Applied Intelligence
https://doi.org/10.1007/s10489-021-02205-9

Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM

Adel Binbusayyis[1] · **Thavavel Vaiyapuri**[1]

Accepted: 7 January 2021

Abstract
With the rapid advancement in network technologies, the need for cybersecurity has gained increasing momentum in recent years. As a primary defense mechanism, an intrusion detection system (IDS) is expected to adapt and secure the computing infrastructures from the ever-changing sophisticated threat landscape. Many deep learning approaches have recently been proposed; however, these techniques face significant challenges in identifying all types of attacks, especially rare attacks due to network traffic imbalances and the lack of a sufficient number of abnormal traffic samples for model training. To overcome these shortcomings and improve detection performance, this paper presents an unsupervised deep learning approach for intrusion detection. Unlike the existing IDS model that extracts features and trains a classifier in two separate stages, a single-stage IDS approach that integrates a one-dimensional convolutional autoencoder (1D CAE) and a one-class support vector machine (OCSVM) as a classifier into a joint optimization framework is introduced in this paper for the first time. Using only the normal traffic samples, the approach simultaneously optimizes the 1D CAE for compact feature representation and the OCSVM for classification by defining a unified objective function combining reconstruction error with classification error. Thus, the generated compact feature representation has not only reconstruction ability but also discriminative ability for classification. An in-depth ablation analysis validates the design decisions and provides further insight of the proposed approach. An extensive set of experiments on two benchmark intrusion datasets, NSL-KDD and UNSW-NB15, demonstrates the generalization ability of the proposed model for unseen attacks and confirms it as a competitive approach over the recent state-of-the-art intrusion detection baselines. Overall, the obtained results emphasize that the proposed approach has potential to serve as a baseline for building an effective IDS.

Keywords Cybersecurity · Network intrusion detection · Deep learning · 1D convolutional autoencoder · Feature representation learning · One-class classifier · Joint optimization framework · OCSVM

1 Introduction

The rapidly evolving trends in networking technology have not only radically changed and enriched people's lifestyles, but have also dramatically transformed the business and social world with vast opportunities for economic growth and progress in all walks of life in all countries [1]. With this growing dependency on technology, cybercriminals are constantly renovating to stay a step ahead with their tactics in exploring sophisticated cyberattacks that could go beyond crippling our economy and cause loss of life [2, 3]. Accordingly, ensuring network security has become a more urgent requirement than ever before with the focus of attention on our society and organizations. In an effort to address this situation, security tools such as firewalls, antispam techniques, antiviruses, etc., are utilized to safeguard the business and social networks against cyberattacks. Undeniably, these tools offer first-line-of-defense security but fail to recognize new and sophisticated attacks. Under such circumstances, IDSs with the potential to adapt to the dynamically changing threat landscape are considered to be at the forefront of cybersecurity [4, 5]. In particular, IDS is effectively a device or software designed with the goal to monitor and analyze network traffic for any attempt that breaks security. Despite decades of significant progress, existing IDSs still remain incompetent in detecting unknown new attacks with a high detection rate (DR) and a

✉ Adel Binbusayyis
a.binbusayyis@psau.edu.sa

[1] College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia

Published online: 24 February 2021

low false alarm rate (FAR), which is a most critical issue in modern cyber ecosystems that needs to be addressed with utmost prudence [6, 7].

In light of the powerful capabilities of artificial intelligence (AI) technology, the interests of many researchers are piqued to design IDS models that capitalize on machine learning to realize intelligent detection of network attacks [8, 9]. Unfortunately, the IDS model based on traditional machine learning techniques exploits shallow architectures that liaise heavily on human-engineered network traffic features rather than the entire raw data [10]. This impedes the real network application of such models owing to the significant requirement of intensive human effort for feature engineering. Recently, the advent of deep learning has propelled AI to new heights and has also opened a promising path for automated feature extraction from large-scale raw data. Deep learning is a branch of machine learning based on artificial neural networks with multiple hidden layers. Analogous to machine learning techniques, deep learning also employs supervised and unsupervised learners to automatically extract complex features in a hierarchical manner and subsequently applies them to make decisions on unseen data [6]. The supervised deep learning networks (SDLN), such as the convolutional neural network (CNN) and recurrent neural network, infer the mapping function from labeled training data. On other hand, the unsupervised deep learning networks (UDLN), such as the deep belief network (DBN) and deep autoencoder (AE), aim to find the hidden structure of unlabeled data. In recent years, SDLNs have proven their potential strides in enhancing the intelligence of IDS to detect the rising cyberattacks. However, all these recent successes of SDLNs heavily depend on the availability of a sufficient quantity of accurately labeled training data. In a large-scale environment, labeling network traffic data will become tedious over time and may sometimes require domain knowledge from experts. Naturally, this may lead to error-prone data labels, especially when the network enters a malicious state. Intuitively, unsupervised deep learning networks are gaining a resurgence of interest and have become the current research hotspot in the field of intelligent intrusion detection with vital practical importance [11].

By virtue of this, there is recently a considerable number of works on the application of UDLN to the field of intrusion detection. Not withstanding that existing works are encouraging, some challenges remain in practically applying these UDLNs for intrusion detection. First, in a real network scenario, considering the system uncertainty and network topology complexity, it is evidently very demanding to collect abnormal traffic data in large sizes. Under such circumstances, the UDLN trained with insufficient abnormal traffic data will fail to learn more generalized features about attack instances and may face challenges in detecting the new attack vectors. Second, the real network traffic data are inherently imbalanced with more normal traffic than abnormal traffic. The UDLNs trained with such imbalanced data are biased toward normal traffic and face challenges in achieving a high detection rate for intrusions.

In light of the aforementioned data sparsity challenge, this paper presents an innovative IDS approach integrating the benefits of a one-class classifier and UDLN within a joint framework by defining a unified objective function that enables to gain an improved performance with regard to intrusion detection. In short, the major contributions of this work are highlighted below,

(a) For the first time, this work proposes a joint optimization framework to optimize CAE and one-class classifier simultaneously for feature representation learning and intrusion detection respectively.
(b) Unlike the existing works, this work combines the reconstruction and classification error to define an unified objective function to ensure that CAE learns the optimal feature representation and minimize the classification error to achieve higher accuracy for intrusion detection.
(c) To address the class imbalance problem, the proposed model is trained only with the normal samples in an unsupervised manner to improve the generalization ability of the proposed model.
(d) The ablation experiments and comparative analysis on benchmark intrusion datasets demonstrate the effectiveness of the proposed model against the state-of-the-art methods.

2 Literature review

The recent literature on cybersecurity reveals how the advancements in AI have led the intrusion detection problem to a more challenging level of study. At the same time, the corresponding computational solutions have also improved its level of efficiency. In particular, deep learning has shown state-of-the-art results in a multitude of applications in the field of network security. This has prompted an increasing number of researchers to turn their attention to utilizing deep learning techniques to improve the detection accuracy of IDS.

Correspondingly, a series of deep learning approaches have been proposed in the recent literature. Nevertheless, most of the reported state-of-the-art deep learning approaches adopt supervised models that have an insatiable appetite for labeled data that severely hinders their applicability in real network environments. To this end, the unsupervised deep learning models that are strong in extracting

and representing the hidden essential features from unlabeled data have gained new inspiration and substantial traction in recent years in the field of network security. For example, Alom et al [12] proposed to employ DBN for feature learning and enhance the accuracy of intrusion detection. Similarly, a study by Kang et al [13] presented an NIDS for the in-vehicular network security, leveraging the benefits of the unsupervised pretraining process for DBN. The results of the study revealed the model efficiency in detecting the intrusion activities. Additionally, a considerable number of studies have investigated the application of DBN in IDS design for feature extraction and have demonstrated its effects on the performance of intrusion detection [14, 15]. Nonetheless, the recent past studies have primarily focused on AE applications for developing efficient IDS due to its simple implementation and attractive computation cost. This paper will also focus on the application of autoencoders since they are more related to the work presented here.

Among them, a plethora of studies have attempted to develop variants of AE with enhanced discriminative and robust feature representation for intrusion detection. For instance, Hassan et al [16] contributed a variant of the sparse AE optimizing the hyperparameters to exhibit better capability in extracting useful features and classifying malicious attacks. Furthermore, the proposed reconstruction error function establishes a balance between feature representation and network regularization by applying a sparsity constraint in the output layer. Sheng et al [17] designed a new framework for IDS with a discriminative encoder and generator that works as a generative adversarial network during the training process and as an AE during the testing process to reconstruct the test samples. The reconstruction error with added extra loss empowers AE encoding ability to discriminate the malicious network activities. Aygun et al [18] enhanced the AE discriminative ability utilizing a stochastically determined threshold for reconstruction error to reach an improved accuracy compared to deterministic AE variants on NSL-KDD intrusion datasets. In the same manner, the authors in [19] presented a statistical analysis to extract more optimized and correlated features to improve the accuracy of AE. Similarly, an IDS is proposed in [20] employing an ensemble variant of AE to collectively differentiate the abnormal traffic patterns from normal traffic behavior. Moreover, Shone et al [21] recently introduced a nonsymmetric variant of a deep AE for unsupervised learning and achieved promising results

Another set of previous studies has suggested that the application of AE for extracting intrinsic features of network traffic data can improve the detection accuracy and performance of the classifier model in IDS. A work by M. Al-Qatf et al [22] developed a deep learning model based on a self-taught learning framework for intrusion detection. Here, the authors have used a sparse AE for feature extraction and an SVM classifier for interpreting the encoded features to identify intrusion. The model efficiency in binary and multiclass classification was investigated and compared with the state-of-the-art shallow machine learning models. The results showed the model capability in improving the training and testing time of the SVM with an improved accuracy rate compared to the previous approaches. Moreover, the authors in [23] applied a sparse AE model exploiting the concepts of self-taught learning to learn useful features for intrusion detection. In addition, they combined the original features with extracted features to improve the model generalization ability in recognizing the network attacks. Furthermore, in [24], a two-stage framework that combines a sparse AE with long short-term memory is investigated for building an efficient IDS. Here, the framework employs the sparse AE for learning effective feature representation and the LSTM model for classifying normal and malicious traffic. In a work by Shuaixin.T [25], the viability of combining stacked AEs and an SVM classifier configured with a piecewise radial basis function to improve the classification performance of the SVM for intrusion detection is examined. Similarly, the authors in [26] combined the advantage of stacked AEs with a CNN to considerably achieve the high-performance demand of network IDS. Likewise, the authors in [27] have studied the effectiveness of a stacked sparse AE model for extracting useful features of intrusion behavior. The study results indicated that the model can extract the more discriminative features and accelerate the detection process. Relatedly, the authors in [28] proposed new interesting online deep learning systems that apply an AE as function approximation in the Q-network of RL to achieve a higher detection accuracy rate for network intrusion detection.

From the above literature review, it is evident that despite the significant performance gain achieved with the application of AEs in IDS design, there is still room for improvement. The causes of weakness include the shortage of intrusion network traffic and the imbalance among the normal and abnormal network traffic. On these grounds, the existing approaches are prone to overfit and show poor generalization performance toward unseen cyberattacks. Thus, research on unsupervised deep learning approaches for IDS is still in its infancy in terms of development. Hence, the proposed research is expected to make a valuable contribution to the existing knowledge pool.

3 Background

3.1 Autoencoder

The autoencoder (AE) was introduced by [29] as an unsupervised neural network to learn robust feature representation by reconstructing the given input as network output. The basic structure of an AE consists of one hidden

layer. Here, the process between the input layer and the hidden layer is called the encoder. The task of the encoder is to map the given input data vector $X = (x_1, x_2, x_3, ...x_n)$ to a lower representation $H = (h_1, h_2, h_3, ...h_r)$ at the hidden layer, which is regarded as high-level features of the input. This is formulated as follows [30]:

$$H = f(WX + b) \tag{1}$$

Similarly, the process between the hidden layer and the output layer is called the decoder. In this process, the decoder aims to reconstruct the input data vector X from the lower hidden representation H, formulated as follows,

$$Z = g(W'H + b') \tag{2}$$

In the above equations, f and g are nonlinear activation functions such as sigmoid, tanh, and the rectified linear unit (ReLU) function. The W and b represent the weight and bias vector of the encoder, respectively. Likewise, W' and b' represent weight and bias vector of the decoder, respectively. These parameters of AE denoted as $\theta = \{W, W', b, b'\}$ are optimized during the training process by minimizing the reconstruction error defined by either using an L1 or L2 loss function.

3.2 One-class classifier

The one-class classifier (OCC) is a promising area of machine learning in which extensive research has been devoted to anomaly detection [31]. OCC aims merely at discriminating a class of interest from all other classes [32]. This class is labeled as normal. Whereas, all other classes that deviate from normal are termed as attacks. Although OCC seems to resemble binary classification, a significant difference lies in its training process, which is based on the assumption that only normal samples are available for training. Accordingly, OCC learns to derive a decision boundary only around normal samples as accurately as possible such that it encloses all normal samples while minimizing the probability of accepting attack samples. Since the OCC training process considers only normal samples, it is also known as learning in the absence of counterexamples.

After the training phase, at prediction time, the OCC uses the decision boundary to determine if the newly arriving sample belongs to the normal class or not. The new sample is classified as normal if it falls within the boundary. Conversely, if the new samples fall outside the boundary, they are rejected and treated as the attack class. An ideal OCC algorithm should not overfit based on the provided training samples; rather, it should generalize from training samples and gain a good discrimination ability to achieve a high detection rate on the attack class. However, since the OCC scenario does not have access to any a priori information about attacks, the training procedure may suffer from complications in making decisions on how to fit the boundary around the data in all directions without overfitting. Numerous solutions were proposed in the literature to address this problem.

4 Methodology

Figure 1 illustrates the proposed unsupervised deep learning approach for IDS. As shown in the figure, the proposed approach includes two essential components viz., AE for normal traffic feature representation learning and a one-class classifier for intrusion detection. The two subsections that follow elaborate the technical details of these two components. Subsequently, the designed unified objective function to achieve joint optimization in the proposed approach is presented.

4.1 Autoencoder for feature representation

This study adopts the convolutional AE (CAE) proposed by Masci et al [33] as an unsupervised feature representation learning method. It essentially combines the advantage of a CNN and an AE to attain strong feature representation. Compared to other variants of the AE, the CAE accomplishes strong feature representation considering the relationships among the features that are more appropriate for intrusion detection. Furthermore, CAE enables weight sharing among the inputs and ensures to preserve the spatial locality of the features. By doing so, the number of parameters to be trained is reduced. This in turn reduces the memory requirement and computational efficiency of the CAE. Thus, the CAE is regarded as special type of AE with a convolutional layer rather than a fully connected layer for the encoding process and a deconvolutional layer for the decoding process.

Taking inspiration from [34], this work adopts a 1D CAE with a hypothesis that the application of a 1D CAE will enable achieving further higher efficiency with the sequential form of network traffic data compared to a 2D CAE. Accordingly, the encoding process for the convolutional layer with feature filters maps 1D network traffic data X to produce a hidden representation with the k^{th} feature map and is represented as $H_k = f(X * W_k + b_k)$. Here, $*$ denotes the 1D convolution operation over the input vector X.

Moreover, to learn high-level network traffic features, a deep learning architecture of the 1D CAE is designed employing a series of convolutional layers. Under such a scenario, the hidden feature representation is obtained by

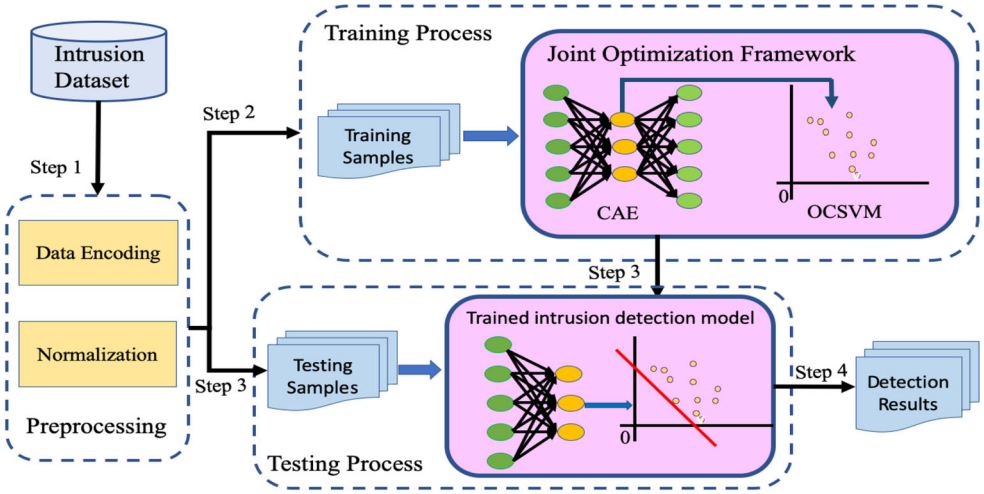


Fig. 1 Illustration of the proposed intrusion detection system architecture

recursive formulation, expressed as follows,

$$H^l = f\left(H^{l-1} * W^l + b^l\right) \tag{3}$$

where $l \in \{1, 2, 3, ..., L\}$ if L convolutional layers are employed and H^0 represents the original input vector X. As a result, H^L represents the final low-dimensional hidden representation of X.

Similarly, the designed 1D CAE employs a series of deconvolutional layers in the decoding process to reconstruct the original input network traffic data from H^L. This can be expressed with recursive formulation as follows,

$$Z^l = g\left(Z^{l-1} * W^{(L-l+1)T} + b'^l\right) \tag{4}$$

Thus, in the above equation, Z^1 represents the input of the first decoding layer which is H^L, the output of the last encoding layer. As result, the output of the last decoding layer Z^L represents the reconstruction of the original input vector X.

Moreover, in this work, the 1D CAE is designed to measure the mean squared error (MSE) as the reconstruction error between the original input network traffic vector X and the reconstructed network traffic vector Z. Therefore, the objective function of the designed 1D CAE is formulated as follows,

$$\begin{aligned} L_r(\theta) &= \min_{\theta} \frac{1}{2N}\|X - Z\|^2 \\ &= \min_{\theta} \frac{1}{2N}\|X - g(f(X))\|^2 \end{aligned} \tag{5}$$

4.2 One-class classifier for attack detection

As discussed in introduction section, collection of intrusion network traffic data in real practice is a major issue. Therefore, most of the available training dataset are imbalanced with a small amount of intrusion traffic data. Under such circumstances, the design of OCC discussed in Section 3.2 is well suited to resolve the problem using only normal traffic data without the requirement of intrusion traffic samples. In this direction, one-class support vector machine (OCSVM), an improved version of traditional SVM is extensively used in anomaly detection and has demonstrated promising results with imbalanced data [35, 36]. Taking inspiration from these literature, this work adopts OCSVM to address the network traffic imbalance and make the feature space more discriminative for intrusion detection. The basic idea of this approach is to deem all attack samples to lie on the origin and to use the advantages offered by the SVM to map the given training samples to a new feature space where they become linearly separable. Intuitively, they reduce the problem of fitting a nonlinear boundary around the data to a linear boundary or a hyperplane in the new feature space that separates all training samples from the origin with the maximum possible margin. The hyperplane is represented as follows for a given input vector X,

$$f(x) = \omega.\varphi(x) - \rho \tag{6}$$

Here, ω represents the weight coefficients, ρ is the distance from the origin to the hyperplane, and $\varphi(.)$ is a feature map obtained by applying certain kernel functions. The kernel function on any two samples from the input vector is defined as follows,

$$K(x_i, x_j) = \varphi(x_i).\varphi(x_j). \tag{7}$$

Here, the problem of finding an optimal hyperplane can be formulated as the following optimization problem,

$$\begin{aligned} &\min_{\omega,\rho,\xi} \quad \tfrac{1}{2}\|\omega\|^2 - \rho + \tfrac{1}{\nu N}\sum_{i=1}^{N}\xi_i \\ &s.t \quad \omega.\varphi(x) - \rho + \xi_i \geq 0,\ \xi_i \geq 0,\ \forall i \end{aligned} \tag{8}$$

Here, N represents the number of training samples, $\nu \in (0, 1]$ represents the regularization term that controls the fraction of outliers in the training set and ξ_i indicates the

slack variable used to model the classification error with respect to the i^{th} sample. Solving the above optimization problem using the Lagrange multiplier method, the decision function can be represented as

$$f(x) = sign\left(\sum_{i=1}^{N}\alpha_i K(x_i, x) - \rho\right), \\ = \begin{cases} 1, & \text{if } x \text{ belongs to the target class} \\ 0, & \text{otherwise} \end{cases} . \quad (9)$$

In the above equation, α_i is obtained by solving its dual form as follows,

$$\begin{aligned} &\max_{\alpha} \quad -\tfrac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j K(x_i, x_j) \\ &s.t \quad \sum_{i=1}^{N}\alpha_i = 1,\ 0 \le \alpha_i \le \tfrac{1}{\nu N},\ \forall i \end{aligned} \quad (10)$$

Having solved α_i, ρ can be calculated as follows by selecting any sample from the training set that satisfies $0 \le \alpha_j \le \frac{1}{\nu N}$ and that the sample is a support vector.

$$\rho = \sum_{j=1}^{N}\alpha_j K(x_i, x_j) \quad (11)$$

Thus, from the decision function of OCSVM, it is evident that OCSVM can effectively detect malicious activities solely with the knowledge of normal network traffic samples with an optimal hyperplane. However, the problem of finding an optimal hyperplane, the decision boundary of the OCSVM, is a pivotal issue and is strongly influenced by hyperparameter selection [37].

To address this issue and smoothly integrate the OCSVM within the proposed IDS architecture, this work applies the hinge loss function, $Hinge_s(z) = \max(0, s - z)$ [38], where s represents the position of the hinge point to penalize the samples classified with an insufficient margin. Accordingly, the optimization problem of the OCSVM given in (8) is transformed as follows by introducing the hinge loss function on the feature representation h_i^L learned from the original training samples [39] :

$$Lc(\nu, \gamma) = \min_{\omega,\rho,\nu} \frac{1}{2}\|\omega\|^2 - \rho + \frac{1}{\nu N}\sum_{i=1}^{N}\max(0, \rho - \omega.\varphi(h_i^L)) \quad (12)$$

The first and second terms in the above equation represent the structural risk loss function, and the last term denotes the empirical risk based on the hinge loss function. Furthermore, in the above equation, the mapping $\varphi(.)$ is usually implicit and indefinite. Therefore, the inner product of the mapped data is generally specified by the kernel function in practice. The most commonly used kernel functions include linear, sigmoid, polynomial and radial basis (RBF). Here, the RBF that usually leads to better performance is chosen. The RBF is defined as follows,

$$K(x_1, x_2) = \exp(-\gamma.\|x_1 - x_2\|^2) \quad (13)$$

Here, the hyperparameter of the kernel function has to be tuned such that $K(h_i^L, h_j^L)$ always produce small values when both h_i^L and h_i^L are not normal traffic samples. On other hand, the larger or very much smaller values of the kernel parameter may lead to underfitting or overfitting, respectively.

Similarly, setting the hyperparameter ν in (12) to 0 will make the last term vanish, meaning that the OCSVM will be forced to find a hyperplane that separates all training samples as normal from the origin. Conversely, if ν is set to 1, then the OCSVM will tend to find a hyperplane that rejects all training samples as attacks. Intuitively, the hyperparameter ν has to be tuned to avoid the distortion of the hyperplane (decision boundary) by outlier data and improve the model generalization ability for unseen data.

Taking into consideration, the sensitivity to false positive rate and generalization ability of OCSVM, this work focuses to tune the hyperparameters ν and γ with regard to the most robust discriminative features learnt by CAE. This is achieved by designing a new objective function that enables to integrate 1D CAE for feature representation learning and OCSVM for classification within a joint optimization framework. The subsection following briefs how the hyperparameters are tuned during model training process in an unsupervised manner defining a new objective function.

4.3 Designed objective function

All the existing AE IDS models perform feature representation learning and classification by learning independently through pretraining and the fine-tuning process without joint optimization [6]. In that case, the learned features do not guarantee strong discriminative ability for the intrusion detection task. To circumvent this problem, our work intends to combine the reconstruction loss term in (5) with the structural and empirical risk term of the classifier given in (12) and define the following objective function

$$L = L_r + L_c \quad (14)$$

This objective function clearly reveals that our work aims to guide the proposed approach to learn strong feature representation for an improved effective intrusion detection by integrating the feature representation and classification process into a joint optimization framework. In doing so, the proposed approach reduces the reconstruction loss and at the same time ensures that the classification hyperplane margin is maximized for improving the detection accuracy of the

proposed approach. Algorithm 1 summarizes the working procedure of the proposed approach.

Algorithm 1 Algorithm of the proposed approach.

Input: X - Training Set
Initialization:

1. CAE parameter θ using Xavier algorithm
2. OCSVM parameter ν using grid search algorithm

Procedure:

repeat

Feature representation Learning:

1. $H \leftarrow$ Feature representation using (3)
2. $Z \leftarrow$ Reconstructed Input using (4)
3. $L_r \leftarrow$ Reconstruction Loss using (5)

Classifier Learning:

1. Obtain optimal kernel parameterγ using grid search
2. Transform H to $kernelspace$ applying (13)
3. Find the hyperplane computing α_i as in (10) and ρ as in (11)

Optimization :

1. compute the gradient minimizing the objective function in (12)
2. Update model parameter θ

until <Convergence of θ >

5 Experimental setup

This section first describes the experimental datasets and then details the methods used for preprocessing the datasets. Subsequently, the structure of the CAE network is described, followed by the training details. Finally, the implementation details and the metrics used for experimental evaluation are presented.

5.1 Datasets

A number of datasets are available publicly for IDS research evaluation. Nonetheless, these datasets suffer from absences of traffic diversity and lack a sufficient number of sophisticated attack styles. Therefore, in order to conduct a fair and effective evaluation of the proposed model, an old benchmark NSL-KDD dataset and a new contemporary UNSW-NB15 dataset are considered in this work. A brief description of these two intrusion datasets is given below,

5.1.1 NSL-KDD dataset

The NSL-KDD dataset is an improved version of the KDD'99 dataset, presented by Tavallaee et in 2009, that resolves the redundancy in the KDD '99 dataset [40]. This dataset contains an optimal ratio of 125,973 training samples to 22,543 testing samples. Thus, NSL-KDD is regarded as one of the most valuable benchmark resources in the field cybersecurity research for IDS evaluation. Each sample in NSL-KDD contains 41 features and 1 class label to characterize whether the network traffic is normal or belongs to the attack category. The distributions of normal traffic samples in the training and testing sets with regard to attacks are given in Table 1.

5.1.2 UNSW-NB15 dataset

The UNSW-NB15 is a modernized dataset recently developed by ACCS with a hybrid representation of real normal and synthesized contemporary attack behaviors from network traffic flow [41]. This dataset includes 9 families of attacks, namely, DoS, Analysis, Generic, Fuzzers, Backdoors, Exploits, Shellcode, Reconnaissance, and Worms. The dataset consists of 175,341 training samples and 82,332 testing samples, each characterized with 42 features and a class label to discriminate the network traffic as normal or malicious activities. The distributions of samples against normal and attack classes is shown in Table 2.

5.2 Data preprocessing

Data preprocessing is essentially crucial for providing quality input for model training and for boosting the detection ability of the IDS. It includes two main operations, namely, data encoding and normalization.

(a) *Data Encoding:* In this work, the label encoding method is used to map all nonnumeric or nominal features to numeric values. This method maps a nominal feature with C different values to an integer in the range of 0 to C-1. For example, the NSL-KDD dataset includes three nominal features, namely, protocol_type, service_type, and TCP status flag with 3, 70 and 11 distinct nominal values, respectively. After label encoding, the feature protocol_type with three values is mapped as follows: tcp:0, udp:1 and icmp:2.

(b) *Normalization:* Generally, the machine learning algorithms are biased by input features with large numeric values. To combat this effect, min-max normalization is applied to adjust the value range of all input features within the range [0,1].

Table 1 Data distributions in NSL-KDD

Class	Training set	Testing set
Normal	67,343	9,710
Attack	58,630	12,833
Total	125,973	22,543

Table 2 Data distributions in UNSW-NB15

Class	Training set	Testing set
Normal	56,000	37,000
Attack	119,341	45,332
Total	175,341	82,332

5.3 CAE model configuration

Notably, the structure of the deep learning CAE network has a great impact on the classification performance. Therefore, in this work, the configuration adopted for 1D CAE was determined by conducting a range of experiments with various structural combinations. Figure 2 illustrates the structure used for the 1D CAE network. It consists of three convolutional layers with an ReLU as the activation function on the encoder side. The number of filters in these layers is reduced by half from low-level to high-level convolutional layers. For example, the first convolutional layer consists of 32 filters, the second layer consists of 16 layers and last with 8. Nonetheless, the size of kernels is kept the same for all convolution layers and is set to 1×3. This pyramid architecture not only reduces the number of trainable parameters but also enables learning the most essential features from the input network traffic by eliminating the redundant and irrelevant features.

A max pooling layer with a pool size of 2 is added after each convolutional layer except the last to extract the most essential features from the network traffic data and improve the representation capability of the model. With two max pooling layers, the dimension of the input network traffic flow is reduced to 10. Furthermore, all convolutional layers are followed by a batch normalization layer to stabilize the training process and accelerate the network convergence. In addition, to avoid model overfitting and improve computational efficiency, a dropout layer is added for regularization with a rate of 0.2 after each max pooling layer and as the last layer of the encoder. A mirrored encoder structure with deconvolution and upsampling operations is employed on the decoder side to reconstruct the original network traffic input of dimension 1×41 from 10 low-dimensional high-level features resulting from the encoder.

5.4 Training process

Fundamentally, the training process plays a key role in optimizing the hyperparameters of a model and directly influences the performance of the model. Therefore, in this work, model training was conducted with utmost care and meticulous planning to learn the robust feature representation that can ensure a strong reconstruction ability for the input network traffic and a discriminative ability of the OCSVM. Accordingly, the training process uses the minibatch gradient descent optimization algorithm to minimize the reconstruction loss and classifier loss jointly. In this approach, the loss computed over a batch of training samples is used in backpropagation to balance the tradeoff between the robustness and efficiency of the proposed deep network architecture. Moreover, the Adam update rule [42] with a minibatch size of 32 and a learning rate of 0.001 is adopted considering its fast convergence rate and fewer memory requirements for computing the gradients of trainable parameters and to achieve an optimal network architecture. To obtain stabilized results, the training process is terminated when the number of epochs exceeds 15 or the loss value of the deep network falls below the threshold value of 0.005. Notably, to keep the backpropagated gradient values and activation values within a reasonable range, all the trainable parameters of the CAE are initialized using the Xavier algorithm [43]. Additionally, the grid search (GS) algorithm is adopted to initialize the OCSVM parameter ν over a subset of values $\{0.1, 0.05, 0.07\}$ on the given training samples. For initializing the parameter γ, GS works on given training samples and the user provided subset of three values

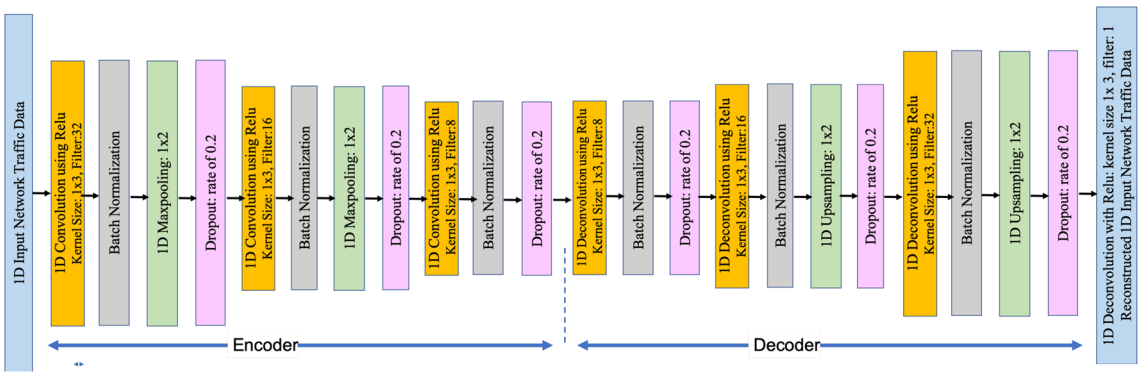


Fig. 2 Structure of the developed Convolutional Autoencoder Network

$\{a, b, c\}$. However, later during each iteration of the training process, the parameter γ is tuned for the learned feature representation using grid search with a new subset of four values that are determined based on their initial optimal value. For example, if the outcome of GS is 50 for the user given subset of values: 50, 100, 150, then the computed new subset will be as follows 10, 20, 30, 40.

Under the above settings, the proposed network architecture is trained with normal samples in the training datasets. During each iteration of the training process, as the normal samples in the training dataset flow through the CAE network, the essential features are extracted automatically with a reduction in dimension to a size of 10 x 8 through three encoding layers. Then, the original input is reconstructed from the extracted features through three decoding layers. Next, the features extracted by the encoder part of the CAE are mapped to the kernel space using an RBF to train the OCSVM. At the end of each iteration, the reconstruction loss and classifier loss are computed to update the trainable parameter and obtain an optimal network architecture.

5.5 Implementation details

All the experiments are conducted on a personal computer with the specifications as follows: an Intel Core i7-8565H @ 1.8GHz with 128 GB RAM and the Windows 10 operating system. The proposed model is implemented in the Jupyter development environment using Python 3 as the programming language. More specifically, the Python libraries, Keras and TensorFlow are used to implement various deep learning tasks [44]. Additionally, the Python Scikit-learn library is used to implement various evaluation measures and data preprocessing tasks.

5.6 Evaluation metrics

The effectiveness of the proposed IDS approach is measured by analyzing four evaluation metrics that are most commonly used in the field of intrusion detection. The relevant definitions of these four metrics are as follows,

(a) *Accuracy (ACC):* Measures the proportion of network traffic flows that are correctly classified and is computed as follows,

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

(b) *Detection rate (DR):* Also called Recall or Sensitivity, measures the proportions of intrusion traffic flow that are correctly classified as given below,

$$DR = \frac{TP}{TP + FN} \tag{16}$$

(c) *F1-measure (F1) :* Also termed the F1-Score, is considered a more effective measure than accuracy to evaluate the performance of an intrusion detection model, especially for imbalanced datasets. It is an harmonic average of the detection rate and precision as follows

$$F1 = \frac{2 \times (DR \times Precision)}{DR + Precision} \tag{17}$$

Here, precision measures the proportions of detected intrusion traffic that are actually correct. It is expressed as follows,

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

(d) *False alarm rate (FAR) :* Also termed the false positive rate, measures the proportion of normal network traffic flows that are incorrectly classified. It is computed as follows,

$$FAR = \frac{FP}{FP + TN} \tag{19}$$

6 Experimental results and discussion

This section describes the three sets of analyses designed to demonstrate the supremacy of the proposed approach. In particular, these experiments aim to achieve the following

(a) Validate the design decision of the proposed approach on the benchmark dataset NSL-KDD
(b) Investigate whether the performance of the proposed approach is stable with the recent intrusion dataset
(c) Compare the performance of the proposed approach with recent state-of-the-art methods published in literature.

6.1 Ablation analysis

At the outset, two sets of ablation experiments are conducted to analyze the design decision of the proposed approach and CAE architecture correspondingly on the standard benchmark intrusion dataset, NSK-KDD, in terms of ACC, DR and FAR. The subsections below describe these experiments in detail.

6.1.1 Experiment 1

As a first step, the design decision of the proposed approach is analyzed investigating how different components in the proposed approach can contribute jointly to the overall performance. For this purpose, an analysis study is

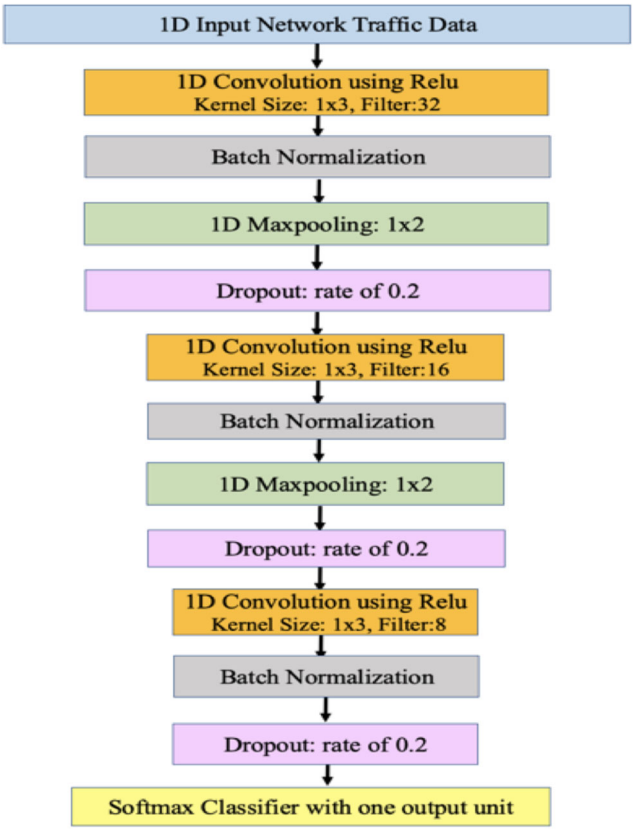


Fig. 3 Structure of the CAE+Softmax variant

conducted developing three variants of proposed approach as follows,

(a) **OC:** This variant is created by removing the CAE component to evaluate the effectiveness of the proposed approach for feature representation learning.
(b) **CAE+softmax:** This variant is created by replacing the decoder of the trained CAE and OCSVM components with a softmax layer as shown in Fig. 3 to evaluate the effectiveness of the proposed approach for one-class unsupervised classification.
(c) **CAE+OC:** This version indeed is developed to evaluate the effectiveness of the joint optimization framework with the CAE and OCSVM. To this end, the CAE is first trained to learn the essential feature representation. Then, the features extracted by the CAE are used to train the OCSVM for classification.

For a fair comparison, the ablation experiments are conducted using the same parameters and environmental setup as the proposed model, and the results are reported in Table 3. Observation of these ablation results demonstrates the significance and relevance of all components in the proposed approach against the achieved performance benefits. Particularly, it can be seen that the baseline variant OC induces a high FAR. This illustrates the significant role of the CAE in learning the most essential high-level features from the normal network traffic flow to deliver an improved performance in terms of DR, FAR and ACC.

Similarly, it is obvious from the experimental results of the variant, CAE+softmax, that without the OC classifier, the overall performance drops significantly. This clearly reflects that the OC classifier is a more efficient component that uses its kernel tricks to contribute to a compact representation of the normal samples in the proposed approach, thereby ensuring the overall improvement in the intrusion detection task.

Furthermore, since timeliness is another critical metric required in modern IDS, the training and testing time of the proposed method is compared on NSL-KDD datasets. To establish a fair comparison, all computation time are calculated under the same operating environment as discussed in Section 5.5. The results of this comparison are shown in Table 3. From these results, it can be seen that the proposed approach takes longer training time of 1123s than other ablations to discover the optimal hyperplane that can effectively compact the features representation learnt by CAE. This might be due to the joint framework adopted to train both CAE and OCSVM simultaneously to minimize the reconstruction and classifier loss. Nevertheless, it is easy to observe that the proposed method ranks second among all the ablations and takes 0.35ms to detect an intrusion traffic instance during the testing period. In turn, this implies that in comparison to offline training time, the test time is more crucial element for an IDS and the proposed method is well-designed to reduce the detection time while

Table 3 Ablation analysis results on NSL-KDD dataset for different variants of proposed approach

Variants	Training set					Testing set				
	DR	FAR	ACC	F1	Time(s)	DR	FAR	ACC	F1	Time(s)
OC	95.8	6.28	95.75	95.45	753	87.2	12.2	87.11	88.51	0.0019
CAE+Softmax	94.77	5.99	93.82	94.17	628	84.39	11.57	86.14	87.39	0.000311
CAE+OC	97.76	3.81	97.48	98.31	609	92.45	7.59	90.42	91.66	0.000427
Proposed	99.54	1.14	98.45	99.22	1123	97.11	2.43	91.58	92.87	0.00035

maintaining the detection accuracy with optimal number of batch normalization, maxpooling and dropout layer to demonstrate comparably better detection time efficiency.

Overall, it can be observed that the proposed approach excels in all three variants in terms of all three evaluation metrics, which confirms the contribution of the joint optimization framework to the success of the proposed approach. The outcome is attributed to the ability of the approach to effectively leverage the benefits of the CAE and OC classifier for representation learning and the one-class classification, respectively, through simultaneous optimization of reconstruction loss and classifier loss.

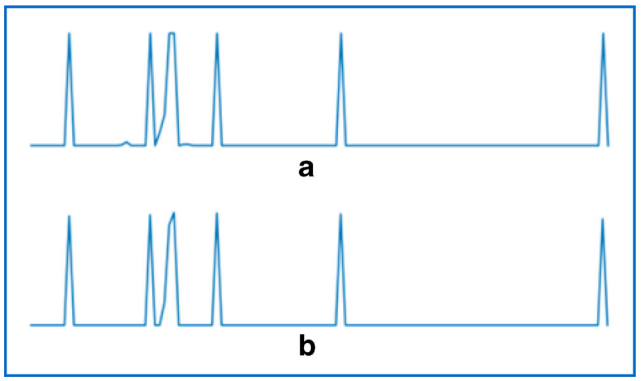


Fig. 4 Illustration of the reconstructed network traffic data by the CAE component when the hidden dimension is 10

6.1.2 Experiement 2

This set of experiments aims to study the relations between intrusion detection performance and other factors, such as the hidden state dimension and number of filters, exploring different network design for CAE.

Successively, a range of experiments were conducted varying the final hidden state dimension to 20, 10 and 5 but keeping other factors fixed. The results reported in Table 4 clearly demonstrate the impact of the hidden state dimension on the performance of the proposed approach. Although the hidden state dimension at 20 delivered higher DR and ACC values on the training set than its counterparts, it fails to maintain a higher performance on the testing set. On the other hand, the hidden state dimension of 10 displays higher performance on both training and testing sets. In addition, Fig. 4 illustrates the ability of CAE to learn the robust feature representation that can reconstruct the original input with a small variation when the hidden state dimension is 10. Therefore, in further experiments, the CAE network of the proposed model was designed with the feature representation of dimension 10.

Next, considering the claim in the previous literature [45] that the number of filters has a great impact on model performance, a series of ablation experiments were conducted to determine the optimal number of filters for the CAE component in the proposed approach. In this analysis, the number of filters in the first convolutional layer was varied among 64, 32, 16 and 8. However, due to successive feature downsampling, the number of filters is reduced by half for each successive convolutional layer to better extract the essential features through the convolution process. Table 5 summarizing the corresponding results evidently indicates the improvement in overall performance with the increasing number of filters, which is also in accordance to the claim in the literature. Nonetheless, it can be observed that this increase in performance is not significant in contrast to the increase in the computational complexity when the number of filters in the first convolution layer is 64 and is reduced by half successively. Therefore, the number of filters for the first convolution layer in the CAE was chosen as 32 and was reduced by half for each successive convolutional layer in our subsequent experiments.

6.2 Performance analysis

In the literature, it is stated that the change of datasets considerably affects and varies the performance of the detection process [53]. Accordingly, to investigate the stable performance of the proposed approach on different datasets, this experiment is conducted choosing a most recent benchmark dataset, UNSW-NB15, that includes many new modern attack styles.

The confusion matrix delivered by the proposed approach on UNSW-NB15 training and testing datasets is shown in Fig. 5. The evaluation metrics computed using these confusion matrices are presented in Fig. 6. The figure demonstrates that the proposed approach is very effective

Table 4 Ablation analysis results on NSL-KDD dataset for different hidden state dimensions in the CAE component of proposed approach

Hidden state dimension	Training set				Testing set			
	DR	FAR	ACC	F1	DR	FAR	ACC	F1
20	99.72	2.19	99.03	98.83	96.63	.3.45	88.98	90.22
10	99.54	1.14	98.45	99.22	97.11	2.43	91.58	92.87
5	90.25	5.83	93.35	92.66	85.14	9.14	84.94	86.55

Table 5 Ablation analysis results on NSL-KDD dataset for different number of filters in the CAE component of proposed approach

Number of filters	Training set				Testing set			
	DR	FAR	ACC	F1	DR	FAR	ACC	F1
64	99.78	1.65	99.93	99.86	97.32	2.26	92.04	92.9
32	99.54	1.14	98.45	99.22	97.11	2.43	91.58	92.87
16	97.48	3.19	97.72	97.55	91.69	3.89	89.21	90.81
8	95.8	3.83	96.34	96.06	88.04	4.66	87.33	87.68

in achieving a DR of 97.7, FAR of 5.55, ACC of 96.7 and F1 of 97.57 on the training dataset. Comparably, a DR of 97.8, FAR of 1.8, ACC of 97.6 and F1 of 98.1 on the testing dataset clearly reveals the efficacy of the proposed approach to generalize even a complex dataset such as UNSW-NB15, and at the same time, the findings confirm that the proposed approach is very competitive for modern attack detection.

It is can be noted that, similar to the results on NSL-KDD, the performance improvement of the proposed approach on the UNSW-NB15 dataset also remains at a promising level. This consistent performance of the proposed approach is evidently attributed to the joint optimization of feature representation and classification learning for the intrusion detection task.

6.3 Comparative analysis

The effectiveness of the proposed approach is further highlighted by a comparison with recent and relevant deep learning approaches from the literature of intrusion detection. Since it is impractical to compare all latest approaches, only those approaches that have used both NSL-KDD and UNSW-NB15 datasets are considered to provide a meaningful comparison. Additionally, the results provided in their published papers are used to maintain a fair comparison and the results of this comparison are presented in Table 6. Here, for clarity purposes, the highest score is highlighted in bold for each metric on both NSL-KDD and UNSW-NB15 datasets.

Now, observing the results on NSL-KDD, it can be realized that the proposed approach outperforms all the recent IDS approaches for all metrics except for the model introduced in [52] with the few-shot supervised learning approach (FSL-IDS) in terms of accuracy. However, while the FSL-IDS model shows slightly higher accuracy than the proposed approach, its probability for FAR is the worst at 7.21%. This indicates that the proposed approach is competitively effective in generating a lower FAR rate than all other recent approaches when applied to intrusion detection. As FAR is one of the extremely important metrics that should be kept low for an ideal IDS model, it is evident that the proposed approach is well designed to exhibit the best performance benefits with regard to all metrics.

Similarly, comparing the results on UNSW-NB15 presented in Table 6, it can be seen that proposed approach displays very competitive results compared to all other recent IDS approaches under study. Nevertheless, one exception is observed with the multilayered echo state machine (ML-ESM) model proposed in [51]. Comparing the performance of ML-ESM on NSL-KDD and UNSW-NB15, it can be observed that ML-ESM presents the highest performance on the UNSW-NB15 dataset but fails to deliver consistent

Detection Results →	Normal	Attack
Normal	52891	3109
Attack	2672	116669

a

Detection Results →	Normal	Attack
Normal	36332	668
Attack	1090	44242

b

Fig. 5 Confusion matrix of the proposed approach on UNSW-NB15. **a** Training Set. **b** Testing Set

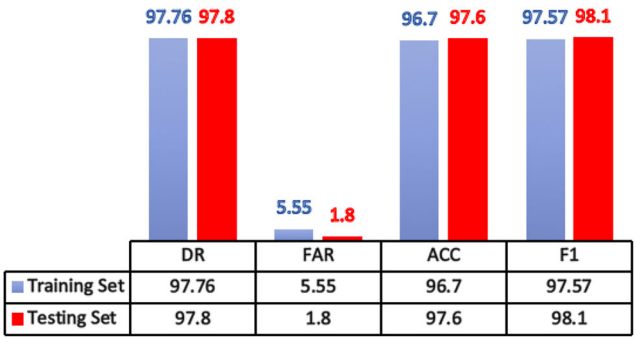


Fig. 6 Performance analysis of the proposed approach on UNSW-NB15

Table 6 Comparative analysis of proposed approach against recent IDS approaches

Recent IDS approaches	NSL-KDD dataset				UNSW-NB-15 dataset			
	DR	FAR	ACC	F1	DR	FAR	ACC	F1
ICVAE-DNN [46]	77.43	2.74	85.97	86.27	95.68	19.01	89.08	90.61
SAVAER [47]	95.98	4.70	89.36	90.08	91.94	5.67	93.01	93.54
DNN with one Layer [48]	96.90	NA*	80.1	80.7	72.5	NA	78.4	82.0
MDPCA +DBN [49]	70.51	2.62	82.08	81.75	96.22	17.15	90.21	91.54
Hybrid ML [50]	86.8	11.7	85.79	NA	91.3	8.9	91.27	NA
Multilayer ESM [51]	83.0	3.3	NA	NA	**98**	**5.10**	NA	NA
Few shot Learning [52]	92.25	7.21	**92.34**	92.26	NA	8.01	92.11	NA
Proposed CAE+OCSVM	**97.11**	**2.43**	91.58	**92.87**	96.49	5.51	**94.28**	**95.06**

*denotes that the corresponding metric is not available/provided in the published paper

results on the NSL-KDD dataset. This reveals its setback in stability with regard to different datasets. Moreover, this confirms the superiority of proposed approach against other recent approaches regardless of data distributions. The reason is possibly attributed to the introduced joint optimization framework that enables the CAE to generate a feature representation with the potential ability for not only reconstruction but also for enhancing the classifier discriminative ability for the intrusion detection task

In summary, it can be concluded that the superior performance of the proposed approach demonstrates that it has great potential to be a used as promising tool for intrusion detection.

7 Conclusion

In this research, a novel unsupervised approach for intrusion detection combining the benefits of deep learning and an OC classifier is introduced and discussed. To the best of our knowledge, the proposed approach is the first attempt to integrate the 1D CAE and OCSVM using a joint optimization framework. The novelty of the proposed approach is twofold, as briefly described below.

(a) It bridges the gap between the feature representation and classifier learning that exists in traditional IDS approaches by combining both reconstruction loss and classification loss into a unified objective function.
(b) In contrast to traditional IDS approaches, the proposed approach simultaneously learns the robust feature representation from network traffic data and optimizes the OC classifier competitively to gain superior detection accuracy for intrusion.

The effectiveness of the proposed approach is evaluated on two benchmark intrusion datasets, NSL-KDD and UNSW-NB15, in terms of DR, FAR, ACC and F1. The comprehensive ablation analysis results not only confirm the design decision rationale of the proposed approach but also demonstrate that the approach can show considerable performance improvement for effective intrusion detection. Furthermore, the experimental results on UNSW-NB15 have proven the potential efficacy of the proposed approach with conformance to our initial discussion, namely, that simultaneous optimization of feature representation and classifier learning in an unsupervised manner serves as an effective approach in detecting unseen modern attack styles. The comparative analysis and discussion also evidently signify the advantage of the joint optimization framework on the generalization ability of the proposed approach and indicate that the proposed approach is a competitive candidate for intrusion detection among the latest state-of-the-art IDS approaches. In conclusion, it is anticipated that the proposed approach will serve as a future benchmark for building a promising tool to safeguard the network environment against intrusion detection.

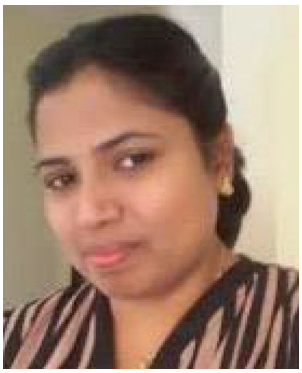
References

1. Kagermann H (2015) Change through digitization—value creation in the age of industry 4.0. In: Management of permanent change. Springer, pp 23–45
2. Kamasa J (2020) Securing future 5g-networks. Policy Perspectives 8:4
3. Bartock M, Cichonski J, Souppaya M (2020) 5g cybersecurity: preparing a secure evolution to 5g. Technical report, National Institute of Standards and Technology
4. Binbusayyis A, Vaiyapuri T (2019) Identifying and benchmarking key features for cyber intrusion detection: an ensemble approach. IEEE Access 7:106495–106513
5. Benmessahel I, Xie K, Chellal M (2018) A new evolutionary neural networks based on intrusion detection systems using multiverse optimization. Appl Intell 48(8):2315–2327

6. Aldweesh A, Derhab A, Emam AZ (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl-Based Syst 189: 105124
7. Binbusayyis A, Vaiyapuri T (2020) Comprehensive analysis and recommendation of feature evaluation measures for intrusion detection. Heliyon 6(7):e04262
8. Truong TC, Zelinka I, Plucar J, Čandík M, Šulc V (2020) Artificial intelligence and cybersecurity: past, presence, and future. In: Artificial intelligence and evolutionary computations in engineering systems. Springer, pp 351–363
9. Kaja N, Shaout A, Ma D (2019) An intelligent intrusion detection system. Appl Intell 49(9):3235–3247
10. Maza S, Touahria M (2019) Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms. Appl Intell 49(12):4237–4257
11. Aleesa AM, Zaidan BB, Zaidan AA, Sahar NM (2020) Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. Neural Comput Appl 32(14):9827–9858
12. Alom MZ, Bontupalli VR, Taha TM (2015) Intrusion detection using deep belief networks. In: 2015 National aerospace and electronics conference (NAECON). IEEE, pp 339–344
13. Kang M-J, Kang J-W (2016) Intrusion detection system using deep neural network for in-vehicle network security. PloS One 11(6):e0155781
14. Ni G, Gao L, Gao Q, Wang H (2014) An intrusion detection model based on deep belief networks. In: 2014 second international conference on advanced cloud and big data. IEEE, pp 247–252
15. Zhang X, Chen J (2017) Deep learning based intelligent intrusion detection. In: 2017 IEEE 9th international conference on communication software and networks (ICCSN). IEEE, pp 1133–1137
16. Musafer H, Abuzneid A, Faezipour M, Mahmood A (2020) An enhanced design of sparse autoencoder for latent features extraction based on trigonometric simplexes for network intrusion detection systems. Electronics 9(2):259
17. Mao S, Guo J, Li Z (2019) Discriminative autoencoding framework for simple and efficient anomaly detection. IEEE Access 7:140618–140630
18. Can Aygun R, Gokhan Yavuz A (2017) Network anomaly detection with stochastically improved autoencoder based models. In: 2017 IEEE 4th international conference on cyber security and cloud computing (CSCloud). IEEE, pp 193–198
19. Ieracitano C, Adeel A, Morabito FC, Hussain A (2020) A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. Neurocomputing 387:51–62
20. Mirsky Y, Doitshman T, Elovici Y, Shabtai A (2018) Kitsune: an ensemble of autoencoders for online network intrusion detection. arXiv:1802.09089
21. Shone N, Ngoc TN, Vu DP, Qi S (2018) A deep learning approach to network intrusion detection. IEEE Trans Emerg Topics Comput Intell 2(1):41–50
22. Al-Qatf M, Yu L, Al-Habib M, Al-Sabahi K (2018) Deep learning approach combining sparse autoencoder with svm for network intrusion detection. IEEE Access 6:52843–52856
23. Qureshi AS, Khan A, Shamim N, Durad MH (2019) Intrusion detection using deep sparse auto-encoder and self-taught learning. Neural Comput Applic 32:1–13
24. Kherlenchimeg Z, Nakaya N (2020) A deep learning approach based on sparse autoencoder with long short-term memory for network intrusion detection. IEEJ Trans Electron Inform Syst 140(6):592–599
25. Shuaixin T (2020) An intrusion detection method based on stacked autoencoder and support vector machine. In: J phys conf series, vol 1453, pp 1–17
26. Yu Y, Long J, Cai Z (2017) Network intrusion detection through stacking dilated convolutional autoencoders. Secur Commun Netw, 2017
27. Yan B, Han G (2018) Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE Access 6:41238–41248
28. Kim C, Park JS (2019) Designing online network intrusion detection using deep auto-encoder q-learning. Comput Electric Eng 79:106460
29. Rumelhart DE, Hinton GE, Williams RJ (1988) Learning representations by back-propagating errors neurocomputing: foundations of research
30. Wang Y, Yao H, Zhao S (2016) Auto-encoder based dimensionality reduction. Neurocomputing 184:232–242
31. Tajoddin A, Abadi M (2019) Ramd: registry-based anomaly malware detection using one-class ensemble classifiers. Appl Intell 49 (7):2641–2658
32. Khan SS, Madden MG (2014) One-class classification: taxonomy of study and review of techniques. Knowl Eng Rev 29(3): 345–374
33. Masci J, Meier U, Cireşan D, Schmidhuber J (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. In: International conference on artificial neural networks. Springer, pp 52–59
34. Chen S, Yu J, Wang S (2020) One-dimensional convolutional auto-encoder-based feature learning for fault diagnosis of multivariate processes. J Process Control 87:54–67
35. Tan FHS, Park JR, Jung K, Lee JS, Kang D-K (2020) Cascade of one class classifiers for water level anomaly detection. Electronics 9(6):1012
36. Tian Y, Mirzabagheri M, Tirandazi P, Mojtaba S, Bamakan H (2020) A non-convex semi-supervised approach to opinion spam detection by ramp-one class svm. Inform Process Manag 57(6):102381
37. Wang S, Liu Q, En Z, Porikli F, Yin J (2018) Hyperparameter selection of one-class support vector machine by self-adaptive data shifting. Pattern Recognit 74:198–211
38. Xiao Y, Wang H, Xu W (2017) Ramp loss based robust one-class svm. Pattern Recogn Lett 85:15–20
39. Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC, et al. (1999) Estimating the support of a high-dimensional distribution. Technical Report MSR-t R-99–87 Microsoft Research (MSR)
40. Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE, pp 1–6
41. Moustafa N, Slay J (2015) Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 military communications and information systems conference (MilCIS). IEEE, pp 1–6
42. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
43. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
44. Géron A (2019) Hands-on machine learning with Scikit-Learn, Keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media

45. Agrawal A, Mittal N (2020) Using cnn for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. Visual Comput 36(2):405–412
46. Yang Y, Zheng K, Wu C, Yang Y (2019) Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors 19(11):2528
47. Yang Y, Zheng K, Wu B, Yang Y, Wang X (2020) Network intrusion detection based on supervised adversarial variational auto-encoder with regularization. IEEE Access 8:42169–42184
48. Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S (2019) Deep learning approach for intelligent intrusion detection system. IEEE Access 7:41525–41550
49. Yang Y, Zheng K, Wu C, Niu X, Yang Y (2019) Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. Appl Sci 9(2):238
50. Tama BA, Comuzzi M, Rhee K-H (2019) Tse-ids: a two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. IEEE Access 7:94497–94507
51. Tchakoucht TA, Ezziyyani M (2018) Multilayered echo-state machine: a novel architecture for efficient intrusion detection. IEEE Access 6:72458–72468
52. Yu Y, Bian N (2020) An intrusion detection method using few-shot learning. IEEE Access 8:49730–49740
53. Fu A, Dong C, Wang L (2015) An experimental study on stability and generalization of extreme learning machines. Int J Machine Learn Cybern 6(1):129–135

Adel Binbusayyis is currently an Assistant Professor with the College of Engineering and Computer Science, Prince Sattam Bin Abdulaziz University, where he is a specialist in cybersecurity and technology transfer. He is also the Vice-Dean of e-learning with the Deanship of Information Technology and Distance Learning, Prince Sattam Bin Abdulaziz University. He is also an Advisor of Vice Rector with Prince Sattam Bin Abdulaziz University, where he is responsible for monitoring the performance executions of the university strategic goals.



Thavavel Vaiyapuri is currently an Assistant Professor with the College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University. Her research interests include the fields of data science, security, computer vision, and high-performance computing. With nearly 20 years of research and teaching experience, she has published more than 50 research publications in impacted journals and international conferences. She is also a member of the IEEE Computer Society, and also a Fellow of HEA, U.K.