Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

TC 11 Briefing Papers

Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions

Wei Wang[1], *Songlei Jian*[1,*], *Yusong Tan*[*], *Qingbo Wu*, *Chenlin Huang*

The Colledge of Computer, National University of Defense Technology, Changsha, China

ARTICLE INFO

Article history:
Received 8 April 2021
Revised 19 October 2021
Accepted 1 November 2021
Available online 9 November 2021

Keywords:
Network intrusion detection
Representation learning
Feature interaction learning
Deep learning
Multiclass classification

ABSTRACT

Network intrusion detection system is an important cyber defence tool to protect a system from illegal attacks. Building an effective network intrusion detection system that makes good use of deep learning methods is a challenging task. From the object perspective, different types of malicious attacks have a quite imbalance distribution, especially compared with normal network behaviour. From the feature perspective, the network behaviour description contains heterogeneous features, including numeric and categorical features and complex interactions among these features. To address these two challenges, we propose a novel Network Intrusion Detection System which by learning explicit and implicit feature interactions based on representation learning, i.e., RL-NIDS, which models the network behaviour by learning explicit and implicit feature interactions in both feature value representation and object representation spaces. Specifically, the RL-NIDS consists of two main modules, i.e., unsupervised Feature Value Representation Learning module (FVRL) which aims to learn the feature interactions among categorical features explicitly, and supervised Neural Network for object Representation Learning (NNRL) which aims to learn the implicit interactions in the representation space. Experiments show the effectiveness of RL-NIDS and the object representation learned by RL-NIDS with multiclass classification on two real-world datasets. The RL-NIDS outperforms the state-of-the-art feature selection-based methods and deep learning-based methods in terms of both overall accuracy, precision, recall, and F1 score. The accuracy of classification of NSL-KDD and AWIDS dataset is 81.38% and 95.72%, respectively, achieve 3.9% and 0.9% improvements compare to the second-best method. Moreover, a thorough ablation study demonstrates the contributions of both FVRL and NNRL which complement each other for capturing feature interactions.

RL, representation learning; NIDS, network intrusion detection system; FVRL, feature value representation learning; NNRL, neural network for object representation learning.
[*] Corresponding authors.
E-mail addresses: wangwei0114@163.com (W. Wang), jiansonglei@nudt.edu.cn (S. Jian), tanyusong@kylinos.cn (Y. Tan), wuqingbo@kylinos.cn (Q. Wu), clhuang@nudt.edu.cn (C. Huang).
[1] Equal contribution.

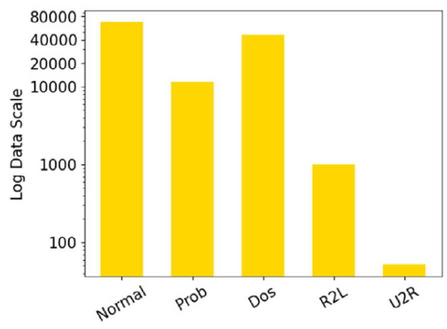
https://doi.org/10.1016/j.cose.2021.102537
0167-4048/© 2021 Published by Elsevier Ltd.

1. Introduction

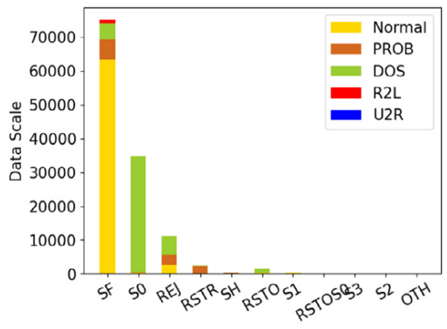
With the widespread use of the Internet, more and more research works focus on cyber security Buczak and Guven (2015); Wang et al. (2017). Among the bunch of cybersecurity defence techniques, the network intrusion detection system (NIDS) is one of the most important tools that can actively protect a system from illegal external attacks. Traditional NIDS is based on pattern matching which compares the patterns of a network against existing malicious patterns which are always summarized by human. Nowadays, an increasing number of researchers try to involve machine learning techniques to make intrusion detection more effective Javaid et al. (2016).

However, the complex internal characteristics of network intrusion behaviours bring challenges to the building of an effective detection system. From the object distribution perspective, different classes of malicious intrusion have quite imbalanced distribution. Take the dataset NSL-KDD Dhanabal and Shantharajah (2015); Mahoney and Chan (2002) as an example, the normal class contains more than 60,000 data objects and the abnormal class R2L and U2R contains only 995 and 52 data objects which are demonstrated in Fig. 1a. From the feature perspective, the description of the network behaviour contains both numeric features (e.g., duration, src_bytes) and categorical features (e.g., protocol_type, service). Also, complex interactions between these features and any individual feature cannot reflect the real data distribution. For example, we sample two features (i.e., flag,protocol_type) and show their distribution in Fig. 1b and Fig. 1c, respectively. According to the figures, the individual feature distribution is quite different from the real data distribution which means different features reflect the network behaviour from different perspectives. Also, we visualize the data with t-sne in Fig. 1d by sampling a few objects from the original dataset. According to the visualization results, the data distribution is quite scattered and the class boundaries are hard to find, especially for the minority class, e.g., 'U2R' and 'R2L'. Therefore, learning the complex interactions between these features and overcoming the imbalance issue are the key to model the network behaviour and detect the malicious intrusion.

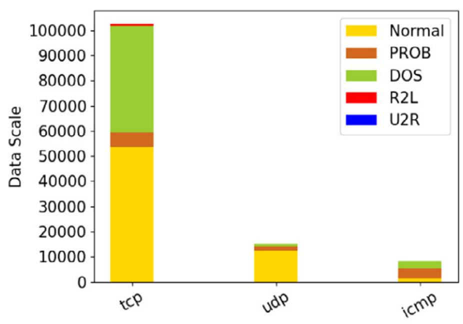
To capture the feature interactions and eliminate the redundancy between features and detect network intrusion automatically, more and more methods try to utilize some effective algorithms including machine learning methods and deep learning methods. And these effective intrusion detection systems can be divided into two categories according to the feature learning methods Panda et al. (2012); Sommer and Paxson (2010), feature selection-based methods and deep learning-based methods. The feature selection methods only consider the interactions through human-defined heuristic rules and some dimension reduction methods. For example, the works in Salo et al. (2019) and Ganapathy et al. (2012) both utilizes information gain to select features in an automatic way. The deep learning-based methods always learn fea-



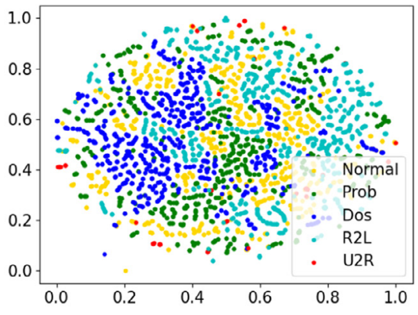
(a) Overall data distribution



(b) Data distribution of 'flag'



(c) Data distribution of 'protocol_type'



(d) T-sne visualization

Fig. 1 – Data analysis of NSL-KDD dataset.

ture representations through deep neural networks. For example, the work Javaid et al. (2016) utilizes sparse autoencoder to learn feature embeddings which are then fed into the multilayer perceptron classifier. And Gamage and Samarabandu (2020) reviews the different neural network architectures for intrusion detection and proposes the best neural network settings. The feature selection-based methods usually rely on some heuristic rules and only capture specific explicit interactions between features, which cannot learn the implicit interactions among the features. Although deep learning can overcome this problem, it is not sensitive to the categorical features and suffers from the limited amount of labelled data.

To address the above challenges, we propose a Representation Learning-based Network Intrusion Detection System, i.e., RL-NIDS, which models the network behaviour by learning explicit and implicit feature interactions in both value and object representation spaces. The RL-NIDS consists of two main modules, i.e., unsupervised Feature Value Representation Learning module (FVRL) and supervised Neural Network for object Representation Learning (NNRL). These two modules are unified by a deep neural network which is constrained by classification loss and triplet loss. The contributions of our work are summarized as follows:

- The proposed FVRL explicitly captures the multi-grain interactions between categorical features values, which can reflect the interactions among multiple features and highlight the abnormal values. Since the FVRL learns feature value representations instead of object embedding, it is not sensitive to the data scale, which can partially overcome the problem of imbalanced data distribution.
- The supervised NNRL harnesses the representation power of neural networks and learns the implicit feature interactions among the heterogeneous features. Also, the learned object representations are denser than the original data by eliminating the redundancy in an automatic way.
- A customized triplet learning scheme is proposed to address the problem of limited labelled data. Through the triplet sampling and triplet loss, the data distribution becomes more balanced, and the abnormal network intrusion behaviours are magnified, which leads to a more discriminative representation and decision boundary.

Experiments show that (1) the RL-NIDS outperforms the state-of-the-art feature selection-based and deep learning-based methods in terms of accuracy, precision, recall and F1 score on two real-world datasets. (2) the object representation learned by RL-NIDS achieves the best performance by feeding into different classifiers compared with the state-of-the-art deep learning methods. (3) the FVRL and NNRL both make contributions to the intrusion detection with a thorough ablation study.

2. Related work

The network intrusion detection can be regarded as a supervised anomaly detection problem which can also be formulated as a classification problem. In recent years, from traditional statistical methods to deep learning-based methods, lots of attempts have been made to extract specific patterns from attack intrusions and to distinguish attack traffic from normal traffic. The NIDS related methods can be categorized as feature selection-based methods and deep learning-based methods.

2.1. *Feature selection-based methods*

Because the original features in network intrusion datasets are always redundant and may contain noise which makes the traditional classifiers hard to find the decision boundaries, the feature selection Eesa et al. (2015) is the preliminary step of following classification tasks in lots of NIDS Aljawarneh et al. (2018); Panigrah and Patra (2016); Selvakumar and Muneeswaran (2019); Zhou et al. (2020). For example, R. Rahmani *et al.* Aslahi-Shahri et al. (2016) employed a hybrid method of support vector machine and genetic algorithm, which reduces the number of features in the dataset of KDDCUP-99 from 41 to 10. The experimental results show that the algorithm shows outstanding true positive values and low false-positive values. Alazzam *et al.* Alazzam et al. (2020) adopted a Pigeon Inspired Optimizer (PIO) to optimize the feature selection process for NIDS. The proposed PIO feature selection algorithm reduced the number of features of KDDCUP99, NSL-KDD, and UNSW-NB15 datasets from 41 features to 7 features, 5 features and 5 features respectively while maintaining a high TPR, accuracy and reducing the required time for building a decision tree. The ensemble learning Seni and Elder (2010); Webb and Zheng (2004) is also widely used in NIDS. For example, Abromman and Reaz Aburomman and Reaz (2016) proposed the Particle Swarm Optimization (PSO) algorithm for intrusion detection, which integrated the results of several learners into a weighted majority vote method to improve accuracy. However, the classifier used in PSO was based on a binary classification algorithm which can only distinguish two states. Also, Yuyang Zhou *et al.* Zhou et al. (2019) proposed an intrusion detection framework which was based on the feature selection and ensemble learning techniques. In his work, An algorithm called CFS-BA was proposed for dimensionality reduction, which selects the optimal subset based on the correlation between features. An ensemble approach which combines C4.5, Random Forest (RF) and Forest by Penalizing Attributes (Forest PA) algorithms by the voting technique was used for classification.

Although feature selection-based methods can reduce the feature dimension, they largely depended on the quality of feature selection algorithms which can not guarantee the best performance on every data scenario, especially for imbalanced dataset. Also, the existing feature selection methods rely on the heuristic rules and metrics, which limits its ability to learn complex interactions between features.

2.2. *Deep learning-based methods*

Deep learning has shown its powerful representation capability in many fields including image representation, text representation and tabular data (Bengio et al., 2013; Jian et al., 2018, 2020) which inspires researchers to apply deep learning methods to learn feature representation in NIDS automatically Gamage and Samarabandu (2020); Parker et al. (2019); Salo et al. (2019); Zhong et al. (2020). For example, Vinayakumar *et al.* Vinayakumar et al. (2019) proposed a hybrid intrusion detection alert system which employs distributed deep learning model with Deep Neural Network (DNN) with five hidden layers for handling and analyzing huge scale data in real-time. They find the most suitable number of layers and nodes by testing the proposed neural network on six datasets for binary classification and multiclass classification. Moreover, Jiang *et al.* Jiang et al. (2018) propose a novel multi-channel effective attack detection method based on long short term memory recurrent neural networks (LSTM-RNNs) which model the attack behaviour as a sequential data by considering the timing of attacks. Autoencoder (AE) is also widely used in NIDS to learn the representation of network behaviour which can be used in the following classifiers. For example, Shone *et al.* Shone et al. (2018) combined non-symmetric deep AE and random forest. In the model, every stacked AE has three hidden layers, and the final encoding layer of the autoencoder acts as the input of the classification model. Also, combing the unsupervised AE with a softmax layer as the classifier to achieve end-to-end training is also adopted in many deep learning-based NIDS Abeshu and Chilamkurti (2018); Javaid et al. (2016); Potluri and Diedrich (2016). Although deep learning can automatically extract the features of the data without any prior manual knowledge, it suffers from the limited training data and cannot capture the interactions between categorical features well since the gradients for categorical data are incalculable. Existing methods only apply deep learning in intrusion detection by transforms categorical feature into one-hot embedding, which cannot model the complex network intrusion behaviour. To address the specific problems in NIDS including the heterogeneous features and limited labelled data in abnormal classes, we fuse the explicit categorical feature learning and implicit deep learning to learn a better representation for network intrusion data.

3. Method

Consider a network behaviour dataset $\mathcal{X}$ with n objects, that is, $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where each object $\mathbf{x}_i = \{x_1, ., x_{d_c}, x_{d_c+1}, \ldots, x_{d_c+d_n}\}$ is described by d_c categorical features and d_n numeric features, and the features belong to $\mathcal{F} = \{f_1, \ldots, f_{d_c}, f_{d_c+1}, \ldots, f_{d_c+d_n}\}$. Each categorical feature f_i has a finite set of values $V_i = \{v_{i1}, v_{i2}, \ldots\}$ and $x_i \in V_i$. Moreover, the values from different features has no intersection such that the number of total feature values is $|V| = \sum_{i=1}^{d} |V_i|$, denoted as m. The $\mathcal{X}$ are with label $\mathcal{Y}$ which contains C classes.

The architecture of RL-NIDS is demonstrated in Fig. 2 which contains two main modules, i.e., the Feature Value Representation Learning module (FVRL) and the Neural Network for object Representation Learning (NNRL). These two modules are two independent parts and they are learning separately. Specifically, the original data features are divided into two parts, i.e., categorical features and numeric features. The categorical features are first fed into the FVRL, which is based on the value-value coupling matrices $\mathbf{M}_c$ and $\mathbf{M}_o$. Then the multi-grain interactions between feature values are explicitly learned by clustering with different granularities which reflect the intrinsic data cluster characteristics. The final feature value embedding is the hidden layers of feature value autoencoder with $\mathcal{L}_{AE}$. The learned feature value embeddings and original numeric features are fused with h to form the input of the NNRL. With the encoder, the representation $\mathbf{r}$ of each data objects is learned under the constraint of both triplet loss $\mathcal{L}_{tri}$ and classification loss $\mathcal{L}_{cls}$. The triplet generator considers the distribution of each class and generates triplets for triplet ranking learning. The total loss is the weighted sum of classification loss and triplet loss.

3.1. *Feature value representation learning*

The unsupervised FVRL captures the feature interactions from the feature value perspective, which make it insensitive to the data scale. Since the value set of numeric features is infinite and neural networks are good at capturing numeric data interactions. FVRL aims to learn numeric embeddings for categorical features. Following the work Jian et al. (2017, 2018), we construct the value-value coupling matrices and then conduct multi-grain clustering on the value matrices to get the value clusters with different granularities. To reduce the dimension of value clusters and learn the nonlinear relationships between different value clusters, we train an autoencoder to learn the feature value embeddings, which form the initial inputs for NNRL.

$p(v_{i1})$ denotes the probability of v_{i1} that calculated by its occurrence frequency and $p(v_{i1}, v_{j1})$ denotes the joint probability of v_{i1} and v_{j1}. Thus the joint probability of v_{i1} and v_{j1} is

$$p(v_{i1}, v_{j1}) = \frac{|\mathbf{x}, \text{s.t.} x_i = v_{i1} \cap x_j = v_{j1}|}{n}, \forall \mathbf{x} \in \mathcal{X} \tag{1}$$

which describe the co-occurrence of feature value v_{i1} and v_{j1}.

To quantify the mutual dependence between two features, we adopt the normalized mutual information (NMI) Estévez et al. (2009). Accordingly, the relation between two features f_a and f_b could be defined as

$$\rho(f_a, f_b) = \frac{2I(f_a, f_b)}{H(f_a) + H(f_b)}, \tag{2}$$

where $I(f_a, f_b)$ is the relative entropy of joint distribution and marginal distribution, and it is written in

$$I(f_a, f_b) = \sum_{v_i \in V_{f_a}} \sum_{v_j \in V_{f_b}} p(v_i, v_j) log \frac{p(v_i, v_j)}{p(v_i)p(v_j)}. \tag{3}$$

$H(f_a)$ and $H(f_b)$ are the marginal entropy of feature f_a and f_b, respectively which can be described by $H(f) = -\sum_{v_i \in V_f} p(v_i) log(p(v_i)), f \in f_a, f_b$.

The value couplings can be quantified by the occurrence frequency and co-occurrence matrix, which are proved to be effective in Jian et al. (2017). Here we use the value couplings to learn value embedding instead of an object to overcome the imbalanced distribution of objects from different classes. In detail, the occurrence-based value coupling function is

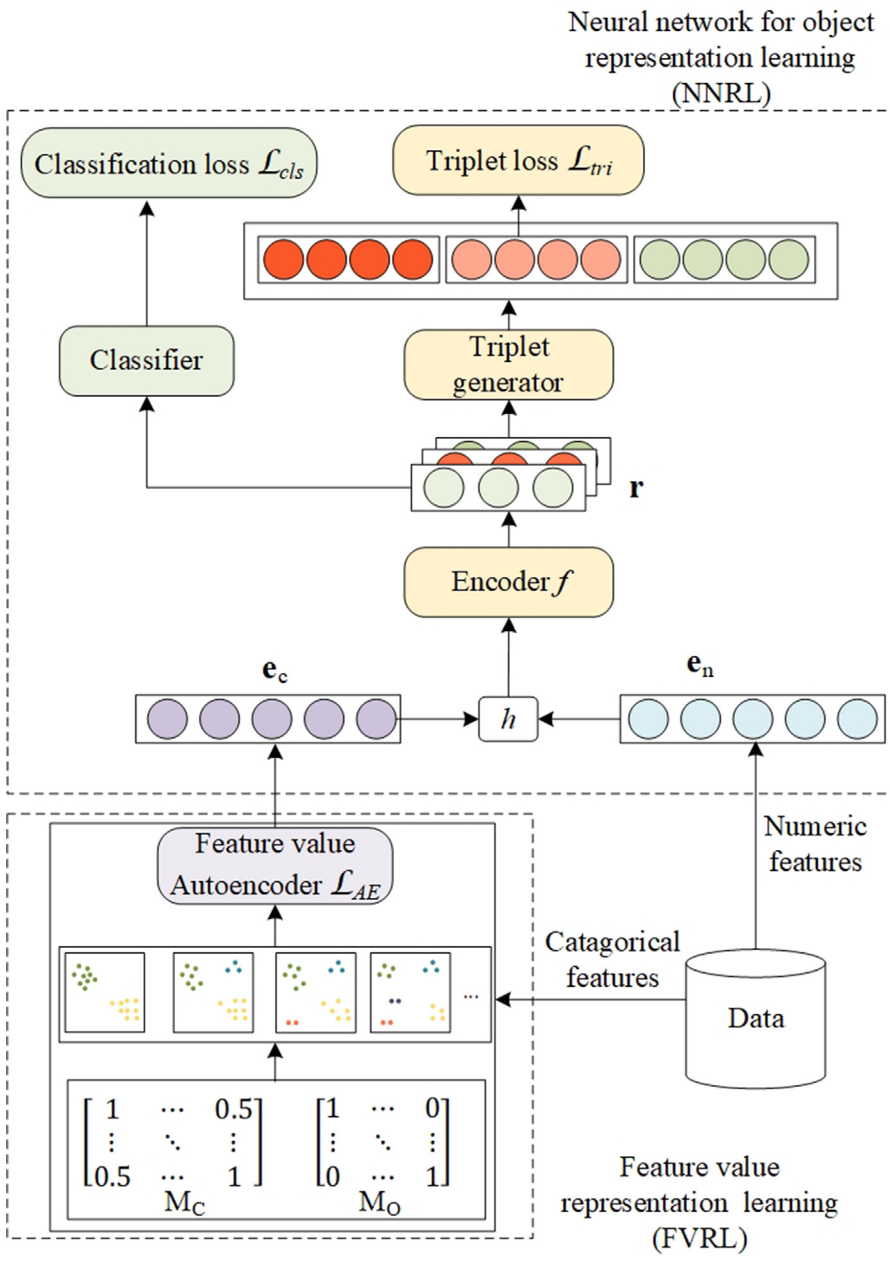

Neural network for object representation learning (NNRL)
Classification loss $\mathcal{L}_{cls}$
Triplet loss $\mathcal{L}_{tri}$
Classifier
Triplet generator
r
Encoder f
$\mathbf{e}_c$
h
$\mathbf{e}_n$
Numeric features
Feature value Autoencoder $\mathcal{L}_{AE}$
Catagorical features
Data
M_C
M_O
Feature value representation learning (FVRL)

Fig. 2 – The architecture of RL-NIDS.

$\xi_o(v_i, v_j) = \rho(f^i, f^j) \times \frac{p(v_j)}{p(v_i)}$, which represents the occurrence frequency of v_i influenced by v_j. In this function, the NMI of two features works as a weight. After constructing the coupling function, the occurrence-based relationship matrix M_o is constructed by:

$$\mathbf{M}_o = \begin{Bmatrix} \xi_o(v_1, v_1) & \cdots & \xi_o(v_1, v_m) \\ \vdots & \ddots & \vdots \\ \xi_o(v_m, v_1) & \cdots & \xi_o(v_m, v_m) \end{Bmatrix} \tag{4}$$

The co-occurrence-based value coupling function is $\xi_c(v_i, v_j) = \frac{p(v_i, v_j)}{p(v_i)}$, which indicates the co-occurrence frequency of value v_i influenced by value v_j. Thus, the co-occurrence-based relationship matrix M_c is designed as follow:

$$\mathbf{M}_c = \begin{Bmatrix} \xi_c(v_1, v_1) & \cdots & \xi_c(v_1, v_m) \\ \vdots & \ddots & \vdots \\ \xi_c(v_m, v_1) & \cdots & \xi_c(v_m, v_m) \end{Bmatrix} \tag{5}$$

Based on the above matrices $\mathbf{M}_o$ and $\mathbf{M}_c$, we can learn the value clusters with different granularities which represent different semantics and well reflect the data characteristics. Here we conduct k-means clustering on the value matrices with different cluster numbers, i.e., $\{k_1, k_2, \ldots, k_o\}$ and $\{k_1, k_2, \ldots, k_c\}$. The clustering results are represented by a cluster membership indicator matrix, where the entry is one if a value is contained in a value cluster and zero otherwise. So we obtain two indicator matrices. We further concatenate these two indicator matrices and obtain a indicator matrix $\mathbf{C} \in \{0,1\}^{m\times k_c}$ where $k_c = (\sum_{i=1}^{o} k_i + \sum_{j=1}^{c} k_j)$. The rows $\mathbf{C}\{0,1\}^{1\times k_c}$ of indicator matrix $\mathbf{C}$ can be regarded as the initial embedding for feature values. However, the dimension is really high, and there may be redundancy information in this indicator matrix. We apply an autoencoder to compress the initial value embedding and learn the non-linear interactions between value clusters. The hidden layer and output layer of the autoencoder are defined as follows:

$$\mathbf{v} = \sigma(\mathbf{W}_e \cdot \mathbf{c}^\top + \mathbf{b}_1) \tag{6}$$

$$\hat{\mathbf{c}} = \sigma(\mathbf{W}_d \cdot \mathbf{v}^\top + \mathbf{b}_2) \tag{7}$$

where the $\mathbf{W}_e \in \mathbb{R}^{k\times k_c}$, $\mathbf{b}_1 \in \mathbb{R}^{k\times 1}$ and $\mathbf{W}_d \in \mathbb{R}^{k_c\times k}$, $\mathbf{b}_2 \in \mathbb{R}^{k_c\times 1}$, and Standardization σ denotes the sigmoid activation function, i.e., $\sigma(z) = \frac{1}{1+e^{-z}}$. Here k is much smaller than k_c. The loss function of value autoencoder is defined as follows:

$$\mathcal{L}_{AE}(\mathbf{c}, \hat{\mathbf{c}}) = -\frac{1}{k}\sum_{j=1}^{k} c_j \cdot \log \hat{c}_j + (1-c_j)\cdot \log(1-\hat{c}_j) \tag{8}$$

The main procedure of feature value embedding is demonstrated in Algorithm 1. Different from the categorical data embedding in Jian et al. (2017), the tiny clusters which contain only one value is kept instead of removing since the tiny clusters reflect the abnormal behaviour of the feature values.

Algorithm 1 Learning Process for FVRL.

```
Input: $\mathcal{X}$ - dataset, $\alpha$ - proportion factor
Output: $\mathbf{V}$ - the embedding of categorical feature values
 1: Generate $\mathbf{M}_o$ and $\mathbf{M}_c$
 2: Initialize $\mathbf{C} = \emptyset$
 3: for $\mathbf{M} \in \{\mathbf{M}_o, \mathbf{M}_c\}$ do
 4:     Initialize $k' = 2$
 5:     $tiny\_cluster = \emptyset$
 6:     repeat
 7:         $\mathbf{C}' = kmeans(\mathbf{M}, k')$
 8:         $\mathbf{C} = \mathbf{C} \cup \mathbf{C}'$
 9:         $k' += 1$
10:     until $size(tiny\_cluster) \geq \lceil \frac{k'}{\alpha} \rceil$
11: end for
12: $\mathbf{V} = Autoencoder(\mathbf{C})$
```

3.2. *Neural network for object representation learning*

By harnessing the powerful representation learning capability of deep neural networks, we design a neural network for object representation learning, i.e., NNRL, based on the FVRL which generates the categorical embedding $\mathbf{e}_c \in \mathbb{R}^{d_c\cdot k\times 1}$ of each data object through the concatenation of its value embeddings. Then the initial input of the neural network is fused by concatenation:

$$\mathbf{e} = h([\mathbf{e}_c, \mathbf{e}_n]) \tag{9}$$

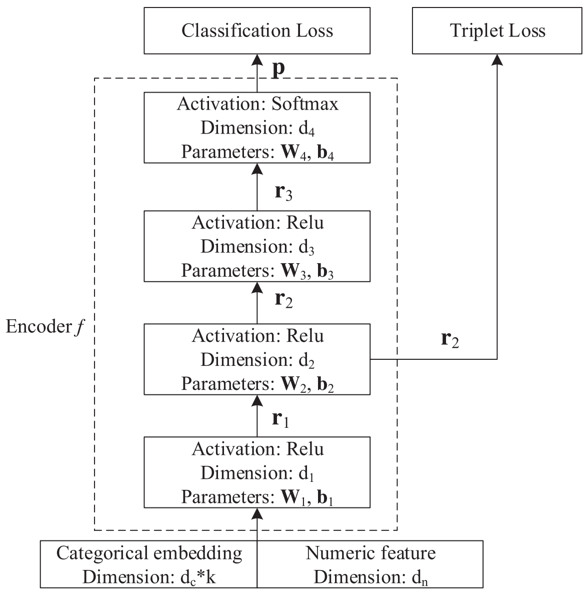


Fig. 3 – The architecture of encoder f in NNRL.

where $\mathbf{e} \in \mathbb{R}^{(d_c\cdot k+d_n)\times 1}$ and h can be a standardization or normalization operation, for example Gaussian standardization or L2 normalization Van Dongen et al. (2003). Then the representation of each data object is learned with an encoder f whose architecture is demonstrated in Fig. 3. As the output of the second layer, i.e., $\mathbf{r}_2$ is be treated as the final representation $\mathbf{r}$:

$$\mathbf{r} = \text{relu}(\mathbf{W}_2 \cdot \text{relu}(\mathbf{W}_1 \cdot \mathbf{e} + \mathbf{b}_1) + \mathbf{b}_2) \tag{10}$$

where the $\mathbf{W}_1 \in \mathbb{R}^{d_1\times(d_c\cdot k+d_n)}$, $\mathbf{b}_1 \in \mathbb{R}^{d_1\times 1}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_2\times d_1}$ and $\mathbf{b}_2 \in \mathbb{R}^{d_2\times 1}$.

Then the predicted probability $\mathbf{p}$ in terms of classification is calculated as follows:

$$\mathbf{p} = \text{softmax}(\text{relu}(\mathbf{W}_3 \cdot \mathbf{r}_2 + \mathbf{b}_3)), \tag{11}$$

where the $\text{softmax}(x_k) = e^{x_k}/\sum_j e^{x_j}$, $\mathbf{W}_3 \in \mathbb{R}^{d_3\times d_2}$, $\mathbf{b}_3 \in \mathbb{R}^{d_3\times 1}$ and $\mathbf{p} \in \mathbf{R}^{d_4\times 1}$ which indicates the classification probability for each class. Then we adopt cross-entropy loss Jang et al. (2016) to train the classifier:

$$\mathcal{L}_{cls} = -\sum_{c=1}^{d_4} y_c \log(p_c) \tag{12}$$

where the d_4 is the number of classes in training data.

Due to the imbalanced distribution, the classes with a few labelled data cannot be well trained only by classification loss. To learn more discriminative representations, we involve triplet loss Hermans et al. (2017) to constrain the representation learning process. Different from classification loss which is decided by the predicted label and the ground-truth label, the triplet loss is based on the distance relationship within one triplet without directly involving labels. Specifically, in terms of one anchor object, i.e., $\mathbf{x}_a$, there are one positive object $\mathbf{x}_p$ and one negative object $\mathbf{x}_n$ which form the triplet

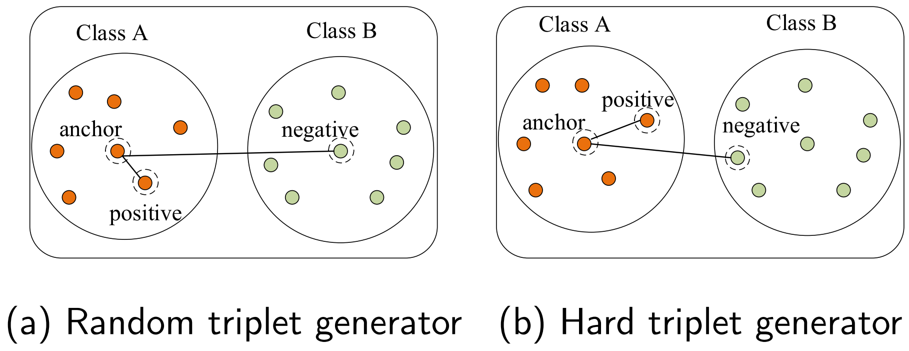


Fig. 4 – Different triplet generator which decides the quality of triplet learning.

$\langle \mathbf{x}_t, \mathbf{x}_p, \mathbf{x}_n \rangle$. The triplet loss in terms of the triplet $\langle \mathbf{x}_a, \mathbf{x}_p, \mathbf{x}_n \rangle$ is defined as follows:

$$\mathcal{L}_{tri} = \max\{0, \varepsilon + dis(\mathbf{x}_a, \mathbf{x}_p) - dis(\mathbf{x}_a, \mathbf{x}_n)\} \tag{13}$$

where the ε is the margin parameter and dis is the distance function based on the object representations. Here we use L2 norm based on data representation, i.e., $dis(\mathbf{x}_a, \mathbf{x}_p) = ||\mathbf{r}_a - \mathbf{r}_p||$.

According to the above loss definition, the distance between anchor object and negative objects will be enlarged while the distance between anchor object and the negative object will be reduced Chen et al. (2017). Therefore, the core problem of triplet loss is the generation of triplets which decides the quality of triplet learning. Since we want to maximize the decision boundary between different classes, it is natural to use class labels to construct the triplet. As shown in Fig. 4a, the anchor and positive objects are from the same class while the negative object is from the different class. Moreover, a good triplet should be hard to distinguish the negative and positive so that the data representation would be more discriminative after training. Therefore, we construct the hard triplet generator, as shown in Fig. 4b, which samples the top k hardest triplets according to the distance difference:

$$dis_{dif} = dis(\mathbf{x}_a, \mathbf{x}_p) - dis(\mathbf{x}_a, \mathbf{x}_n). \tag{14}$$

Then the final loss of representation learning is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{tri}} \tag{15}$$

where the α is the hyperparameter.

We use mini-batch learning to learn the parameters $\mathbf{\Theta} = \{\mathbf{W}_j, \mathbf{b}_j, j \in \{1, 2, 3\}\}$ in NNRL. Then the mean loss of a mini-batch $\mathcal{B}$ for computing gradients is given as:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{\langle \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \rangle \in \mathcal{B}} (\mathcal{L}_{\text{cls}} + \alpha \mathcal{L}_{\text{tri}}) \tag{16}$$

We adopt Adam Kingma and Ba (2014) for optimizing the gradients $\nabla_{\mathbf{\Theta}} \mathcal{L}$ to find the optimal model parameters $\mathbf{\Theta}$. The detailed learning algorithm of the representation neural network is shown in Algorithm 2. We firstly generate triplet by only considering the class label without the dis_{dif} since the initial representations are meaningless. After some training for some epoch, we generate the hard triplet according to the

Algorithm 2 Learning Process for NNRL.

```
Input: $\mathbf{E}_c$ - categorical data embedding, $\mathbf{E}_n$ - numeric data
Output: $\mathbf{R}$ - the representation for data objects
 1: Randomly generate triplets set $\mathbf{T}_r$
 2: for $i = 1$ to epch_r do
 3:     Sample minibatch $\mathcal{B}_j$ from $\mathbf{T}_r$
 4:     $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - Adam[\nabla_{\mathbf{\Theta}} \mathcal{L}]$
 5: end for
 6: for $i = 1$ to epoch do
 7:     Calculate $\mathbf{R}$ according to $\mathbf{\Theta}$ ref. to Eq. (10)
 8:     Initialize $\mathbf{T}_h = \emptyset$
 9:     for each data in class $\mathcal{C}$ do
10:         Randomly generate triplets $\mathbf{T}_C$
11:         $\mathbf{\Omega} \leftarrow$ top $k$ hardest triplets from $\mathbf{T}_C$ ref. to Eq. (14)
12:         $\mathbf{T}_h = \mathbf{T}_h \cup \mathbf{\Omega}$
13:     end for
14:     for $j = 1$ to epoch_h do
15:         Sample minibatch $\mathcal{B}_j$ from $\mathbf{T}_h$
16:         $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - Adam[\nabla_{\mathbf{\Theta}} \mathcal{L}]$
17:     end for
18: end for
```

dis_{dif} by using the updated representations. Then the hard triplet generating process and the parameter learning process, i.e., representation learning process, are iteratively optimized.

4. Experiments

In this section, we firstly introduce detailed experiment setup which includes the evaluation datasets, data preprocessing, comparison methods and evaluation metrics. Then we demonstrate the evaluation performance and analysis the results. Finally, we do a ablation study to reveal the contributions of the two mains parts in RL-NIDS.

4.1. *Experiment setup*

4.1.1. *Datasets*

Data Mining and Knowledge Discovery (KDD) CUP is an annual competition organized by ACM, and KDD Cup99 dataset has always been a classic dataset in the field of intrusion detection. Tallaee *et al.* Tavallaee et al. (2009) proposed a revised version of the KDD Cup99 dataset and named it NSL-KDD in 2009 which deletes some of the duplicate records in KDD Cup99. Zeeshan Ahmad Ahmad et al. (2020) conducted a systematic study and select recent journal articles focusing on various ML- and DL-based NIDS which are published from 2017 to 2020, and the proportion of NSL-KDD or KDD Cup99 was used for testing and validating purposes is 60%. This data shows the NSL-KDD dataset is widely used for NIDS. NSL-KDD dataset covers the KDDTrain dataset as the training set and KDDTest as the testing set, which has one normal record, i.e., Normal, and four different types of attack records, i.e., Dos, Prob, R2L, and U2R. The semantic meaning for each class of NSL-KDD dataset is as follows:

- Normal: this indicates normal connection records.
- Dos: this describes the attacks aiming at making network resource down.
- Probe: this describes the attacks aiming at obtaining detailed statistics of system and network configuration details.
- R2L: this refers to illegal access from a remote computer.
- U2R: this describes the attacks aiming at obtaining the root or super-user access on a particular computer.

The dataset contains 41 features which include 10 basic features, 12 content features, and 19 traffic features. There are 32 numeric features and 9 categorical features in the NSL-KDD dataset. The statistic information for NSL-KDD training and test data is shown in Table 1.

Another dataset we use is Aegean WiFi Intrusion Dataset (AWID) Kolias et al. (2015) which captures 37 million packets with a small network environment in one hour. The dataset contains training set and test set as shown in Table 2 which have one normal class and three types of attacks, i.e., Flooding, Impersonation and Injection, whose semantic meaning is shown in follows:

- Normal: this indicates normal connection records.
- Flooding: this refers to the attacks which create a sudden increase in the management frames per time unit.

Table 1 – The statistic of NSL-KDD dataset.

Class	Training #	Training %	Test #	Test %
Normal	67,343	53.59%	9710	43.07%
Dos	45,927	36.54%	7458	33.08%
Probe	11,656	9.27%	2421	10.74%
R2L	995	0.79%	2754	12.22%
U2R	52	0.04%	200	0.89%
Total	125,673	100%	22,543	100%

Table 2 – The statistic of AWID dataset.

Class	Training #	Training %	Test #	Test %
Normal	1,631,218	90.96%	530,785	92.21%
Flood	48,484	2.70%	8097	1.41%
Imper	48,522	2.70%	20,079	3.49%
Injection	65,379	3.64%	16,682	2.90%
Total	1,793,603	100%	575,643	100%

Table 3 – The 23 features we use in AWID dataset.

Index	Feature Name	Type
4	frame.time_epoch	numeric
5	frame.time_delta	numeric
7	frame.time_relative	numeric
8	frame.len	numeric
38	radiotap.mactime	numeric
47	radiotap.datarate	numeric
48	radiotap.channel.freq	categorical
50	radiotap.channel.type.cck	categorical
61	radiotap.dbm_antsignal	numeric
64	wlan.fc.type_subtype	categorical
66	wlan.fc.type	categorical
67	wlan.fc.subtype	categorical
68	wlan.fc.ds	categorical
69	wlan.fc.frag	categorical
70	wlan.fc.retry	categorical
71	wlan.fc.pwrmgt	categorical
72	wlan.fc.moredata	categorical
73	wlan.fc.protected	categorical
75	wlan.duration	numeric
141	wlan.wep.key	categorical
146	wlan.qos.priority	categorical
151	wlan.qos.bit4	categorical

- Impersonation (i.e., Imper): this refers to the attacks which introduce additional access points in the neighbourhood broadcasting Beacon frames that advertise a pre-existing valid network.
- Injection: this refers to the attacks which cause a deluge of validly encrypted data frames of smaller size.

4.1.2. *Data preprocessing*

The following pre-processing steps are performed on the datasets.

- Data cleaning: The original AWID dataset can be organized as tabular data with 156 features and contains many missing data Aminanto and Kim (2016). We firstly pre-process the dataset by removing the duplicated features and invalid values in the fields. For example, the value of normalized mutual information between the ’radiotap.length’ feature and the ’radiotap.present.tsft’ feature is 1. It means these two features have the same distribution and can offer duplicated contribution to a classifier. The value of the feature ’frame.interface.id’ is 0 for all objects, and it offers no contribution to a classifier, it also should be removed from features. Then we remove these objects with missing feature values and generate a new dataset with 23 features with 14 categorical features and 9 numeric features, shown in Table 3. For NSL-KDD dataset, all objects can be saved.
- Numericalization: There are 9 categorical features in NSL-KDD dataset and 14 categorical features in AWID dataset. Because the input value of contrast models should be a numeric matrix, we convert these non numeric features into numeric form by one-hot encoded Buckman et al. (2018) in comparison methods.
- Scaling features: The difference between the maximum and minimum values has a very large scope in some features Van et al. (2017). All features of two datasets are scaled by standard normalization.

4.1.3. *Comparison methods*

We choose seven state-of-the-art network intrusion detection methods which including five feature selection-based methods and two deep learning-based methods for different datasets.

- CPIO Alazzam et al. (2020): this work proposed a pigeon inspired optimizer-based feature selection algorithm for intrusion detection on NSL-KDD dataset. They selected 18 and 5 features by Sigmoid-PIO and Cosine-PIO methods as the input data separately, and selected decision tree as the classifier.
- IG-Hybrid Aljawarneh et al. (2018): this work filtered features by using the voting algorithm with Information Gain (IG) in NSL-KDD dataset and used hybrid classifiers which contains: J48, Meta Pagging, RandomTree, REPTree, AdaBoostM1, DecisionStump and NaiveBayes.The number of features they selected is 8
- CFS-BA-Ensemble Zhou et al. (2020): this work combined the correlation-based feature selection and bat algorithm to select features in NSL-KDD dataset and select 10 feature as the input data of classifier. The classifier adopts an ensemble approach that combines C4.5, Random Forest (RF), and Forest by Penalizing Attributes (Forest PA) algorithms.
- IG-PCA Salo et al. (2019): this work combined IG and principal component analysis (PCA) to select the features and reduce the dimension of NSL-KDD dataset, used an ensemble classifier based on support vector machine (SVM), Instance-based learning algorithms (IBK), and multilayer perceptron (MLP).
- IG-CH Thanthrige et al. (2016): this work used Information Gain (IG) and Chi-Squared statistics (CH) to select the features in AWID dataset and use the random forest as the classifier.
- AE-MLP Javaid et al. (2016): this work adopted Autoencoder to learn a feature embedding in an unsupervised way and then trained multilayer perceptron classifier.
- DNN Gamage and Samarabandu (2020): this work explored different parameter settings for deep neural network and proposed the state-of-the-art deep neural network which contains three hidden layers.

All parameters in the above comparison methods are set according to the original papers.

4.1.4. *RL-NIDS Implementation*

As discussed in the previous section, the original data features are divided into two parts, categorical features and numeric features. The number of categorical features of NSL-KDD dataset and AWID dataset is 9, 14 respectively. The categorical features are first fed into the FVRL module to calculate the value-value coupling matrices $\mathbf{M}_c$ and $\mathbf{M}_o$. Then the multi-grain interactions between feature values are explicitly learned by clustering with different granularities which reflect the intrinsic data cluster characteristics. The dimension of hidden layer of feature value autoencoder is 10 and 5 for NSL-KDD dataset and AWID dataset respectively.

The learned feature value embeddings and original numeric features are fused to form the input of the NNRL module.To NSL-KDD dataset, the parameters in encoder f (refer to Fig. 3) of the NNRL are set to: $d_1 = 64$, $d_2 = 32$, $d_3 = 16$ and $d_4 = 5$. The activation function is RELU except the last activation function is Softmax. The optimizer is Adam and the learning rate is 0.0006. In the generation of triplets, we randomly select part of samples as anchor points from large number of class, and take all samples as anchor points from fewer class, which is beneficial to improve the detection rate of fewer samples. The final loss of NNRL is the sum of classification loss and triplet loss. In NNRL,we set the batch_size is 256 and epoch is 20 in training step, and we adopt 10-fold cross-validation to get a better model.

4.1.5. *Evaluation metrics*

In the field of network intrusion detection, samples that contain attacks can be defined as positive samples, and samples that do not contain attacks can be defined as negative samples Kasongo and Sun (2020). The result of the classification may be correct or incorrect. All possible results can be divided into the following four situations:

- True Positive (TP): Positive samples are classified as positive samples; it means the number of Attack objects classified correctly.
- True Negative (TN): Negative samples are classified as negative samples; it means the number of Normal objects classified correctly.
- False Positive (FP): Negative sample is classified as a positive sample; it means the number of Normal objects wrongly classified as an attack.
- False Negative (FN): Positive sample is classified as a negative sample; it means the number of Attack objects wrongly classified as normal.

We adopt four metrics Yin et al. (2017) to evaluate the classification performance:

- Accuracy: the proportion of correctly classified objects out of a given set of objects

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: the proportion of correct classifications out of a set of predictions classified as attack

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall: the proportion of correct attack classifications out of a given set of attack objects

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1 score: the harmonic mean of precision and recall which suits for class-imbalanced problems

$$\text{F1} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

For multiclass classification problems, precision, recall and F1 scores can be calculated per-class, treating the classification as a one-vs-all problem. The final metrics are obtained

Table 4 – The mutli-class classification results with the state-of-the-art methods on NSL-KDD dataset.

Method	Accuracy	Precision	Recall	F1	F1-Normal	F1-Prob	F1-DOS	F1-R2L	F1-U2R	Time(min)
CPIO Alazzam et al. (2020)	0.7579	0.7543	0.7579	0.7117	0.7909	0.8024	0.8475	0.0349	**0.0198**	0.003
IG-Hybrid Aljawarneh et al. (2018)	0.6701	0.6055	0.6701	0.5996	0.7511	0.2377	0.7572	0.0000	0.0000	1.383
CFS-BA-Ensemble Zhou et al. (2020)	0.7211	0.7593	0.7211	0.6777	0.7830	0.6439	0.7869	0.0882	0.0193	0.115
IG-PCA Salo et al. (2019)	0.5790	0.5043	0.5790	0.5111	0.7451	0.1236	0.5348	0.0000	0.0000	2.317
AE-MLP Javaid et al. (2016)	0.7295	0.6661	0.7295	0.6783	0.7726	0.6474	0.8344	0.0000	0.0000	2.367
DNN Gamage and Samarabandu (2020)	0.7821	0.7975	0.7821	0.7455	0.8045	0.7681	0.8854	0.1932	0.0000	2.152
RL-NIDS	**0.8138**	**0.8369**	**0.8138**	**0.7879**	**0.8296**	**0.8165**	**0.9100**	**0.3424**	0.0000	4.883

by weighted macro averaging every class whose weight is decided by the number of objects in the class.

4.2. *Results and analysis*

To demonstrate the effectiveness of our proposed RL-NIDS, we evaluate the RL-NIDS and the state-of-the-art intrusion detection methods with multiclass classification tasks on both NSL-KDD dataset and AWID dataset. Also, to show the effectiveness of the representation learned by RL-NIDS, we compare the representations with other deep learning-based methods by feeding the representations into different types of classifiers.

4.2.1. *The effectiveness of RL-NIDS*

The multiclass classification results on NSL-KDD dataset are demonstrated in Table 4, which includes the overall performance and detailed performance for each class. The CPIO, IG-Hybrid, CFS-BA-Ensemble and IG-PCA are feature selection-based methods which reduce the original 41 features to 5 features, 8 features, 13 features and 12 features, respectively. According to Table 4, RL-NIDS achieves the best performance in terms of accuracy, precision, recall and F1-score compared with all other methods. Specifically, RL-NIDS achieves 6.9%, 17.7%, 11.4%, 28.8%, 10.4%, 3.9% improvements over CPIO, IG-Hybrid, CFS-BA-Ensemble, IG-PCA, AE-MLP and DNN in terms of accuracy, respectively.

Among feature selection methods, CPIO beats other feature selection-based methods since it adopts the sophisticated feature interaction learning process. Overall, the deep learning-based methods outperform feature selection-based methods which indicate the neural networks is more capable of capturing complex and implicit interactions among features.

For the specific detection performance of each class in NSL-KDD dataset, the proposed RL-NIDS achieves the 3.0%, 5.9%, 2.7%, 43.6% F1-score improvements over the second-best method, i.e., DNN, on Normal, Prob, DOS, R2L, respectively. Especially, the objects of R2L class only occupy 0.79% percent of all training data according to Table 1. The good performance of RL-NIDS on R2L attack detection indicates its capability of handling imbalanced data. However, the U2R class only contains 52 objects (0.04%) in training data which leads to the difficulty of detection for all methods, which can be proved by the t-sne visualization results 1d.

The multiclass classification results on AWID dataset are demonstrated in Table 5. Because the CPIO, IG-Hybrid and CFS-BA-Ensemble did not report the feature selection results of AWID dataset and the original papers are not detailed enough to reproduce the feature selection process, we only report the IG-CH result which has similar feature selection method with IG-Hybrid. The IG-CH selected 10 features from the original 156 features while other deep learning-based methods use the 23 features shown in Table 3. According to Table 5, RL-NIDS achieves the best overall performance compared with IG-GH, AE-MLP and DNN in terms of accuracy, recall and F1-score. For the specific detection performance of each class in AWID dataset, the feature selection-based method IG-GH can only detect normal data and Injection attack which occupy the majority of the training dataset. In contrast, the deep learning-based methods can detect the Flooding attacks which occupy only 2.7% in training data. Moreover, the RL-NIDS achieves the best detection performance on both Flooding and Impersonation attacks.

Moreover, we test the training time of each methods in the experiment and the experimental platform adopts the Linux operating system. The CPU of the host is Intel(R) Xeon(R) Silver 4214 CPU and the main frequency is 2.2GHz.The result as shown in the last column of Table 4 and Table 5. The training time of the feature selection-based methods are less than the training time of methods based on deep learning while the performance of deep learning-based methods is better than them. Since we capture more sophistic feature interactions through feature value representation learning and deep neural network, the training time of our proposed is longer than other methods. Luckily, deep learning-based methods can be largely accelerated by GPU or other hardware since the ma-

Table 5 – The mutli-class classification results with the state-of-the-art methods on AWID dataset.

	Accuracy	Precesion	Recall	F1	F1-Normal	F1-Flooding	F1-Imper	F1-Injection	Time(min)
IG-CH Thanthrige et al. (2016)	0.9305	0.9018	0.9305	0.9124	**0.9943**	0.0000	0.0000	0.5639	0.189
AE-MLP Javaid et al. (2016)	0.9484	**0.9683**	0.9484	0.9399	0.9905	0.5177	0.0365	0.6246	7.238
DNN Gamage and Samarabandu (2020)	0.9479	0.9287	0.9479	0.9343	0.9726	0.5455	0.0335	0.9901	6.882
RL-NIDS	**0.9572**	0.9398	**0.9572**	**0.9447**	0.9923	**0.7251**	**0.1023**	**0.9910**	20.838

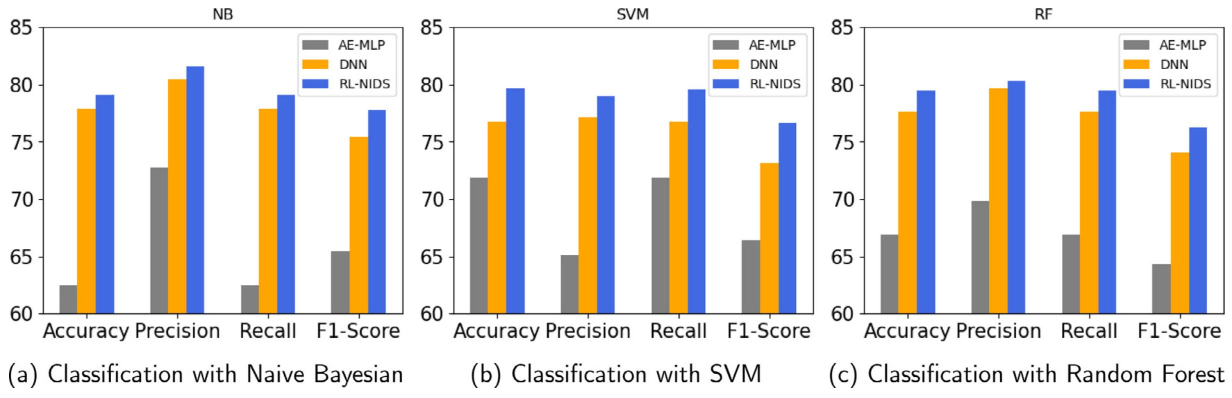


Fig. 5 – The multiclass classification results (%) of different representations in different classifier on NSL-KDD dataset.

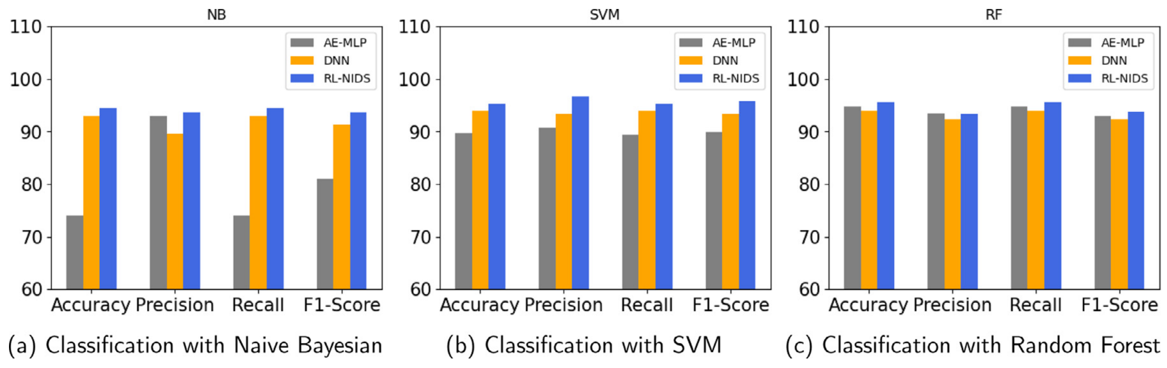


Fig. 6 – The multiclass classification results (%) of different representations in different classifier on AWID dataset.

trix operations on GPU is constant time. Also the most time-consuming part, i.e., triplets sampling process, in our method can be accelerated by batch operation and stratified sampling Poorheravi et al. (2020).

4.2.2. *The effectiveness of object representations*

To illustrate the quality of object representations learned by deep learning-based methods, we apply the representations learned by AE-MLP, DNN and RL-NIDS into three different types of classifier, i.e., Naive Bayesian (NB), Support Vector Machine (SVM) and Random Forest (RF). A good representation should have generalization ability which means it should not be sensitive to the classifiers. The multiclass classification results on NSL-KDD dataset and AWID are demonstrated in Fig. 5 and Fig. 6, respectively. According to Fig. 5, no matter with which classifier, RL-NIDS achieves the best performance in terms of overall accuracy, precision, recall and F1 score. Also, the representation learned by RL-NIDS performs relatively stably on each classifier while results of AE-MLP fluctuate with the change of classifier.

The classification results are decided by three important elements: data distribution, representation quality, and classifier quality. According to the Fig. 5 and Fig. 6, the results are quite different because the distribution of the NSL-KDD and AWID datasets are quite different. The normal class of AWID (90.96% according to Table 2) are much larger than the normal class of NSL-KDD (53.59% according to Table 1), which means the overall performance of AWID is mainly decided by the detection rate of the normal class. The importance of representation is vividly demonstrated by the Fig. 5 and Fig. 6. In terms of the same dataset and the same classifier, different representations perform quite different. Among the representation learned by different deep learning-based methods, our proposed RL-NIDS benefits from the triplet loss which overcomes the imbalance issue of the dataset to a certain extend. With a good representation, the classifier can boost its best performance according to the figures.

4.3. *Ablation study*

To demonstrate the effectiveness of each module in RL-NIDS, i.e., FVRL and NNRL, we do a thorough ablation study on NSL-KDD dataset. The FVRL accepts the original categorical features and generates the numeric embedding which is fed into the NNRL module. The NNRL is trained with the loss in Eq. 15 which contains classification loss and triplet loss.

4.3.1. *The effectiveness of FVRL*

To test the influence of different data input on the final classification results, we construct three types of inputs:

- Cate+Num: the concatenation of one-hot encoding for categorical features and numeric features.

Table 6 – The multiclass classification results with different data inputs on NSL-KDD dataset.

Classifier	Input	Accuracy	Precesion	Recall	F1	F1-Normal	F1-Prob	F1-DOS	F1-R2L	F1-U2R
RF	Cate+ Num	0.7538	0.8157	0.7538	0.7129	0.7798	0.7547	0.8607	**0.0827**	0.0291
	FVRL	0.7211	0.6568	0.7211	0.6796	0.7880	0.5174	0.8522	0.0046	**0.2333**
	FVRL+Num	**0.7710**	**0.8197**	**0.7710**	**0.7247**	**0.7927**	**0.7736**	**0.8922**	0.0335	0.1105
NNRL	Cate+ Num	0.7828	0.7762	0.7828	0.7312	0.7918	0.7928	0.8893	0.0457	0.0000
	FVRL	0.7235	0.6429	0.7235	0.6582	0.7971	0.4084	0.8193	0.0000	0.0000
	FVRL+Num	**0.8137**	**0.8369**	**0.8138**	**0.7879**	**0.8296**	**0.8165**	**0.9100**	**0.3424**	0.0000
ssss										

Table 7 – The multiclass classification results with different loss functions in NNRL on NSL-KDD dataset.

	Accuracy	Precesion	Recall	F1	F1-Normal	F1-Prob	F1-DOS	F1-R2L	F1-U2R
NNRL ($\mathcal{L}_{cls\_mse}$)	0.7255	0.6601	0.7255	0.6787	0.7542	0.6781	0.8493	0.0000	0.0000
NNRL ($\mathcal{L}_{cls\_mae}$)	0.7121	0.6590	0.7121	0.6634	0.7516	0.6862	0.8033	0.0014	0.0000
NNRL ($\mathcal{L}_{cls\_possion}$)	0.7905	0.8252	0.7905	0.7498	0.8102	0.7901	0.9032	0.1737	0.0000
NNRL ($\mathcal{L}_{cls}$)	0.7911	0.8249	0.7911	0.7513	0.8126	0.7957	0.9017	0.1437	0.0000
NNRL ($\mathcal{L}_{cls} + \mathcal{L}_{tri}$)	**0.8137**	**0.8369**	**0.8138**	**0.7879**	**0.8296**	**0.8165**	**0.9100**	**0.3424**	0.0000

- FVRL: the output embedding of FVRL with only categorical features as input.
- FVRL+Num: the concatenation of the FVRL output embedding and numeric feature which is the input for RL-NIDS.

These three inputs are evaluated by two classifiers, i.e., Random Forest (RF) and NNRL. The classification results are demonstrated in Table 6. According to the results of Cate+Num and FVRL+Num in both RF and NNRL, the embedding learned by FVRL is more informative than one-hot encoding, which means the FVRL capture more sophisticated feature interactions in categorical features. The NNRL with Cate+Num as input denotes the RL-NIDS without FVRL which performs worse than the NNRL with FVRL+Num (i.e., full RL-NIDS model). These results indicate the contribution of FVRL and the importance of capturing the feature interactions explicitly, which plays a complementary role with NNRL.

4.3.2. *The effectiveness of NNRL*

To test the contribution of the different loss function in Eq. 12 and Equation 15, we construct the following loss functions:

- $\mathcal{L}_{cls\_mse}$: only classification loss with Mean Square Error (MSE) loss function.
- $\mathcal{L}_{cls\_mae}$: only classification loss with Mean Absolute Error (MAE) loss function.
- $\mathcal{L}_{cls\_possion}$: only classification loss with Possion loss function.
- $\mathcal{L}_{cls}$: only classification loss with cross-entropy loss function in Eq. 12.
- $\mathcal{L}_{cls} + \mathcal{L}_{tri}$: the combination of classification loss and triplet loss.

The classification results with different loss functions are demonstrated in Table 7. According to the results, the cross-entropy performs the best among other classification loss functions. Moreover, the combination of triplet loss and classification loss, i.e., NNRL ($\mathcal{L}_{cls} + \mathcal{L}_{tri}$), achieves the best performance which indicates the importance of triplet loss. According to Table 6, Table 4 and Table 7, the class U2R cannot be detected by all deep learning-based methods. The main reason is the very few training objects lead to the poor fitting for neural networks. This is also a future work or direction for detecting these few labeled attacks.

5. Conclusion

NIDS is essential to cybersecurity and making good use of deep learning techniques to build NIDS is not a trivial task. In this work, we propose an effective NIDS, i.e., RL-NIDS, which contains explicit feature interaction learning and implicit deep representation learning. The explicit feature interaction learning captures the network behaviour through a multi-grain clustering and highlights the abnormal feature value in the learned embedding in an unsupervised way. The implicit representation learning is implemented by a neural network which is constrained by classification loss and triplet loss. We design a customized triplet generating and learning process to learn a more discriminative representation and decision boundaries to overcome the imbalanced data distribution issue. By applying the RL-NIDS and representation learned by RL-NIDS to multiclass classification on NSL-KDD and AWID datasets, the superior performance of RL-NIDS shows its effectiveness and the generalization ability of the representations. Specifically, RL-NIDS achieves 3.9%, 4.1%, 3.9%, 4.2% improvements over the second-best method in terms of accuracy, precision, recall and F1 score, respectively. For the specific detection performance of each class in the NSL-KDD dataset, the proposed RL-NIDS achieves 43.6% F1-score improvements over the second-best method on R2L. Especially the objects of R2L class only occupy 0.79% percent of all training data and it indicates its capability of handling imbalanced data. Moreover, the ablation study demonstrates the contribution of both explicit feature interaction learning and implicit representation learning neural network.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Wei Wang: Methodology, Software, Investigation. **Songlei Jian:** Conceptualization, Writing – original draft. **Yusong Tan:** Resources, Project administration. **Qingbo Wu:** Supervision. **Chenlin Huang:** Writing – review & editing.

Acknowlgedgments

This work is supported by the National Natural Science Foundation of China (Nos. 62002371 and 62172431), State Administration of Science Technology and Industry for National Defense Foundation (No. WDZC20205250104), the National University of Defense Technology Foundation (Nos. ZK21-17 and ZK20-09) and Natural Science Foundation of Hunan Province (No. 2021JJ40692).

REFERENCES

Abeshu A, Chilamkurti N. Deep learning: the frontier for distributed attack detection in fog-to-things computing. IEEE Commun. Mag. 2018;56(2):169–75.
Aburomman AA, Reaz MBI. A novel svm-knn-pso ensemble method for intrusion detection system. Appl Soft Comput 2016;38:360–72.
Ahmad Z, Khan AS, Shiang CW, Abdullah J, Ahmad F. Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies 2020.
Alazzam H, Sharieh A, Sabri KE. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst Appl 2020;148:113249.
Aljawarneh S, Aldwairi M, Yassein MB. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. J Comput Sci 2018;25:152–60.
Aminanto ME, Kim K. Detecting impersonation attack in wifi networks using deep learning approach. In: International Workshop on Information Security Applications. Springer; 2016. p. 136–47.
Aslahi-Shahri B, Rahmani R, Chizari M, Maralani A, Eslami M, Golkar MJ, Ebrahimi A. A hybrid method consisting of ga and svm for intrusion detection system. Neural computing and applications 2016;27(6):1669–76.
Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence 2013;35.
Buckman J, Roy A, Raffel C, Goodfellow I. Thermometer encoding: One hot way to resist adversarial examples. International Conference on Learning Representations, 2018.
Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications surveys & tutorials 2015;18(2):1153–76.
Chen W, Chen X, Zhang J, Huang K. Beyond triplet loss: a deep quadruplet network for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 403–12.
Dhanabal L, Shantharajah S. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. International Journal of Advanced Research in Computer and Communication Engineering 2015;4(6):446–52.
Eesa AS, Orman Z, Brifcani AMA. A new feature selection model based on id3 and bees algorithm for intrusion detection system. Turkish Journal of Electrical Engineering & Computer Sciences 2015;23(2):615–22.
Estévez PA, Tesmer M, Perez CA, Zurada JM. Normalized mutual information feature selection. IEEE Trans. Neural Networks 2009;20(2):189–201.
Gamage S, Samarabandu J. Deep learning methods in network intrusion detection: a survey and an objective comparison. Journal of Network and Computer Applications 2020;169:102767.
Ganapathy S, Yogesh P, Kannan A. Intelligent agent-based intrusion detection system using enhanced multiclass svm. Comput Intell Neurosci 2012;2012.
Hermans A, Beyer L, Leibe B. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 2017.
Jang E, Gu S, Poole B. Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 2016.
Javaid A, Niyaz Q, Sun W, Alam M. A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS); 2016. p. 21–6.
Jian S, Cao L, Pang G, Lu K, Gao H. Embedding-based representation of categorical data by hierarchical value coupling learning. IJCAI International Joint Conference on Artificial Intelligence, 2017.
Jian S, Hu L, Cao L, Lu K. Metric-based auto-instructor for learning mixed data representation. Proceedings of the AAAI Conference on Artificial Intelligence 2018.
Jian S, Hu L, Cao L, Lu K. Representation learning with multiple Lipschitz-constrained alignments on partially-labeled cross-domain data. Proceedings of the AAAI Conference on Artificial Intelligence 2020.
Jian S, Pang G, Cao L, Lu K, Gao H. Cure: flexible categorical data representation by hierarchical coupling learning. IEEE Trans Knowl Data Eng 2018;31(5):853–66.
Jiang F, Fu Y, Gupta BB, Lou F, Rho S, Meng F, Tian Z. Deep learning based multi-channel intelligent attack detection for data security. IEEE Trans. Sustainable Comput. 2018.
Kasongo SM, Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. J Big Data 2020;7(1):1–20.
Kingma D, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 2014.
Kolias C, Kambourakis G, Stavrou A, Gritzalis S. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials 2015;18(1):184–208.
Mahoney MV, Chan PK. Learning nonstationary models of normal network traffic for detecting novel attacks. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining; 2002. p. 376–85.
Panda M, Abraham A, Patra MR. A hybrid intelligent approach for network intrusion detection. Procedia Eng 2012;30:1–9.
Panigrah A, Patra MR. Fuzzy rough classification models for network intrusion detection. Transactions on Machine Learning and Artificial Intelligence 2016;4(2):07.

Parker LR, Yoo PD, Asyhari TA, Chermak L, Jhi Y, Taha K. Demise: Interpretable deep extraction and mutual information selection techniques for iot intrusion detection. In: Proceedings of the 14th International Conference on Availability, Reliability and Security; 2019. p. 1–10.

Poorheravi PA, Ghojogh B, Gaudet V, Karray F, Crowley M. Acceleration of large margin metric learning for nearest neighbor classification using triplet mining and stratified sampling. arXiv preprint arXiv:2009.14244 2020.

Potluri S, Diedrich C. Accelerated deep neural networks for enhanced intrusion detection system. In: 2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA). IEEE; 2016. p. 1–8.

Salo F, Nassif AB, Essex A. Dimensionality reduction with ig-pca and ensemble classifier for network intrusion detection. Comput. Networks 2019;148:164–75.

Selvakumar B, Muneeswaran K. Firefly algorithm based feature selection for network intrusion detection. Computers & Security 2019;81:148–55.

Seni G, Elder JF. Ensemble methods in data mining: improving accuracy through combining predictions. Synthesis lectures on data mining and knowledge discovery 2010;2(1):1–126.

Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence 2018;2(1):41–50.

Sommer R, Paxson V. Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE symposium on security and privacy. IEEE; 2010. p. 305–16.

Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the kdd cup 99 data set. In: 2009 IEEE symposium on computational intelligence for security and defense applications. IEEE; 2009. p. 1–6.

Thanthrige USKPM, Samarabandu J, Wang X. Machine learning techniques for intrusion detection on public dataset. In: 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). IEEE; 2016. p. 1–4.

Van NT, Thinh TN, et al. An anomaly-based network intrusion detection system using deep learning. In: 2017 International Conference on System Science and Engineering (ICSSE). IEEE; 2017. p. 210–14.

Van Dongen J, Langerak A, Brüggemann M, Evans P, Hummel M, Lavender F, Delabesse E, Davi F, Schuuring E, García-Sanz R, et al. Design and standardization of pcr primers and protocols for detection of clonal immunoglobulin and t-cell receptor gene recombinations in suspect lymphoproliferations: report of the biomed-2 concerted action bmh4-ct98-3936. Leukemia 2003;17(12):2257–317.

Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access 2019;7:41525–50.

Wang H, Gu J, Wang S. An effective intrusion detection framework based on svm with feature augmentation. Knowl Based Syst 2017;136:130–9.

Webb GI, Zheng Z. Multistrategy ensemble learning: reducing error by combining ensemble learning techniques. IEEE Trans Knowl Data Eng 2004;16(8):980–91.

Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 2017;5:21954–61.

Zhong Y, Chen W, Wang Z, Chen Y, Wang K, Li Y, Yin X, Shi X, Yang J, Li K. Helad: a novel network anomaly detection model based on heterogeneous ensemble learning. Comput. Networks 2020;169:107049.

Zhou Y, Cheng G, Jiang S, Dai M. An efficient intrusion detection system based on feature selection and ensemble classifier. arXiv preprint arXiv:1904.01352 2019.

Zhou Y, Cheng G, Jiang S, Dai M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput. Networks 2020:107247.

Wei Wang received the M.S. degree in computer science and technology from the National University of Defense Technology, in 2015, where she is currently pursuing the Ph.D. degree. Her research interests include intrusion detection and artificial intelligence in computer system.

Songlei Jian received the B.Sc. degree and Ph.D. degree in computer science from College of Computer, National University of Defense Technology, Changsha, China, in 2013 and 2019, respectively. She is currently an assistant professor with the College of Computer, NUDT. Her research interests include representation learning, anomaly detection, computer security and artificial intelligence in computer system.

Yusong Tan received the Ph.D. degree in computer science and technology from the National University of Defense Technology, in 2004, where he is currently a Professor with the College of Computer, NUDT. His research interests include cloud computing, operating systems and computer security.

Qingbo Wu received the Ph.D. degree in computer science and technology from the National University of Defense Technology, in 2010, where he is currently a Professor with the College of Computer, NUDT. His research interests include operating systems and computer security

Chenlin Huang received his Ph.D. in computer science from National University of Defense Technology in 2005, where he is currently a Professor with the College of Computer, NUDT. His major research fields include trust management, computer security, operating system and cloud computing.