depict the variation of data in the feature space for each dataset and for each technique (feature selection and features extraction). Notice that for many

TABLE 2. Distribution of packets in datasets.

Dataset	Class	Packets	Total
MAWILab-2018	Normal	19383493	52788477
	Unknown	27500773	
	Attack Class 1 (DoS)	2147726	
	Attack Class 2 (Multi. Points mptmp)	2193412	
	Attack Class 3 (Multi. Points mptp)	54339	
	Attack Class 4 (Multi. Points ptmp)	1082426	
	Attack Class 5 (HTTP attack)	215698	
	Attack Class 6 (Network scan TCP)	145081	
	Attack Class 7 (Network scan UDP)	65517	
	Attack Class 8 (TTL error)	12	
ISCX-2012	Normal	11000795	28000002
	Unknown	16787734	
	Attack Class 1 (L2R)	96846	
	Attack Class 2 (R2L)	46616	
	Attack Class 3 (L2L)	68011	

TABLE 3. Distribution of distribution of training and testing packets in datasets.

Dataset	Class	Training	Training Total	Testing	Testing Total
MAWILab-2018	Normal	12917244	35192318	6466249	17596159
	Unknown	18337756		9163017	
	Attack Class 1 (DoS)	1432894		714832	
	Attack Class 2 (Multi. Points mptmp)	1462422		730990	
	Attack Class 3 (Multi. Points mptp)	36353		17986	
	Attack Class 4 (Multi. Points ptmp)	721504		360922	
	Attack Class 5 (HTTP attack)	143427		72271	
	Attack Class 6 (Network scan TCP)	97035		48046	
	Attack Class 7 (Network scan UDP)	43673		21844	
	Attack Class 8 (TTL error)	10		2	
ISCX-2012	Normal	7701727	19600001	3299068	8400001
	Unknown	11750726		5037008	
	Attack Class 1 (L2R)	67605		29241	
	Attack Class 2 (R2L)	32399		14217	
	Attack Class 3 (L2L)	47544		20467	

selected features, the data values are concentrated closely near the median. However, data values are more widely spread out from the median for extracted features because of the use of the prior distribution (a standard normal distribution). Note that it may be more difficult to separate the data into different categories when they are represented by close data points in the feature space.

A. EXPERIMENT 1

In this experiment, SVMo feature selection, Random Forest classifier, and the entire MAWILab-2018 dataset are used. The applied dataset is categorized into 10 classes (normal, unknown, and 8 attack categories). However, there are only 12 samples in the attack class 8 (TTL error from the attack category Other). The lack of enough samples in this class caused a huge challenge for the classifier to learn the right pattern. As a result, RF is classifying the samples of this very skewed class randomly for both SVM and VAE. Therefore, this class is removed from all MAWILab-2018 experiments. The mentioned class imbalance issue and a method to overcome the challenge will be addressed in a separate paper.

In order to be able to use the whole MAWILab-2018 dataset and solve the memory problem, the following techniques are available:

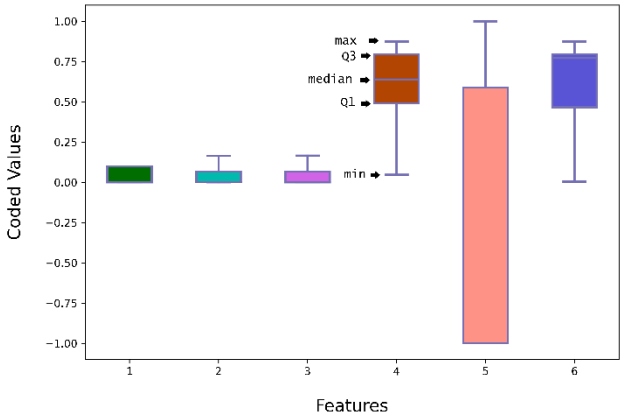


FIGURE 6. Selected features for MAWILab-2018.

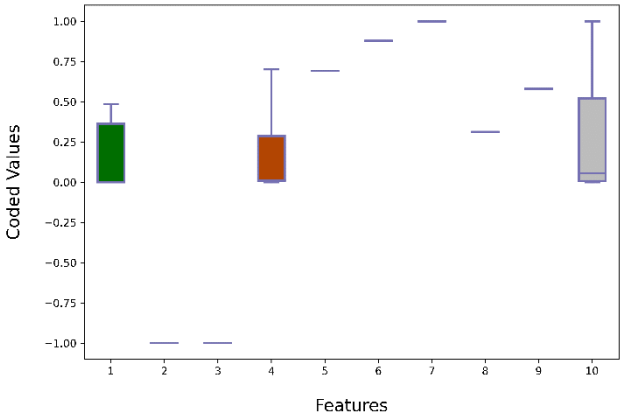


FIGURE 7. Selected features for ISCX-2012.

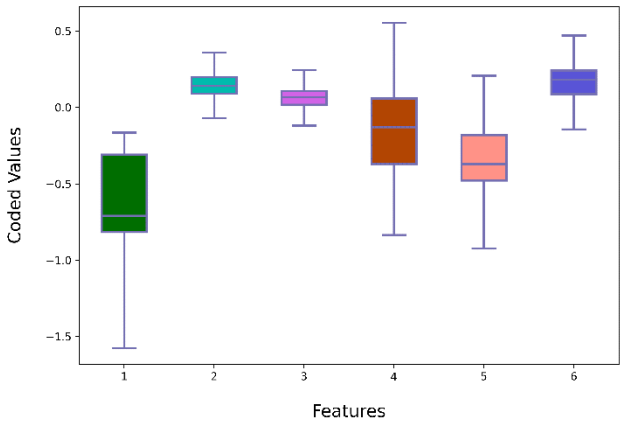


FIGURE 8. Extracted features for MAWILab-2018.

a) Using partial fit which is not implemented with Random Forest in sklearn [28], [29].
b) Using warm start that takes the first model as initialization and retrains it [28].
c) Training separate random forests using a part of MAWILab-2018 for each RF and aggregating them in one forest at the end.
d) Doing the data processing in two times and then merging the obtained datasets in order to use them later as a whole.

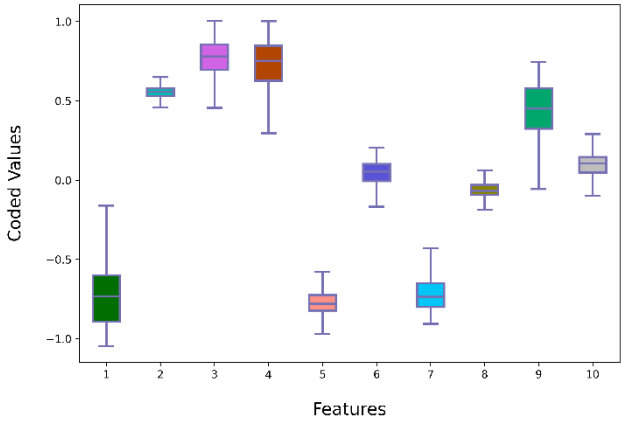


FIGURE 9. Extracted features for ISCX-2012.

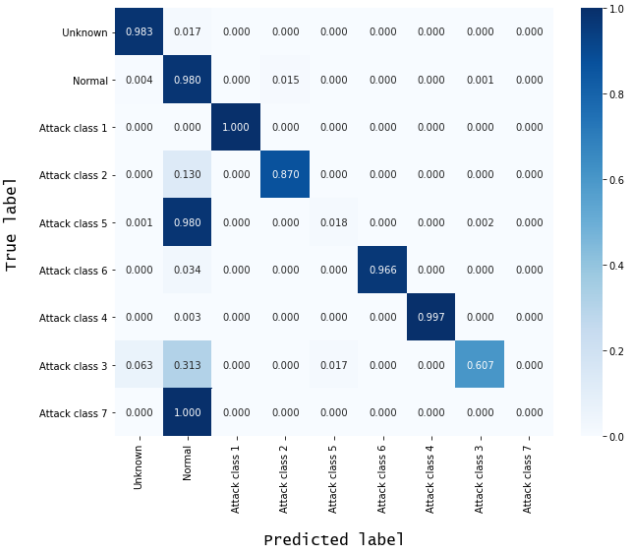


FIGURE 10. Confusion matrix for SVMonline, RF on MAWILab-2018.

For this experiment, the last solution is used, since it allows to train the model on all MAWILab-2018 and in a single step. Figures 10-12 present the experimental result for this scenario.

Figure 10 represents a normalized confusion matrix that has the recall of each class on its diagonal. This confusion matrix shows that the model which is composed of SVMo and RF is confusing many attack classes with normal traffic which is highly undesirable in an intrusion detection scenario.

Figure 11 shows classification metrics, for each class. These metrics emphasize the fact that the performance of this model (SVMo and RF) is unsatisfactory for many attack classes.

A more complete characterization of the combined SVMo and RF performance is the ROC curves depicted in Fig. 12 along with AUC scores for all classes (unknown, normal, and attack categories).The ROC curve of a random classifier (the worst scenario) is represented (in red) in this figure as a reference. Notice that the goal of the classifier is to be in the upper-left-hand corner in ROC space for each

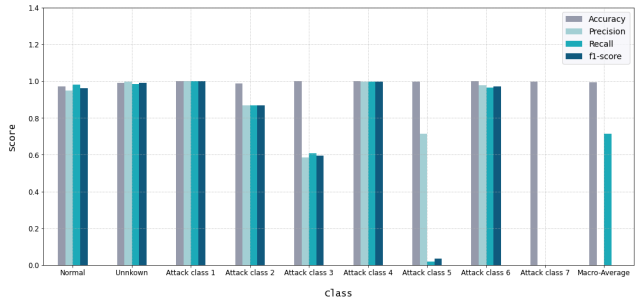


FIGURE 11. Accuracy, precision, recall and f1-score for SVMonline, RF on MAWILab-2018.

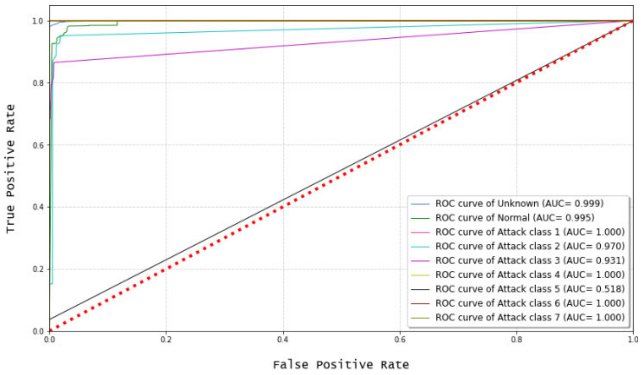


FIGURE 12. ROC curve for SVMonline, RF on MAWILab-2018.

class. In this experiment, the classifier doesn't have a very good discriminant ability for most of the classes as shown in this figure. Note that the ROC curve of the attack class 5 is close to the curve of a random classifier. This means that the model has no discriminative capacity to distinguish class 5 from other classes.

For this experiment, computation time is 47.14 s and the log loss score is 0.3043.

B. EXPERIMENT 2

In this experiment, a Random Forest classifier, CVAE feature extraction, and MAWILab-2018 dataset are used. Conditional Variational AutoEncoder reduces dimensionality in preparation for the classification algorithm (Random forest).

The CVAE's encoder is used after the training and the decoder will be used only during the training. As the CVAE describes the variability in the data it will be used to synthesize the input data that has 42 features in order to extract only 6 features.

The model set up is as follow:

a) The prior distribution is a standard normal distribution.
b) Encoder and decoder distributions are multivariate Gaussian distributions.
c) Both encoder and decoder have only one dense layer with a dimension of 20 and hyperbolic tangent activation function.
d) The used optimizer is Nadam (Nesterov-accelerated Adaptive Moment Estimation) that combines Adam and Nesterov's Accelerated Gradient (NAG) [30].
e) The best performing learning rate is 0.001.
f) The selected batch size is 30000.
g) The number of epochs is set to 200 while using early stopping and restoring of best weights.
h) To avoid overfitting problem, l2 regularization is used and its parameters is set to the commonly used value of 0.001.

For implementation, Tensorflow and Tensorflow-probability are used to create the model as they have many choices for non-probabilistic and probabilistic layers. The experimental results are shown in Fig. 13, 14, 15.

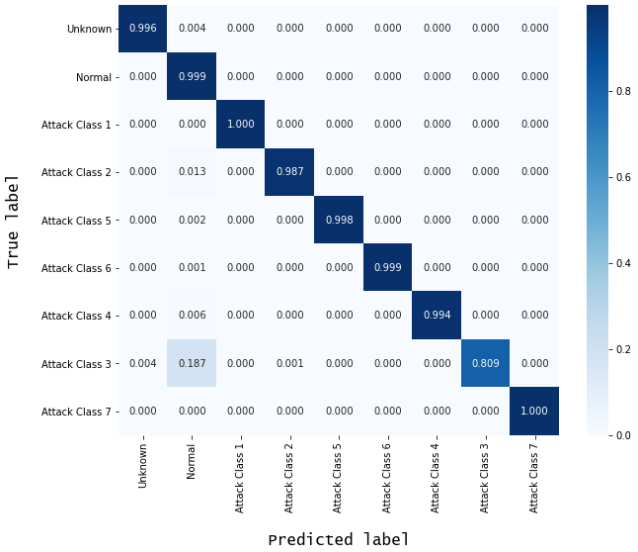


FIGURE 13. Confusion matrix for CVAE, RF on MAWILab-2018.

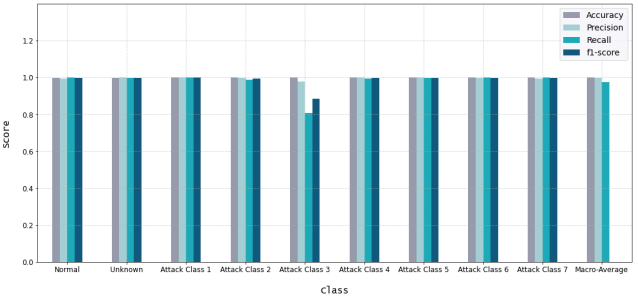


FIGURE 14. Accuracy, precision, recall and f1-score for CVAE, RF on MAWILab-2018.

Figure 13 represents a normalized confusion matrix that has the recall of each class on its diagonal. By comparing this confusion matrix to the one obtained using SVMo and RF (Fig. 8), we can see that the CVAE helps the RF to distinguish all the attack classes from normal traffic and to correctly classify input samples.

Figure 14 shows the classification metrics for each class (normal, unknown, and attack classes). By comparing this figure to Fig. 11, notice that the combined CVAE and RF model has significantly improved the performance of most of the classes. Note that the overall performance that is

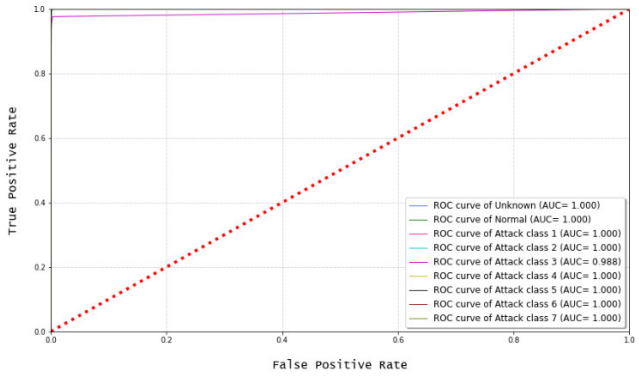


FIGURE 15. ROC curve for CVAE, RF on MAWILab-2018.

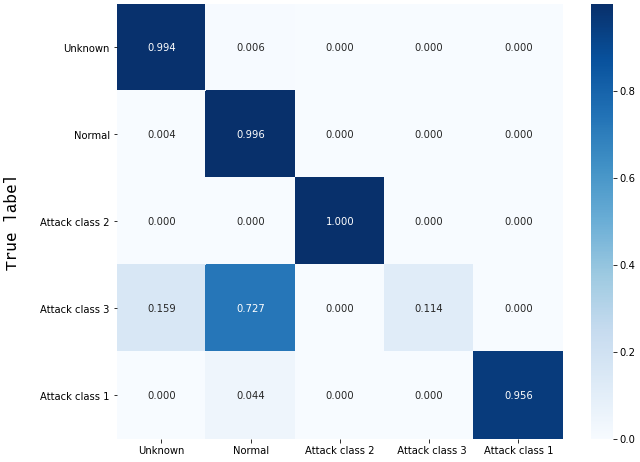


FIGURE 16. Confusion matrix for SVMonline, RF on ISCX-2012.

represented by the macro-averaging metrics is notably better than the previous one (the performance of SVMo and RF).

A more complete characterization of the combined CVAE and RF performance is the ROC curves depicted in Fig. 15 along with AUC scores for all classes (unknown, normal, and attack categories).The ROC curve of a random classifier (the worst scenario) is represented (in red) in this figure as a reference. Notice that all these ROC curves dominate the ROC curves of the SVMo and RF classifier that are represented in Fig. 12. This can be also checked by comparing AUC scores that are better for CVAEwRF model. Note that the problem of class 5 is totally solved and that the RF is no longer classifying the samples of the latter class randomly.

For this second experiment, the computation time is 82.72 s and the log loss score is 0.0249.

C. EXPERIMENT 3

In order to check the robustness of our approach, in the two following experiments, a subset of ISCX-2012 dataset has been used. This subset doesn't depend on the days. It is selected randomly but still, it keeps the original statistics with respect to the proportion of each attack.

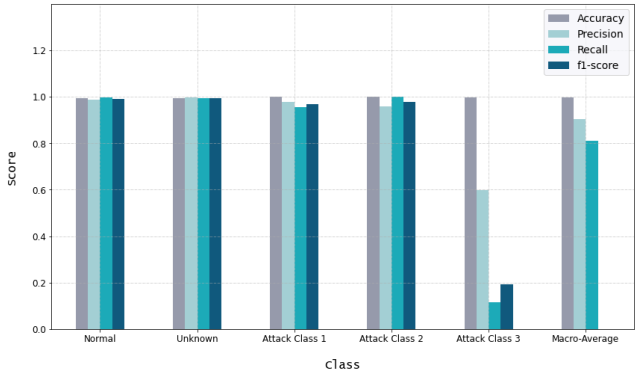


FIGURE 17. Accuracy, precision, recall and f1-score for SVMonline, RF on ISCX-2012.

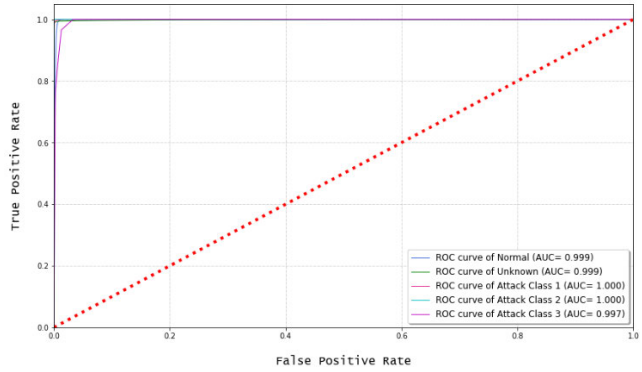


FIGURE 18. ROC curve for SVMonline, RF on ISCX-2012.

The current scenario utilizes SVMonline feature selection algorithm and a Random Forest classifier. Figures 16-18 illustrate the experimental results.

Figure 16 represents a normalized confusion matrix having the recall of each class on its diagonal. Notice that the classifier is confusing attack class 3 with normal traffic. This problem is similar to the one we had with MAWILab-2018 dataset.

This confusion between attack class 3 and normal traffic has a disastrous effect on the classification metrics of this same class, which are represented in Fig. 17 along with all classification metrics of the other classes. This problem affects the overall performance, which is shown through macro-averaging metrics, too. This impacts the ROC curves which characterize the discriminant ability of the model, as shown in Fig. 18. Notice that attack class 3 has the worst performance as its ROC curve is dominated by all other ROC curves.

For this third experiment, the computation time is 19.55 s and the log loss score is 0.0367.

D. EXPERIMENT 4

The current experiment applies CVAE feature extraction method and furthermore, in order to label the traffic, output of CVAE is fed to Random Forest classifier. The result of

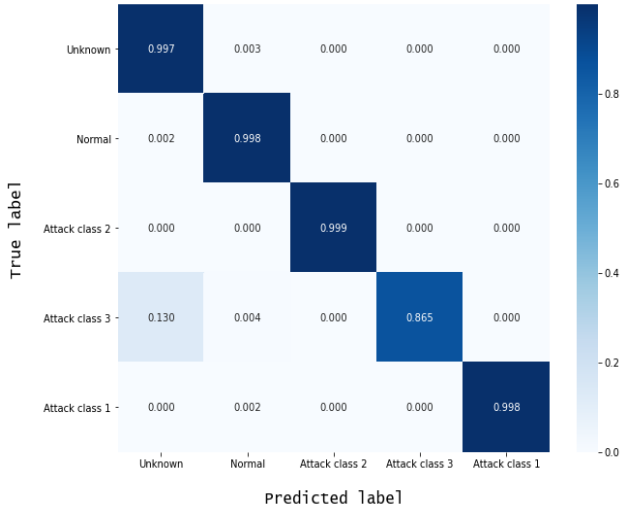


FIGURE 19. Confusion matrix for CVAE, RF on ISCX-2012.

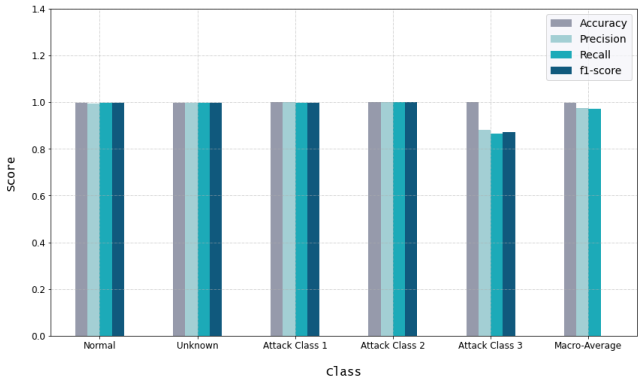


FIGURE 20. Accuracy, precision, recall and f1-score for CVAE, RF on ISCX-2012.

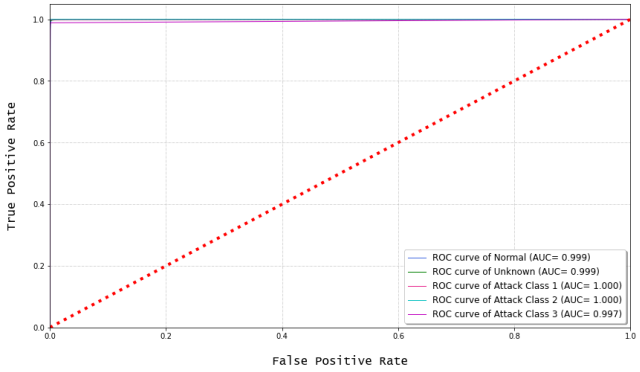


FIGURE 21. ROC curve for CVAE, RF on ISCX-2012.

mentioned combination is shown in Figures 19-21. As a result, 10 features are extracted from the original 42 features.

The normalized confusion matrix depicted in Fig. 19 proves that CVAE not only improves significantly the ability of the RF classifier to distinguish attack class 3 from other classes but also gives better results for other classes most of the time.

This improvement is also reflected through the classification metrics which are represented in Fig. 20. Note

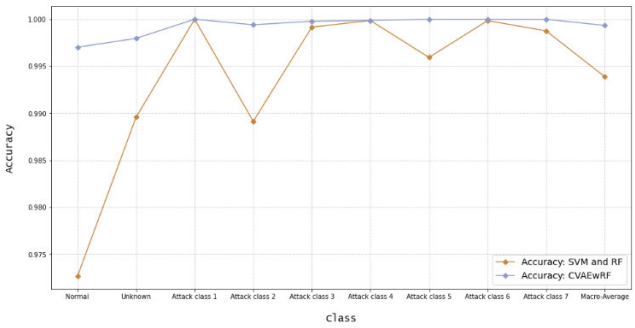


FIGURE 22. Accuracy for SVMonline, CVAE on MAWILab-2018.

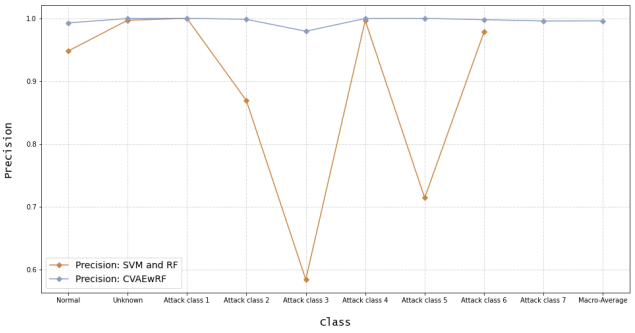


FIGURE 23. Precision for SVMonline, CVAE on MAWILab-2018.

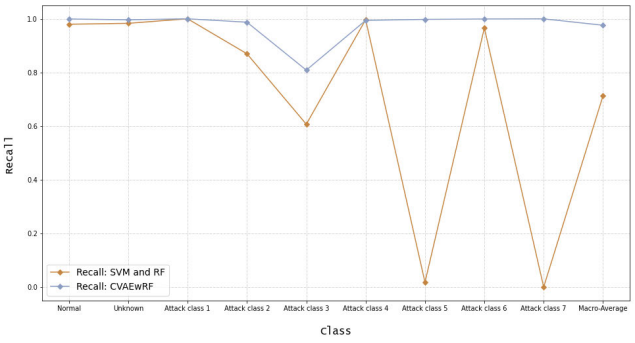


FIGURE 24. Recall for SVMonline, CVAE on MAWILab-2018.

this significant improvement by comparing this figure with Fig. 17 where the metrics of the previous classifier (SVMo and RF) are shown.

Notice that the ROC curve of attack class 3, which is shown in Fig. 21 along with all the other ROC curves, dominates the ROC curve of this same class that was obtained with SVMo and RF. This means that the discriminant ability of the model has improved.

For this experiment, the computation time is 46.37 s and the log loss score is 0.0229.

E. PERFORMANCE EVALUATION FOR SVMonline VS CVAE ON MAWILab-2018

The following figures compare the accuracy, precision, recall, and F1 score metrics which are obtained for SVMo with RF and CVAEwRF when they are applied to MAWILab-2018.

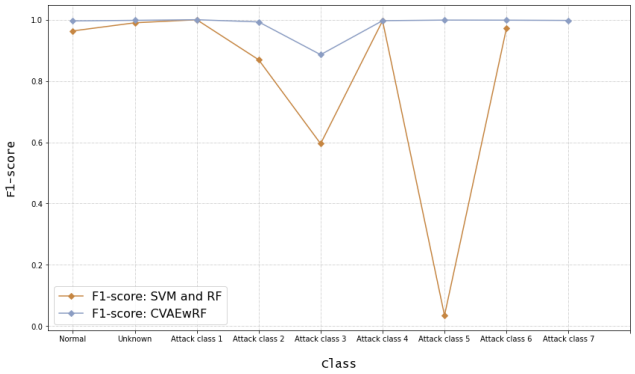


FIGURE 25. F1-Score for SVMonline, CVAE on MAWILab-2018.

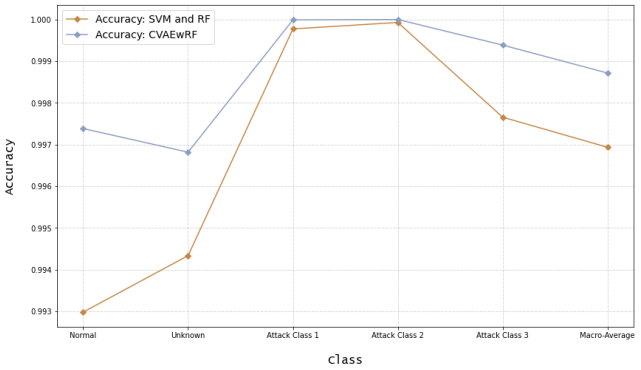


FIGURE 26. Accuracy for SVMonline, CVAE on ISCX-2012.

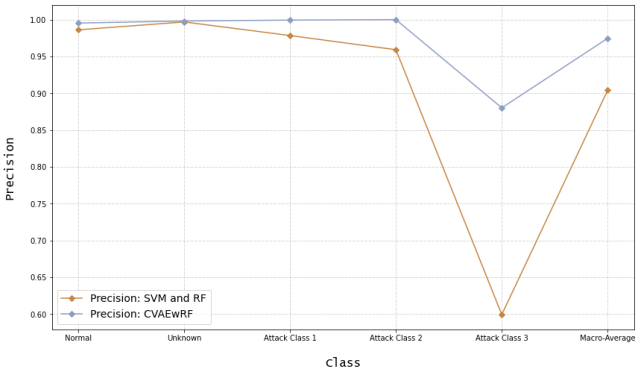


FIGURE 27. Precision for SVMonline, CVAE on ISCX-2012.

These metrics are represented for every class (normal, unknown, and attack) and for the overall classifier (through macro-averaging).

All these figures show that the class performance and the overall performance of CVAEwRF is better than the performance of SVMo with RF.

F. PERFORMANCE EVALUATION FOR SVMonline VS CVAE ON ISCX-2012

The following figures compare the accuracy, precision, recall, and F1 score metrics which are obtained for SVMo with RF and CVAEwRF when they are applied to ISCX-2012.

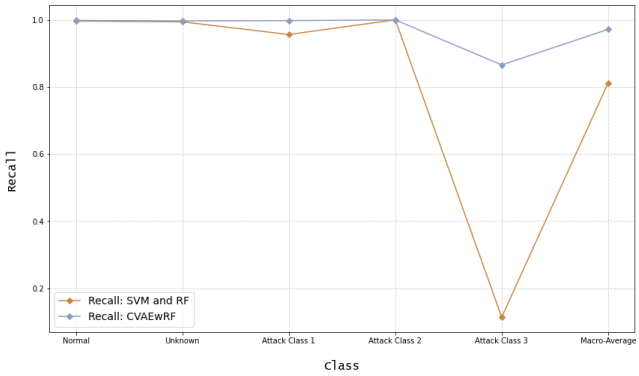


FIGURE 28. Recall for SVMonline, CVAE on ISCX-2012.

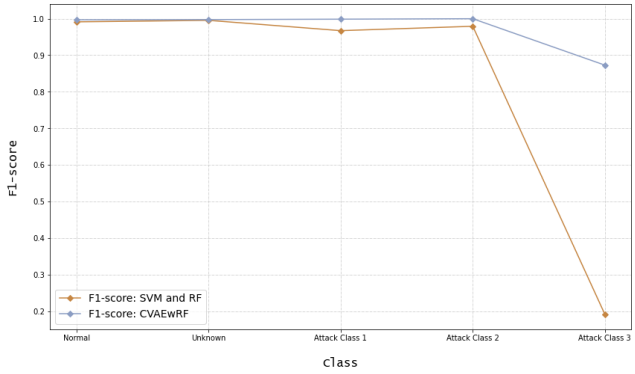


FIGURE 29. F1-Score for SVMonline, CVAE on ISCX-2012.

These metrics are represented for every class (normal, unknown, and attack) and for the overall classifier (through macro-averaging).

All these figures show that the class performance and the overall performance of CVAEwRF is better than the performance of SVMo with RF.

VI. CONCLUSION AND FUTURE WORK

In a prior study [1], the authors applied various feature selection methods to achieve the highest efficiency for attack detection. However, in the earlier study, various challenges such as data generalization and overfitting had been discovered and in the current paper, authors propose an architecture to overcome the addressed issue.

Feature selection techniques have been widely used in intrusion detection for many years. However, due to the lack of labelled datasets, these methods suffer from data generalization which may considerably degrade the accuracy.

While, there are manual techniques such as cross-validation to solve to some extent the overfitting problem, yet they will not be efficient for real time intrusion detection. On the other hand, deep generative models can provide a feature representation by estimating of latent space of data. Following this characteristic and to improve detection accuracy, this paper proposes an effective deep learning method, namely CVAEwRF (Conditional Variational AutoEncoder with

Random Forest). CVAE automatically learns similarity among input features, provides data distribution in order to extract discriminative features from original features and finally RF efficiently classifies various types of attacks. The efficiency of the proposed model is evaluated against the well-known feature selection method (SVMo). To verify the versatility of the proposed architecture, two publicly available datasets have been used in the experiments.

In this paper, we proposed CVAEwRF, an effective deep learning method to automatically learn similarity among input features, provide data distribution in order to extract discriminative features from original features and finally efficiently classify various types of attacks for securing cyberspace. Applying various evaluation metrics, CVAEwRF demonstrates considerable improvement in the precision (mostly above 99%), regardless of the pattern of the applied dataset. These results show that the performance of anomaly detection is highly dependent on feature representation techniques.

Furthermore, the study shows for classes that have very few samples, the class imbalance stays a critical challenge. As it became evident for a class that does not have enough sample classifier is not capable of learning the pattern of class correctly and classifies samples of the skewed class randomly for both SVM and VAE. The mentioned class imbalance issue and a method to overcome the challenge will be addressed in our future work.

REFERENCES

[1] M. Monshizadeh, V. Khatri, B. G. Atli, R. Kantola, and Z. Yan, "Performance evaluation of a combined anomaly detection platform," *IEEE Access*, vol. 7, pp. 100964–100978, 2019.

[2] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, and Y. Yang, "AdVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection," *Knowl.-Based Syst.*, vol. 190, Feb. 2020, Art. no. 105187.

[3] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 3854–3861.

[4] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," *Sensors*, vol. 19, no. 11, p. 2528, Jun. 2019.

[5] Y. Yang, K. Zheng, B. Wu, Y. Yang, and X. Wang, "Network intrusion detection based on supervised adversarial variational auto-encoder with regularization," *IEEE Access*, vol. 8, pp. 42169–42184, 2020.

[6] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," *IEEE Access*, vol. 6, pp. 33353–33361, 2018.

[7] Y. Wei and K.-P. S.-M. Chow ja Yiu, "Insider threat detection using multi-autoencoder filtering and unsupervised learning," in *Advances in Digital Forensics*. New Delhi, India: Springer, 2020, pp. 273–290. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-56223-6, doi: 10.1007/978-3-030-56223-6_15.

[8] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: An improved siam-IDS for handling class imbalance in network-based intrusion detection systems," *Int. J. Speech Technol.*, vol. 51, no. 2, pp. 1133–1151, Feb. 2021.

[9] A. M. Liaw ja Wiener, "Classification and regression by randomForest," *R News*, vol. 2, pp. 18–22, Dec. 2007.

[10] S. Ronaghan. *The Mathematics of Decision Trees, Random Forest and Feature Importance in Scikit-Learn and Spark*. Accessed: Oct. 11, 2020. [Online]. Available: https://towardsdatascience.com/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

[11] X. Jing, Z. Yan, and W. Pedrycz, "Security data collection and data analytics in the Internet: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 586–618, 1st Quart., 2019.

[12] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016.

[13] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2014, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114

[15] Scikit Learning. *Confusion Matrix*. Accessed: Nov. 13, 2020. [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html?highlight=confusion%20matrix

[16] S. Loukas. *Multi-class Classification: Extracting Performance Metrics From The Confusion Matrix*. Accessed: Jan. 19, 2020. [Online]. Available: https://towardsdatascience.com/multi-class-classification-extracting-performance-metrics-from-the-confusion-matrix-b379b427a872

[17] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, Pittsburgh, PA, USA, 2006, pp. 233–240.

[18] L. XuKui, C. Wei, Z. Qianru, and W. Lifa, "Building auto-encoder intrusion detection system based on random forest feature selection," *Comput. Secur.*, vol. 95, 2020.

[19] Scikit-Learn. *Sklearn Metrics Log_Loss*. Accessed: Nov. 10, 2020. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html

[20] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4204–4212, May 2019.

[21] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, Sep. 2019.

[22] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, May 2012.

[23] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.

[24] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking," in *Proc. 6th Int. Conf. (Co-NEXT)*, 2010, pp. 1–12.

[25] W. Wang, X. Zhang, S. Gombault, and S. J. Knapskog, "Attribute normalization in network intrusion detection," in *Proc. 10th Int. Symp. Pervas. Syst., Algorithms, Netw.*, 2009, pp. 448–453.

[26] Scikit-Learn. *Machine Learning in Python*. Accessed: Nov. 10, 2020. [Online]. Available: https://scikit-learn.org/stable/

[27] TensorFlow. *TensorFlow Probability*. Accessed: Oct. 26, 2020. [Online]. Available: https://www.tensorflow.org/probability

[28] Scikit-Learn. *Glossary of Common Terms and API Elements*. Accessed: Oct. 28, 2020. [Online]. Available: https://scikit-learn.org/stable/glossary.html

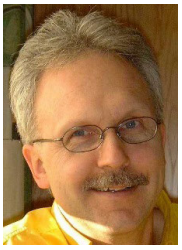
[29] Scikit-Learn. *Strategies to Scale Computationally: Bigger Data*. Accessed: Oct. 28, 2020. [Online]. Available: https://scikit-learn.org/0.15/modules/scaling_strategies.html#incremental-learning

[30] T. Dozat, "Incorporating Nesterov momentum into ADAM," Tech. Rep., 2016.

MEHRNOOSH MONSHIZADEH (Graduate Student Member, IEEE) is currently pursuing the Ph.D. degree with the Electrical School, Aalto University, Finland. She is currently working with Nokia Bell Labs as a Security Research Specialist. Her research interests include cloud security, mobile network security, the IoT security, and data analytics.

VIKRAMAJEET KHATRI received the M.Sc. degree in information technology from the Tampere University of Technology, Finland. He is currently working as a Security Specialist the Nokia Bell Labs. His research interests include intrusion detection, malware detection, the IoT security, and cloud security.

MARAH GAMDOU is currently pursuing the M.Sc. degree in computational science and applied mathematics with the Centralesupélec Engineering School, Paris-Saclay University, France. She is currently working as a Security Research Intern with Nokia Bell Labs. Her research interests include intrusion detection, generative deep learning models, and the IoT.



RAIMO KANTOLA (Member, IEEE) received the D.Tech. degree in computer science from the Helsinki University of Technology, Finland. He is currently a Professor in networking technology with the Department of Comnet, Aalto University, Finland. His research interests include SDN, customer edge switching, trust in networks, and cloud security.

ZHENG YAN received the D.Sc. degree in technology from the Helsinki University of Technology, Espoo, Finland, in 2007. She is currently a Professor with the School of Cyber Engineering, Xidian University, Xi'an, China, and a Visiting Professor and Finnish Academy Research Fellow with Aalto University, Helsinki, Finland. Her research interests include trust, security, privacy, and security-related data analytics. She received many awards, including Distinguished Inventor Award of Nokia, the Best Journal Paper Award issued by IEEE Communication Society Technical Committee on Big Data, and the Outstanding Associate Editor of 2017/2018 for IEEE Access. She also served as a general chair or a program chair for numerous international conferences, including IEEE TrustCom 2015 and IFIP Networking 2021. She is also the Founder Steering Committee Co-Chair of IEEE Blockchain Conference. She is also an Area Editor or an Associate Editor of *IEEE Network*, IEEE INTERNET OF THINGS JOURNAL, *Information Fusion*, *Information Sciences*, IEEE ACCESS, and *Journal of Network and Computer Applications*.