IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 15, 2020, accepted November 19, 2020, date of publication November 25, 2020,
date of current version December 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040510

NADS-RA: Network Anomaly Detection Scheme Based on Feature Representation and Data Augmentation

XU LIU[1,2], XIAOQIANG DI[1,2,3], QIANG DING[3], WEIYOU LIU[1], HUI QI[1,2], JINQING LI[1,2], AND HUAMIN YANG[1,2]

[1]School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China
[2]Jilin Province Key Laboratory of Network and Information Security, Changchun University of Science and Technology, Changchun 130022, China
[3]Information Center, Changchun University of Science and Technology, Changchun 130022, China

Corresponding author: Xiaoqiang Di (dixiaoqiang@cust.edu.cn)

This work was supported in part by the Science and Technology Development Plan Project, Jilin, China, under Grant 20190302070GX, and in part by the Education Department of Jilin Province under Grant JJKH20190598KJ, Grant JJKH20190546KJ, and Grant GH180148.

ABSTRACT Network anomaly detection aims to identify network anomalies, and it has obtained many achievements using the supervised classification technique. Since the supervised classifier depends on the prior data, it is difficult to accurately classify the rare anomalies when they account less in the training set. Data augmentation can tackle the imbalanced training set problem through creating artificial rare anomaly samples. However, the existing data augmentation methods either ignore the data distribution or ignore the spatial knowledge between features. Therefore, this article addresses this issue by proposing a **N**etwork **A**nomaly **D**etection **S**cheme based on feature **R**epresentation and data **A**ugmentation (NADS-RA). Re-circulation Pixel Permutation strategy is first designed as feature representation strategy to construct images, and it rotates each feature left by the times of feature number to maintain the spatial knowledge between original network traffic features. An image-based augmentation strategy is thus designed to produce augmented images according to the distribution characteristics of rare network anomaly images with the help of Least Squares Generative Adversarial Network, which alleviates the effect of imbalanced training set and avoids over-fitting. After that, NADS-RA is implemented on the Convolutional Neural Network classification model. We conduct experiments on five public benchmark datasets, including NSL-KDD and UNSW-NB15, and so on, and compare against 12 detection methods and 17 data generation methods. The experimental results demonstrate the superior effectiveness of our work to state-of-the-art methods and the general applicability in different scenarios.

INDEX TERMS Anomaly detection, rare anomalies, feature representation, data augmentation, network security.

I. INTRODUCTION

With the fast development of the Internet, network security has become increasingly challenging. Network anomaly detection, as an effective scheme to identify the anomalous behavior, has received some achievements by supervised classification methods [1], [2]. Most supervised classification models depend on prior data, and they assume an equal distribution of training data [3]. However, real-world situations do not usually in line with such an assumption. For example, in the nine weeks of network connectivity data collected from a simulated US Air Force LAN [4], the number of the normal samples is more than 60,000, but that of user-to-root (U2R) attack is less than 100. In this case, U2R can be seen as a kind of rare anomalies. When the number of rare anomalies is less than that of the normal samples [5], [6], supervised methods are usually limited in classifying rare anomalies [7]. Generally speaking, in the training set, when the data of one class is significantly outnumbered by the data of at least another class, it can be considered imbalanced [8]. The classifiers are not generally prepared for the imbalanced training dataset, and they are likely to predict new coming samples as the majority class [9], and miss the real minority class that might be harmful to the network, like U2R attack [10]. Hence,

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

it is essential to classify rare anomalies from the imbalanced data.

Data augmentation is well-known to tackle the imbalanced classification by creating artificial rare data, but it is difficult to create realistic-looking data [11]. The commonly used Random Over-Sampling (ROS) and Synthetic Minority Oversampling Technique (SMOTE) [12] methods produce data along the line segment that joins rare data [13]. They over-sample data based on the local information rather than the overall distribution of the rare class, so the data generated by these methods might disturb the global distribution of the original data and further weaken the training effectiveness. Least Squares Generative Adversarial Network (LSGAN) [14] has been successfully applied in the image processing to produce similar images by learning the data distribution. It was firstly applied in network security to produce similar network traffic [15] by learning the data distribution of each feature independently, which loses the spatial knowledge among the features. Representing network traffic features as images might overcome this issue, but the existing network traffic feature representation strategies [10], [16]–[18] encode the features by One-Hot first, and then transform the encoded vectors into the pixel values, which disrupts the feature unity and the spatial knowledge of partial adjacent features. What's more, it is prone to over-fit [16]. Therefore, how to produce the realistic-looking data, upon maintaining the spatial knowledge of features and the distribution of data is especially challenging [19].

To solve this problem, we design an image-based data augmentation strategy. To the best of our knowledge, it is the first time that LSGAN is used to produce the augmented network anomalies on the basis of network traffic feature images. We rotate each feature left by the times of feature number to construct a circulant matrix as pixel values [20]. The raw network traffic features are thus represented as images, thereby maintaining the spatial knowledge. Subsequently, data augmentation strategy based on LSGAN is designed to over-sample the rare anomalies automatically according to the imbalance ratio, thus balancing the imbalanced training set and avoiding over-fitting. The Convolutional Neural Network (CNN) is trained on the obtained balanced dataset to learn the spatial knowledge of training data. Finally, a complete network anomaly detection scheme based on representation and augmentation (NADS-RA) is constructed in this article.

We conduct experiments on five public benchmark datasets including two well-known network anomaly detection datasets, namely, NSL-KDD [4] and UNSW-NB15 [21], and one credit card fraud detection dataset [22], and two software defect detection datasets [23], namely JM1 and PC5. The experimental results validate the effectiveness of NADS-RA, it not only improves the overall accuracy, but also decreases the False Negative Rate (FNR) of rare anomalies [24]. Besides, it suggests the superior performance to other state-of-the-art methods, and the general applicability of other areas.

The contributions of this study are summarized as follows:

(1) We design an image-based rare network anomalies augmentation strategy that not only maintains the spatial knowledge of network anomaly features, but also keeps the distribution of rare network anomalies.

(2) The experimental results of comparing with state-of-the-art methods suggest that the designed data augmentation strategy can alleviate the effect of imbalanced training set and further avoid over-fitting.

(3) The proposed NADS-RA is validated on five public benchmark datasets, including network traffic anomalies, software defects, and credit card frauds. The experimental results suggest that NADS-RA is adaptable in different areas.

The remaining of this article is structured as follows. Section II introduces the related work and Section III states the problem. Section IV illustrates the main methodology. The experimental results are described in Section V. Section VI concludes the full article and the future work.

II. RELATED WORK

This article summarizes related work about network anomaly detection based on CNN published in recent five years in Table.1. They are all implemented in the real public datasets. According to their experimental results analysis and described future work, as well as the real situation where the number of anomalies is less than that of the normal behavior, the main challenges of network anomaly detection based on deep learning technique can be summarized as feature representation and imbalanced classification. How to represent the network traffic as the images upon maintaining the spatial knowledge, and how to train an effective classifier on an imbalanced training dataset to classify the rare anomalies are the main problems in the reported works. Therefore, the related works are introduced from two perspectives, feature representation and imbalanced data classification, and they are marked in the second column ''Representation'' and the fifth column ''Balance'' of Table.1, respectively.

A. FEATURE REPRESENTATION

Deep learning technique has been widely used in images-related area since its layer-by-layer processing ability can automatically mine the hidden and high-level characteristics of the input images. To utilize deep learning techniques to improve the network anomaly detection, transforming the raw network traffic features into images is the first step. Many visualization techniques [31] have been designed, and they can be divided into three types: data filtration and transformation, graph theory, and pixel-based representation. (1) Among the data filtration and transformation techniques, principle component analysis (PCA) [32] is often used, but PCA is considered as an orthonormal linear transformation because it assumes all base vectors are orthonormal, so it is not recommended to use PCA for analyzing categorical data [31]. (2) Graph theory is suitable for the connection network that includes nodes and links, and often used in the network-communication-related scenarios. (3) On the

TABLE 1. A Brief Review of Related Work.

Study	Representation	Classes	Unity	Balance	Dataset
[9]	512*1	2	N	N	CSIC2010, CNTC2017, DARPA1998, KDD99
[10]	122 → 11*11	5	N	Y	KDD99
[16]	464 → 8*8*8bit	2	N	N	NSL-KDD
[17]	122*1	2	N	N	NSL-KDD, MAWILab, KyotoHoneypot
[18]	464 → 8*8*8bit	2, 3, 4, 5	N	N	NSL-KDD, UNSW-NB15
[25]	177*39	2 and 5	N	N	NSL-KDD
[26]	10*1	2(DoS)	N	N	NSL-KDD
[27]	7*7	2	Y	N	NSL-KDD
[28]	5*5	4 (-U2R)	N	N	NSL-KDD, UNSW-NB15
[29]	41*1	2 and 5	Y	N	KDD99
[30]	122*1	2 and 5	N	N	NSL-KDD, UNSW-NB15, et al.
Ours	41 → 37 → 37*37	2 and 5	Y	Y	NSL-KDD, UNSW-NB15

Classes: the number of distinct labels;
Unity: retain the structure of every feature or not;
Balance: solve imbalance issue or not

contrary, pixel-based representation technique opens the opportunities to apply deep learning techniques in the network anomaly detection [33] and can be used to analyze categorical data. It aims to change the feature elements into colored pixels, and it supports the fixed size data conversion.

This article investigates the anomaly detection based on network traffic features, so the pixel-based representation is deployed. To represent the network features as pixel-based images, different feature encoding methods have been explored. They can be deployed on the extracted features [4] or the raw payload contents [9] or the combination of both [34]. Payload contents are often seen as the natural language, and all the entities within the payload are encoded by the word embedding methods. To control the same length of encoded vectors, the short payloads are usually filled with zeros to obtain the same length of the longest payload [25]. For the extracted features, they are generally continuous or discrete values [26], [27]. Encoding discrete features [10], [17], [30] or encoding all features [16], [18] by One-Hot will also generate many zeros in the encoded vector. For both encoding methods, the encoded vectors are represented as a sequence of 0 and 1. And then every eight bits are transformed into a decimal pixel value. In this case, the representation process can be seen as a sequential operation of decomposition and combination. The raw features or raw contents are encoded into the binary space, and then transformed into the decimal space, which might destroy the unity of the raw feature or weaken the spatial relationship between the raw features. The related research is marked in the fourth column ''Unity'' of Table.1. Besides, there are also some representation methods that reshape the original features' long vectors into a pixel matrix directly [28], [29]. Though this kind of methods retains the unity of the feature, it will break the relationship between partial adjacent features.

Hence, a pixel-based image conversion strategy to retain the original spatial characteristics and the unity of features is required.

B. DATA AUGMENTATION

In the real world, the anomalies generally occur less frequently than the normal behaviour, so the number of anomalies and that of the normal will be imbalanced in the collected data. For example, in the nine weeks of network connectivity data collected from a simulated US Air Force LAN [4], the number of the normal samples is more than 60,000, but that of user-to-root (U2R) attack is less than 100. In this case, the supervised classifiers will be unable to learn the characteristics of rare anomalies so that they are likely to predict the rare anomalies as the normal, and might miss the real attack, which will make the system enter into a dangerous status. As shown in Table.1, the anomaly detection is studied in the form of binary classification or multi-classification. For the datasets that include many different attack labels, all the attack labels are treated as the anomaly class generally to perform the binary classification. No matter how many classes to be classified, the imbalance problem exists. The fifth column ''Balance'' indicates whether this research has solved the imbalance issue. It can be found that the issue has only been solved in a few studies.

To discover rare anomalies from imbalanced data, commonly used strategies are data re-sampling methods [35] and classifier modification methods [10]. The classifier modification methods aim to make the classifiers to be sensitive to the rare classes. A sequential classifier [7] containing five classifiers was proposed to identify a specific attack in sequence.

In spite of good performance, it needs expert knowledge to judge the intermediate classification results. On the contrary, the data re-sampling methods aim to equilibrate the imbalanced training dataset to enhance the characteristics of the rare anomalies before the classification. Since most classifiers are not generally prepared for the imbalanced training dataset, how to balance the training dataset in advance and then adapt the classifiers flexibly attracts many academic attentions.

Over-sampling strategies [3], [12], [13] that aim to generate the similar data to increase the proportion of rare classes has been widely used in the case where the number of rare classes is very tiny. For example, in NSL-KDD, the number of U2R is less than 100. They produce the similar data according to the distance between samples, but they ignore the data distribution. Generative Adversarial Networks (GAN) [36] and Least Squares GAN (LSGAN) [14] have been proved effective to learn the data distribution and produce the similar data. They are firstly applied in network security to generate network traffic data for enhancing the rare labeled raw packet streams and then the enhanced data are used to train a classifier to classify TCP streams [15], [37]. They provide a promising guide for the imbalanced classification, but they mimic each feature independently, which might lose the spatial knowledge between features.

Inspired by the above over-sampling methods applied in network anomaly detection, we absorb the advantage of LSGAN to learn the distribution characteristics of rare anomalies. Combined with the feature representation, an intelligent image-based augmentation method is designed. It not only keeps the characteristics of data distribution, but also maintains the spatial knowledge of features. NSL-KDD [4] and UNSW-NB15 [21] datasets are two well-known benchmark datasets, and are used most often, so they are also utilized in this article.

III. PROBLEM DESCRIPTION

According to the definition [8], any data set that exhibits an unequal distribution between its classes can be considered imbalanced. Analyzing the recent research [7], [9], [16]–[18], [25], [28]–[30], it can be found that the main challenge faced by many experiments is that some rare anomalies sometimes are difficult to be discovered even though the overall accuracy of imbalanced classification is high. Our goal is to precisely classify rare anomalies: Given a training set $D = \{N_1, \ldots, N_n, A_1, \ldots, A_a\}$, where n is the number of normal data and a is the number of abnormal data, by assuming that n is far more than a, $\frac{n}{a} \gg 1000$ is considered in this article, the problem is to train a classifier C, so that when a new abnormal sample comes, the model C can accurately predict whether this sample is abnormal or not.

To achieve this goal, constructing a balanced set through over-sampling technique has been widely used [3], [12], [13], [15]. Inspired by them, we focus on designing an effective data augmentation strategy through over-sampling the minority data. Thus our task can be summarized as follows: given an imbalanced training set, how to generate high-quality samples for augmenting the raw imbalanced training set.

Considering that most of the traditional data synthesis approaches either ignore the characteristics of data distribution [12], [13] or disrupt the spatial features within the data [15], [37], and these two issues have not been solved well in the state-of-the-art works. So it is necessary for us to provide a new solution to the data augmentation. To overcome the influence of imbalanced data on the supervised classifier in the network anomaly detection, how to produce the augmented data to further enhance the characteristics of rare network anomalies, upon maintaining the spatial knowledge within each anomaly data and keeping the distribution of rare anomalies is the main problem to be solved in this article. Besides, obtaining the augmented network anomalies data, how to control the mixing ratio of each class in the augmented training set is also explored in this article.

IV. NADS-RA: A SCHEME FOR NETWORK ANOMALY DETECTION

In this article, we design a network anomaly detection scheme NADS-RA to decrease the FNR of rare anomalies. The implementation details of NADS-RA are illustrated in Fig.1. Features are first extracted from the collected network packets captured by the tcpdump, and we validate NADS-RA on the public benchmark datasets that include the extracted features. Then, all the features are pre-processed by feature encoding, reduction, normalization and representation. Afterwards, data augmentation is used to balance the training dataset. Finally, the classifier CNN is trained on the balanced dataset and then evaluated on the new coming test data.

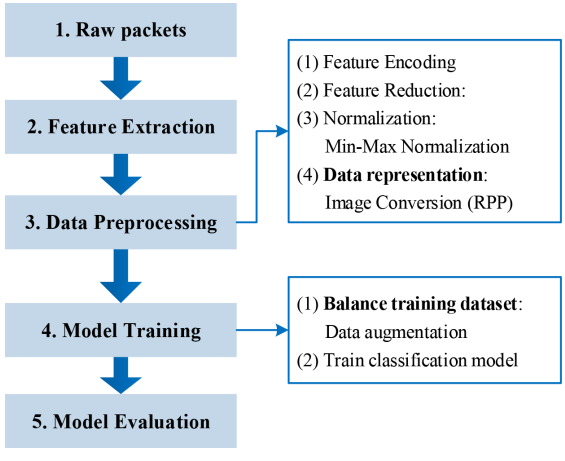


FIGURE 1. The framework of NADS-RA.

A. DATA PRE-PROCESSING

1) FEATURE ENCODING

There are some discrete features or symbolic features that cannot be directly accepted as the input of classifier, for example, protocol (tcp, udp, http). We first encode the

TABLE 2. Encoded Features After One-Hot.

Raw features	Encoded features		
protocol	f1	f2	f3
tcp	1	0	0
udp	0	1	0
http	0	0	1

discrete features into numerical vectors using One-Hot encoder. Each discrete feature is transformed into a N-bit digit that includes only one 1 and N-1 0, where N indicates the unique values number of this feature. Thus, a discrete feature is transformed into a sequence of binary digits. For example, in Table.2, "tcp" is encoded into (1 0 0), "udp" is encoded into (0 1 0) and "http" is encoded into (0 0 1). In this case, one symbolic feature ("protocol") is represented as three features ("f1", "f2" and "f3").

If one feature contains too many unique values (a bigger N), it will generate many zeros, and the encoded vectors are sparse with more zeros and less ones, which will influence the convolution and optimization effect of CNN. Besides, the feature dimension after One-Hot will increase. Take an example in Fig.2, assuming that there are five discrete features and they have different number of unique values ($N_1 = 2, N_2 = 4, N_3 = 2, N_4 = 5, N_5 = 3$), the dimension of encoded feature vectors obtained from One-Hot encoder is 16. Obviously, the feature dimension is increased from 5 to 16.

FIGURE 2. The example of binary feature bundling.

Therefore, bundling the binary vectors is executed to avoid the feature dimension increasing. We assume each discrete feature has an equal weight and do not consider the order of the feature. Then the bundling process transforms every eight-bit binary digits into one decimal value. Continue to see the example in Fig.2, 16 binaries obtained from One-Hot will be transformed into two decimal values. In this case, the feature dimension is reduced from 16 to 2. In the process of bundling, we try to avoid splitting one discrete feature's binary vectors into two decimal values, so it can maintain the integrity of each feature.

2) FEATURE REDUCTION

Later, to optimize the remaining continuous features, a feature filter is designed to remove the useless. As the dimensions of features are different, the standard deviation is inappropriate to compare discreteness of features, so the coefficient of variance C_v, as a type of classical statistical theory is introduced, and the computation is defined as Equation 1.

$$C_{vi} = \frac{\sigma_i}{\mu_i} * 100\% \tag{1}$$

where σ_i and μ_i are standard deviation and mean of i^{th} feature. Generally, a higher C_v indicates a higher discreteness, and the feature of a higher C_v plays a more important role. Specially, when the mean μ_i of i^{th} feature is zero, this feature will be seen as unimportant relatively.

3) DATA NORMALIZATION

Normalization can eliminate differences among diverse dimensional data, so it is therefore widely used in machine learning. Because features of different scales will result in unreliability of training model, we normalize them in the same range. Rescale-min-max normalization is used in this article as Equation 2.

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}} * (1 - a) + a \tag{2}$$

where x_{max} and x_{min} represent the maximum and minimum value of feature x_i respectively, x_i and x_i' represent the raw feature and the normalized feature respectively. To avoid too many zeros within the feature matrix, we rescale the range of the normalized feature from $[0, 1]$ to $[a, 1]$ where the indicator a is a predefined nonzero parameter, $a \in (0, 1)$. Afterwards, the minimum value of the normalized variable will be changed into a.

4) IMAGE REPRESENTATION

To learn the deep characteristics of traffic feature automatically, we first convert traffic feature vectors into 2-dimension (2D) pixel-based images and then construct 3-channel images [20]. A Re-circulation Pixel Permutation (RPP) strategy is designed as Equation 3, it is used to convert a long vector into a circulant matrix, where x_i is the i^{th} sample, and it is an original long vector with M elements. x_i' is obtained by moving every element $x_{ij}(j = 1, 2, \cdots, M)$ of x_i one unit forward every time, then x_i' is used to represent pixel values of the transformed image whose dimension is $M * M$. Afterwards, every pixel value is extended to the RGB image pixel by adjusting the pixel value to different percentage on the same position for 3 channels.

$$x_i = [x_{i1}, x_{i2}, \ldots, x_{iM}] \rightarrow \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{iM} \\ x_{i2} & \cdots & x_{iM} & x_{i1} \\ \cdots & x_{iM} & \cdots & \cdots \\ x_{iM} & x_{i1} & x_{i2} & \cdots \end{bmatrix}_{M*M} = x_i' \tag{3}$$

Compared with the representation approach that reshapes a long vector into a square matrix directly, RPP retains the original spatial structure of sample, and every sub-image of the converted images consists of the adjacent elements of the original vector.

B. DATA AUGMENTATION

From the perspective of data sampling, data augmentation aims to increase the number of minority samples to make the imbalance ratio closer to 1. The imbalance ratio is defined as Equation 4, where N^i_{min} and N_{max} represents the number of i^{th} class of minority samples and the maximum number of majority class samples, respectively.

$$\gamma_i = \frac{N^i_{min}}{N_{max}} \tag{4}$$

A dynamic data synthesis strategy is designed in this article with the consideration of data distribution. As shown in Fig.3, it includes two roles, generator and discriminator. The generator generates random vector z first, and then samples partial data that obeys the distribution $z \sim p_z$. The discriminator judges the reality of z by comparing them with the real data that obeys the distribution $x \sim p_{data}$. If the discriminated result is false, the generator will refine its generating algorithm to make the generated data more similar to the real data until the discriminator cannot judge the generated data, thus they will be output.

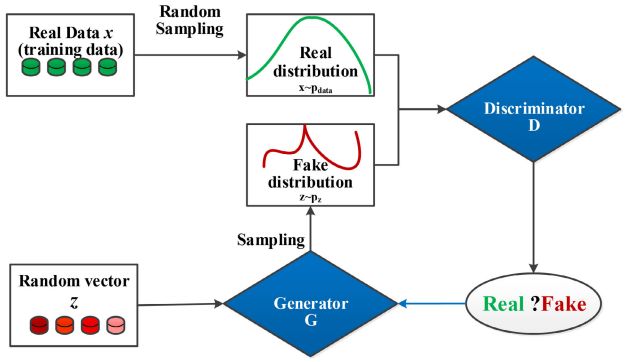


FIGURE 3. The structure of data augmentation strategy.

To ensure the distribution of generated samples is realistic looking to that of the real samples, original samples of the same class are fed into the generation model in a batch every time. The generation model will be executed multiple times dynamically until the number of generated samples is not less than that of the majority. The termination condition is $\gamma_i > r$ where r is a threshold to control the amount of generated samples.

The data augmentation process is illustrated in Algorithm.1, it will train the discriminator D D_steps times first when parameters of the generator G are fixed, then train the generator G G_steps times when parameters of the discriminator D are fixed. In most cases, D_steps is greater than G_steps in order to conduct a better G. Finally, the generative data from G will be output.

$$\min_D J(D) = \min_D \{\frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[D(x)-1]^2 + \frac{1}{2}\mathbb{E}_{z\sim p_z}[D(G(z))]^2\} \tag{5}$$

Algorithm 1 Data Augmentation Algorithm

Input: Fake sample: random noise data $z(dim\_z)$; Real sample: $x(dim\_x)$;
Parameters: batch size (mb_size), training steps (G_steps, D_steps), training times (t_n);
Loss function: least square loss function;
Optimization solver: Adam optimizer;
for iteration in t_n **do**
  while $i < D\_steps$ **do**
    Train D in the unit of mb_size
    Minimize discriminator's loss function in Equation 5
  end while
  while $j < G\_steps$ **do**
    Train G in the unit of mb_size
    Minimize generator's loss function in Equation 6
  end while
end for
Output: generative data from G

$$\min_G J(G) = \min_G \{\frac{1}{2}\mathbb{E}_{z\sim p_z}[D(G(z))-1]^2\} \tag{6}$$

The data augmentation strategy increases the proportion of minority class in dataset, and tries to maintain the data distribution in the same class. The enlarged training dataset is nearly balanced and is used to train the classification model to perform the final anomaly detection task.

C. CLASSIFICATION MODEL

After feature representation and data augmentation, a classification model is trained on the balanced training dataset and further used to validate the NADS-RA. Convolution neural network (CNN), as a type of deep learning algorithms, has achieved great classification performance in learning the spatial knowledge of images. This article uses CNN to extract spatial characteristics of network traffic features, and then compare with other methods.

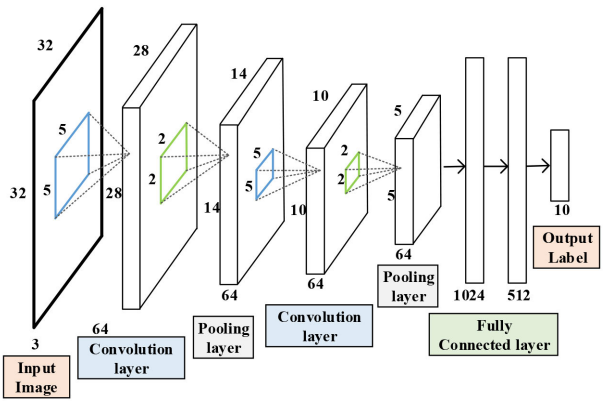


FIGURE 4. The architecture of CNN.

The architecture of the commonly used CNN is shown in Fig. 4. A complete CNN model contains multiple

convolution layers and pooling layers. Convolution layer mines the local spatial knowledge using the moving convolution kernel whose size is set as 5∗5, and pooling layer reduces the dimension of images using pooling kernel whose size is set as $2 * 2$ in Fig. 4. Repeating the convolution and pooling operations, the spatial knowledge is obtained through layer-by-layer processing. Finally, one or more fully connected layers are used to accept the learned knowledge for decision making, and the predicted label is output.

D. OVERALL WORKFLOW

The execution of NADS-RA is illustrated in Algorithm.2. There are four main steps: data preprocessing, feature representation, data augmentation and classification model training. Data preprocessing includes feature encoding, feature reduction and normalization. Feature representation is performed on training, validation and test datasets. Judging from the imbalanced ratio of the training dataset, if it is checked as imbalanced, data augmentation will be executed as Algorithm.1 to generate the synthesis data. Through mixing the original training dataset with the synthesis data, a balanced dataset will be constructed to train the classification model.

Algorithm 2 NADS-RA Workflow

1: Input: Dataset D = training data, validation data, test data;
2: **Preprocess:** Feature Reduction, Data Normalization;
3: **Representation:**
4: **while** $x \in$ D **do**
5: $x'(1,:) = x = [x_1, x_2, \cdots, x_n]$
6: **for** $1 \leq i \leq n$ **do**
7: $x'(i+1,:) = [x_{i+1}, x_{i+2}, \cdots, x_n, x_1, \cdots, x_i]$
8: **end for**
9: **end while**
10: An image dataset D' is represented.
11: **Augmentation:** Calculate the imbalance ratio: γ_i;
12: **if** $\gamma_i < 1$ **then**
13: perform **Data augmentation algorithm**.1 → produced samples
14: update training data ← training data + produced samples
15: **end if**
16: $x' \in D'$: **Train** → **Validation** → **Test**
17: Output: acc_test, loss_test, confusion matrix and test report

V. EXPERIMENT

A. EXPERIMENT CONFIGURATION

All the experiments are conducted on an Ubuntu 16.04 LTS machine with Intel Xeon (R)W-2123, 3.6GHz CPU, GeForce GTX TITAN Xp COLLECTORS EDITION GPU and 12GB VRAM. We use ResNet50 as the CNN model for evaluation. 50 epochs with a batch size of 1024 are used to train ResNet50 where the Adaptive moment estimation (Adam) optimizer is utilized, learning rate is 1e-3 and cross entropy is used as the cost function. All datasets used in this article are collected from different application scenarios of the real world, and they have been labeled as normal and abnormal or specific attack type. Except for the NSL-KDD and UNSW-NB15, which have been divided into the training set and test set, 80% of the other datasets are selected randomly for training, and the remaining are used for testing. The experimental results are averaged from 100 groups of experiments without special description.

1) DATASETS

We evaluate NADS-RA on five public datasets: NSL-KDD [4] and UNSW-NB15 [21] are two well-known network datasets and used as the main datasets. JM1 and PC5 [23] are two software defect detection datasets, they and Credit card [22] dataset are used to validate the general applicability of NADS-RA in other scenarios.

a: NSL-KDD DATASET

There are four subsets in NSL-KDD [4], namely KDDTrain$^+$, KDDTrain$^+$_20 percent, KDDTest$^+$ and KDDTest^{-21}. There are 41 features and 5 labels including one normal type and four attack types: Denial of Service (DoS), Probe, User-to-Root (U2R) and Remote-to-Login (R2L). As shown in Table. 3, normal traffic accounts for more than half, but U2R and R2L account for only 0.04 and 0.79 percent in KDDTrain$^+$ set.

TABLE 3. Details of the NSL-KDD Dataset.

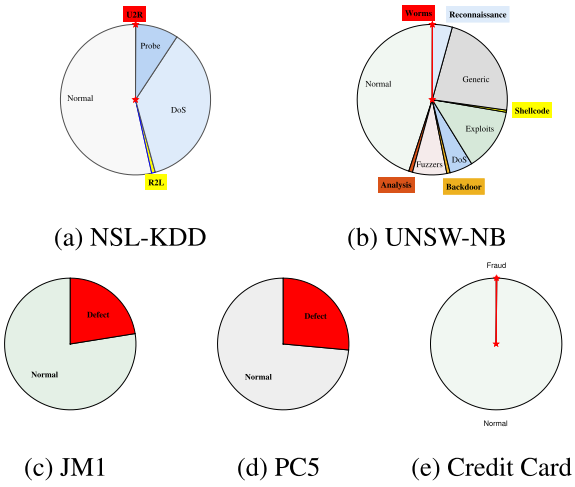
Label		KDDTrain$^+$	Ratio(%)	KDDTest$^+$	KDDTest^{-21}
Normal		67345	53.46	9711	2152
Attack	DoS	45926	36.46	7458	4342
	Probe	11655	9.25	2421	2402
	R2L	995	0.79	2754	2754
	U2R	52	0.04	200	200
Total		125973	100	22544	11850

b: UNSW-NB15 DATASET

It contains a large number of recent, legitimate and malicious network instances, and it contains about 100GB of data including 2,540,044 records which are stored in four CSV files [21]. There are 42 features and ten types of labels (one normal type and nine malicious types, Worms, Reconnaissance, Generic, Shellcode, Exploits, DoS, Backdoor, Fuzzers and Analysis). It can be seen from Fig. 5 (a) that the normal traffic accounts for more than half, but Worms, Backdoor and Shellcode account for less than 0.1 percent.

c: JM1 AND PC5 DATASETS

They are two software defect datasets of NASA MDP project [23]. JM1 dataset has 22 static code attributes and



(a) NSL-KDD (b) UNSW-NB

(c) JM1 (d) PC5 (e) Credit Card

FIGURE 5. Distribution of five real Datasets.

8904 distinct modules out of 10878 modules. There are 2001 positive defect modules (account for 22.473 percent) in this dataset. PC5 dataset has 39 static code attributes and 1830 distinct modules out of 17186 modules. There are 484 positive defect modules (account for 26.448 percent) in this dataset.

d: CREDIT CARD DATASET

It contains 284807 credit card transactions made over two days in September 2013 by European cardholders [22]. After removing duplicates, there are 446 positive fraudulent transactions (account for 0.157 percent) in this dataset. Every sample is represented as 28 numerical features.

The distribution details of the five datasets are shown in Fig. 5, it can be found that NSL-KDD and these four datasets all have serious class imbalance problems. Therefore, they are all used to evaluate NADS-RA's effectiveness. NSL-KDD and UNSW-NB15 are the main datasets, and the other three datasets are used to evaluate the general applicability of NADS-RA.

2) METRICS

A good anomaly detection approach requires high true rate as well as low false rate. The metrics are calculated using the confusion matrix in Table 4, where TP (True Positive) and TN (True Negative) mean the number of positive instances (referred to Anomaly) and negative instances (referred to Normal) that are correctly classified, and FN (False Negative) and FP (False Positive) mean the number of positive instances and negative instances that are incorrectly classified.

We evaluate the performance of our approach by the following metrics, Precision, Recall, F1, False Positive Rate (FPR) and False Negative Rate (FNR) and Gmean as well as AUC. We report nearly all these metric values since it is widely agreed that the accuracy alone is unable to provide an accurate evaluation of the classification performance, especially for imbalanced datasets.

TABLE 4. Confusion Matrix of Binary-Classification.

		Prediction	
		Anomaly	Normal
Target	Anomaly	TP	FN
	Normal	FP	TN

Precision is the ratio of true positive samples to the samples that are labeled by the system as positive. It represents the confidence of retrieval. Thus, it should be as maximum as possible.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

Recall, also called as Detection Rate (DR), is the ratio of true positive samples to the real positive samples. It represents the completeness of retrieval, and it is a core metric commonly used to measure the quality of the anomaly detection under consideration. Thus, it should be as maximum as possible.

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$F1$ is defined as the harmonic mean of Precision and Recall. It represents a synthesis of the performance of retrieval. The higher value of $F1$ indicates that the approach performs better on Recall and Precision. Thus, it should be as maximum as possible.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

False Negative Rate (FNR) is the ratio of false negative samples to the real positive samples. It represents the inability to detect the real positive. If this value is high, the real attacks will be missed, which makes the system to be exposed to the malicious users and enter into a dangerous status. Thus, it should be as minimum as possible.

$$FNR = \frac{FN}{TP + FN} \tag{10}$$

False Positive Rate (FPR), also termed as False Alarm Rate (FAR), is the ratio of false positive samples to the real negative samples. If this value is consistently elevated, the security analysis operator will intentionally disregard the system warnings, which makes the system to enter into a dangerous status [38]. Thus, it should be as minimum as possible.

$$FPR = \frac{FP}{FP + TN} \tag{11}$$

Accuracy is the most used metric from the overall view. It is the ratio of correctly classified samples to the total samples. It represents the confidence of the classification. Thus, it should be as maximum as possible.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{12}$$

Area under Curve (AUC), is the ability to avoid false classification. It can be approximately seen as the arithmetic mean of DR (Recall) and TNR (1-FPR) as Equation 13, and it represents a good compromise between DR (Recall) and FPR metrics [39]. It is effective in measuring the performance of classifiers for imbalanced data [40]. Thus, it should be as maximum as possible.

$$AUC = (DR(Recall) + (1 - FPR))/2 \tag{13}$$

Gmean, indicates the geometric mean of sensitivity and specificity, where $sensitivity = \frac{TP}{TP+FN}$ and $specificity = \frac{TN}{TN+FP}$, and it can also be seen as the comprehensive measurement of Recall and FPR. Thus, it should be as maximum as possible.

$$Gmean = \sqrt{sensitivity * specificity} \tag{14}$$

In the binary-classification task, these metrics are used directly. In the multi-classification task, the overall metric is computed by weighted average to judge the overall effectiveness of multi-type attack detection comprehensively in this article.

3) OUTLINES

We conduct four groups of experiments in total as shown in Fig. 6.

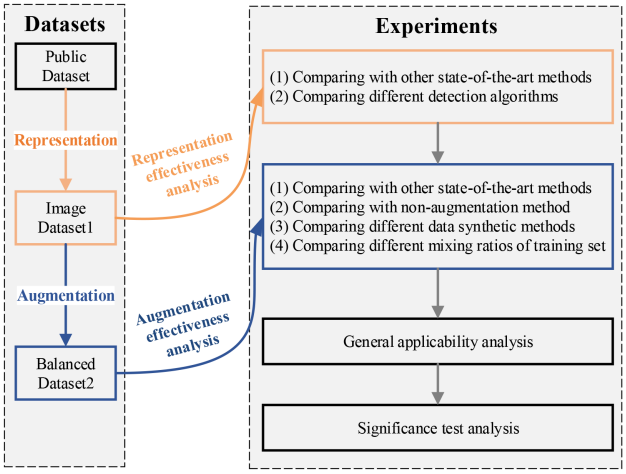


FIGURE 6. The outline of experiments.

Experiment 1: After feature representation, the raw data are represented as images (marked as Image Dataset1). Since the number of each specific attack and that of the normal is imbalanced, we label all the attacks as Anomaly to avoid the influence of imbalance on representation. Then, a binary-classification task for identifying anomalies from the normal is abstracted to validate the effectiveness of the representation strategy of NADS-RA. The experiments include comparing with state-of-the-art representation methods, and comparing with different detection algorithms.

Experiment 2: To detect multiple types of attack simultaneously, multi-classification task is needed. To improve the detection accuracy of rare anomalies in the raw imbalanced dataset, the imbalanced Image Dataset1 is re-built by data augmentation strategy of NADS-RA, and then the balanced dataset is marked as Balanced Dataset2. The Balanced Dataset2 is used to evaluate the effectiveness of augmentation. A multi-classification task is abstracted for classifying the known attack. The experiments include comparing with state-of-the-art methods, and non-augmentation methods, and different data synthesis methods, as well as different mixing ratios of training sets.

Experiment 3: NADS-RA focuses on network anomaly detection, and the effectiveness is validated by two public network datasets, namely NSL-KDD and UNSW-NB15. Besides, we explore to apply it in other scenarios to evaluate its general applicability, such as software defect detection and credit card fraud detection.

Experiment 4: Statistical significance tests are conducted to compare the performances of various approaches on multiple datasets.

B. REPRESENTATION ANALYSIS

We first construct the image datasets on raw NSL-KDD [4] and UNSW-NB15 [21] datasets using the feature representation strategy of NADS-RA, and then abstract the anomaly detection as a binary classification problem on image datasets. The comparison test includes two experiments: comparing with other representation methods, and comparing different detection algorithms.

1) COMPARING WITH OTHER REPRESENTATION METHODS

We compare our approach with those reported results in other studies. Among them, supervised methods including convolution neural networks (CNN) [16], [17], deep neural networks (DNN) [41], and unsupervised methods including clustering [42]–[46] are state-of-the-art methods. CNN and DNN are recent methods based on feature representation. Our approach can represent original feature vectors as pixel-based images with spatial knowledge remained. Other deep learning methods can also represent the feature vectors as images but cannot maintain the spatial knowledge and feature unity. We implement the baseline methods according to the descriptions provided in the appropriate papers [16], [17], [41]–[46] and compare NADS-RA with these methods using metrics: accuracy, precision, recall, F1, Gmean, FPR, FNR and AUC.

The results on NSL-KDD and UNSW-NB15 are shown in Tables.5 and 6. The overall classification measurements of our approach are relatively better than that of the other methods. Though previous methods [16], [17] take CNN as the classifier as well, they encode all features [16] or symbolic features [17] by One-Hot encoder, and then take the encoded vectors as input directly. There are too many zeros in the represented sparse vectors that influence the optimization and convolution effect. On the contrary, we only encode the symbolic features by One-Hot and subsequently bundle the binary bits together into the decimal value, which alleviates the influence of massive zeros on optimization and

TABLE 5. The Comparison Results Of Binary-Classification of NSL-KDD.

Test set	Methods	Acc	Precision	Recall	F1	FPR	FNR	Gmean	AUC
$test^{+}$	**Ours**	**0.813**	**0.97**	0.7	**0.81**	**0.031**	**0.3**	**0.823**	**0.834**
	[16]-CNN	0.791	0.92	0.694	0.791	0.08	0.306	0.799	0.807
	[41]-DNN	0.801	0.692	**0.969**	0.807	0.326	0.031	0.808	0.821
	[17]-CNN	0.768	-	-	-	-	-	-	-
	[42]-Decision Tree	0.64	0.629	0.4	0.49	0.179	0.599	0.574	0.611
	[42]-Rule Induction	0.637	0.624	0.395	0.484	0.18	0.605	0.569	0.608
	[42]-Nearest Neighbour	0.628	0.599	0.413	0.489	0.209	0.587	0.571	0.602
	[42]-Naive Bayes	0.558	0.485	0.434	0.458	0.348	0.566	0.532	0.543
$test^{-21}$	**Ours**	0.647	**0.96**	0.51	0.67	**0.129**	0.49	**0.666**	**0.69**
	[16]-CNN	**0.816**	0.818	**0.996**	**0.898**	0.998	**0.004**	0.043	0.499
	[17]-CNN	0.556	-	-	-	-	-	-	-
full set	**Ours**	**0.988**	0.984	**0.991**	0.988	**0.014**	0.009	0.985	0.989
	[43] - SVM	0.850							
	[44] - unsupervised	0.841							
	[45]-DAE+DFFNN	0.986	-	0.99		0.018			

TABLE 6. The Comparison Results of Binary-Classification of UNSW-NB15.

Methods	Acc	Precision	Recall	F1	FPR	FNR	Gmean	AUC
Ours	**0.949**	**0.982**	0.925	**0.953**	**0.021**	**0.075**	**0.952**	**0.952**
[41]-DNN	0.784	0.944	0.725	0.82	-	0.275	-	0.934
[45]-DAE+DFFNN	0.924	0.92	**0.93**	0.925	0.082	0.07	0.924	0.924
[46]-Random forest	0.757	0.75	0.76	0.73	-	0.24	-	-
[46]-XG Boost	0.717	0.69	0.72	0.67	-	0.28	-	-
[46]-Bagging metaestimator	0.751	0.74	0.75	0.72	-	0.25	-	-
[46]-Decision tree	0.745	0.75	0.74	0.72	-	0.26	-	-
[46]-KNN	0.744	0.74	0.74	0.72	-	0.26	-	-

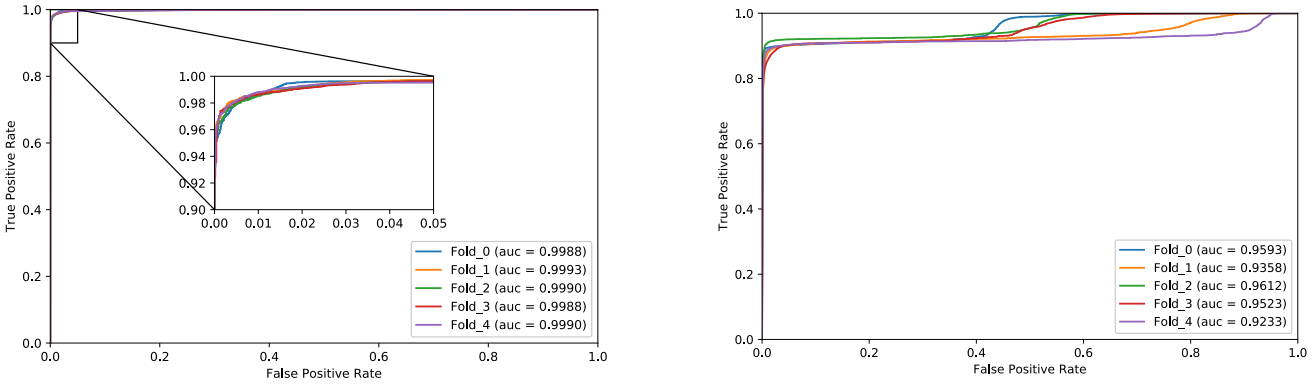
convolution. Specially, in the comparison results on $test^{-21}$, method of research [16] obtains the highest Accuracy, Recall, F1 and the lowest FNR. However, its FPR is 0.998. Combining its overall measurements, it occurs the over-fitting, nearly all the test samples are detected as the anomaly, which leads to the imbalanced results. The number of the anomaly samples account for more than 80% leads to that the accuracy is about 0.816. Hence, its biased results cannot reflect the generalization ability. Compared to the methods [42]–[46] that do not involve the representation, the results obtained from the clustering methods are almost the lowest, which further suggests the superior performance of our approach.

We additionally utilize the full NSL-KDD set and UNSW-NB15 dataset to evaluate the generalization ability of NADS-RA. The Receiver Operating Characteristic (ROC) curves of 5-Fold cross validation test are shown in Fig. 7. It can be found that the detection results of five groups are close. So NADS-RA has a better generalization ability.

Considering all metrics, we come to a conclusion that our approach has clear advantages in feature representation. We can maintain the spatial knowledge of original feature vectors, and further contribute to training an effective CNN classifier. The comparison of the experimental results obtained from other state-of-the-art works deeply show the superior performance.

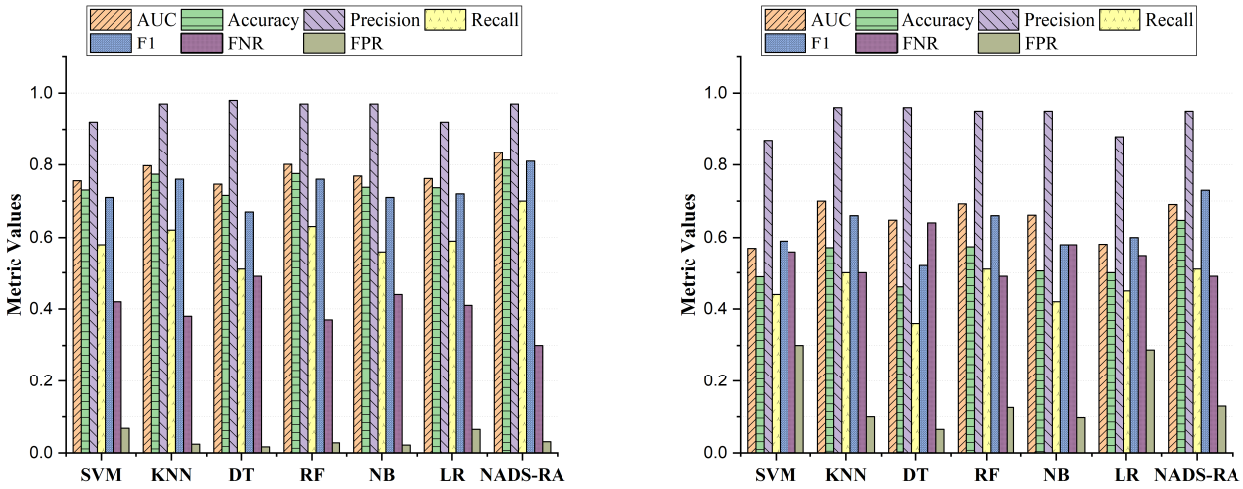
2) COMPARING WITH DIFFERENT TRADITIONAL MACHINE LEARNING ALGORITHMS

NADS-RA trains ResNet50 using represented images and then detects the anomalies. To judge the general adaptability of the feature representation method, various detection algorithms are tested on the NSL-KDD dataset. Since the representation methods combined with state-of-the-art deep learning classifiers have been compared in the last subsection, we implement different traditional machine learning classifiers and evaluate their performance. Figure.8 shows a



(a) NSL-KDD full set (b) UNSW-NB15

FIGURE 7. ROC curves of 5-Fold cross validation on NSL-KDD and UNSW-NB15.



(a) test^{+} (b) test^{-21}

FIGURE 8. Comparison results of different detection algorithms on NSL-KDD.

comparison of experimental results obtained from two test sets of NSL-KDD dataset.

The X-axis locates seven approaches: support vector machine (SVM), k-Nearest Neighbor (KNN), Decision Tree (DT), Random Forest (RF), Naive Bayesian (NB), Logistic Regression (LR) and NADS-RA. Y-axis indicates seven metrics. It shows that our approach yields the highest AUC, Accuracy, Precision, Recall as well as F1, and the lowest FNR, while FPR is the third lowest. When performing data fitting, deep learning models can extract more complex features than traditional machine learning models and mine the hidden characteristics of the samples. Hence, deep learning models have better representation ability than the shallow learning models [9]. Considering all metrics used in these experiments, we can find that our approach performs globally better than other traditional machine algorithms.

C. AUGMENTATION ANALYSIS

To validate the data augmentation effect of NADS-RA, we conduct four goups of multi-classification experiments on two real imbalanced datasets, namely NSL-KDD [4] and UNSW-NB15 [21]. The comparison test includes comparing with state-of-the-art works, evaluating the necessity of data augmentation, comparing different data augmentation methods and comparing different mixing ratios of training set.

1) COMPARING WITH OTHER WORKS

We first compare our approach to other state-of-the-art works. Tables. 7 and.8 show the multi-classification results on NSL-KDD full set and UNSW-NB15 dataset. Since only partial metrics are used in the reported works, we exhibit the same metric values. Data augmentation strategy of NADS-RA aims at creating new similar samples and then injecting them into the original training set to clarify the

TABLE 7. The Multi-Classification Results of Different Methods on NSL-KDD Full Set.

		Precision	Recall	F1	FNR	FPR	AUC	Acc
Ours	U2R	0.597	0.511	0.551	0.489	0.001	0.755	
	R2L	0.74	0.883	0.805	0.117	0.013	0.935	
	Probe	0.95	0.964	0.957	0.036	0.006	0.979	
	DoS	0.986	0.975	0.981	0.025	0.008	0.984	
	Normal	0.971	0.96	0.965	0.04	0.028	0.966	
Ours	**Overall**	0.849	**0.859**	0.852	**0.141**	**0.011**	**0.924**	**0.962**
Others	[47] - sNDAE	**1**	0.854	**0.874**	0.146	0.146	0.856	0.854
	[18] - CNN	0.792	0.784	0.779	0.216	-	-	0.911
	[42]-kMeans	-	-	-	-	0.23	-	0.654
	[42]-kMedoids	-	-	-	-	0.215	-	0.767
	[42]-EMclustering	-	-	-	-	0.207	-	0.781
	[42]-Distance	-	-	-	-	0.211	-	0.802

TABLE 8. The Recall Values of Different Multi-Classification Methods on UNSW-NB15.

Label	Normal	Generic	Exploit	Fuzzers	DoS	Reconnaissance	Analysis	Backdoor	Shellcode	Worms	Overall
Number	37000	18871	11132	6062	4089	3496	1346	583	378	44	83001
[18]	**0.997**	**0.977**	0.618	0.068	0	0	0	0	0	0	0.754
[48]	0.934	0.863	0.794	0.528	**0.896**	0.556	**0.834**	0.638	0.487	0.478	0.846
[49]	0.942	0.877	**0.81**	0.528	0.881	0.572	0.811	0.713	0.522	0.483	0.855
Ours	0.98	0.964	0.731	**0.617**	0.744	**0.645**	0.497	**0.861**	**0.894**	**0.614**	**0.882**

characteristics of rare anomalies more evidently within the same or nearly-same distribution. On the contrary, clustering [42] or deep learning methods [18], [47] do not involve the data augmentation, so the detection model cannot learn enough characteristics from the original raw rare data. sNDAE method proposed in [47] obtains the best precision and F1, but it's worth noting that its FPR is more than 14.6%, yet the FPR of ours is only 1.1%. Consequently, we work well in learning knowledge from the rare anomalies. The comparison results show that we have an obvious advantage on the overall classification performance.

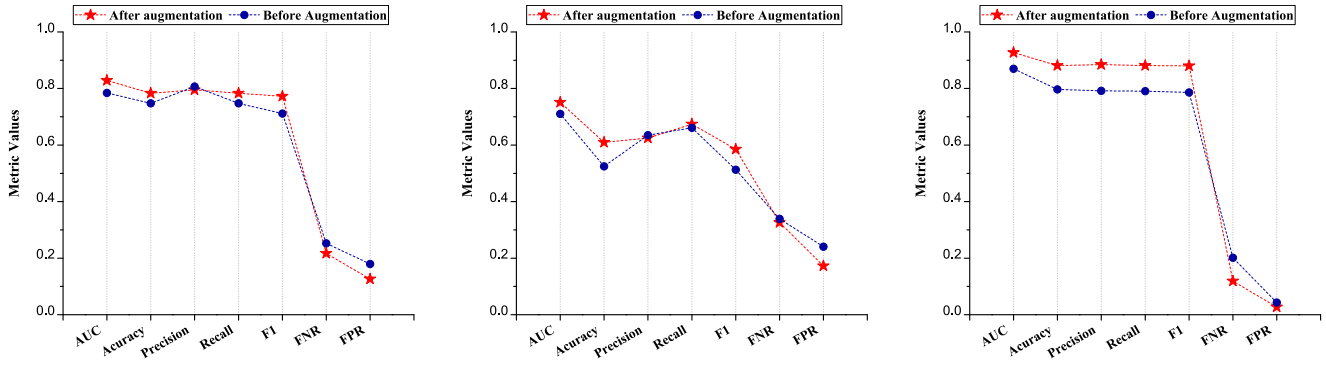
2) COMPARING WITH NON-AUGMENTATION METHOD

To confirm the necessity of augmentation for detection method, this subsection trains the same classifier on the augmented balanced dataset and raw imbalanced dataset (marked as "After augmentation" and "Before augmentation", respectively) using NSL-KDD [4] and UNSW-NB15 [21] datasets. The multi-classification results are exploited from the perspectives of AUC, Accuracy, Precision, Recall, F1, FNR and FPR.

Figure. 9 plots the results obtained from two test sets of NSL-KDD and UNSW-NB15 dataset, respectively. Red star and blue circle symbols indicate the results obtained from NADS-RA which is trained after augmentation and before augmentation, respectively. For the experiment on UNSW-NB15, 80% of the full set are selected for training, and the remaining are used for test. It can be found that for all test sets, the AUC, Accuracy, Precision, Recall, F1 measurements are improved, and the values of FNR and FPR are both decreased. The trends of all these metrics have demonstrated that the data augmentation is effective for improving true rates and decreasing false rates compared with non-augmentation method. Therefore, it's necessary to augment the imbalanced training dataset for pursuing a better detection result.

The detailed detection results for each class of NSL-KDD and UNSW-NB15 are shown in Table. 9. There are two values separated by "/" in each cell, and they indicate the value obtained after augmentation and before augmentation, respectively. For all classes, especially for the attacks, the global metrics, such as F1, AUC and Accuracy, have improved after augmentation. Observations can be found:

The most obvious observations of NSL-KDD can be found in the first three rows, where "U2R", "R2L" and "Probe" attack detection results are given. The FNR of them has been decreased by 14.5%, 29.4% and 15.2%, respectively on both two test sets. It has almost no influence on the other classes. This phenomenon can be reasoned by the data augmentation that increases the proportion of "U2R" and "R2L" in the training set without information loss, and simultaneously clarifies the distribution of rare classes, which then facilitates



(a) test^{+} of NSL-KDD (b) test^{-21} of NSL-KDD (c) UNSW-NB15

FIGURE 9. Augmentation necessity on NSL-KDD and UNSW-NB15.

TABLE 9. The Data Augmentation Necessity Analysis Results of NSL-KDD and UNSW-NB15.

TestSet		Label	Precision	Recall	F1	FNR	FPR	AUC	Acc
NSL-KDD	test^{+}	U2R	0.58/0	0.145/0	**0.232**/0	0.855/1	0.001/0	**0.572**/0.5	**0.783** /0.748
		R2L	0.689/0.98	0.383/0.089	**0.492**/0.164	0.617/0.911	0.024/0	**0.679**/0.545	
		Probe	0.732/0.838	0.813/0.661	**0.77**/0.739	0.187/0.339	0.036/0.015	**0.889**/0.823	
		DoS	0.944/0.961	0.734/0.737	0.826/**0.834**	0.266/0.263	0.022/0.015	0.856/**0.861**	
		Normal	0.731/0.649	0.94/0.98	**0.823**/0.781	0.06/0.02	0.261/0.401	**0.839**/0.789	
	test^{-21}	U2R	0.617/0	0.145/0	**0.235**/0	0.855/1	0.002/0	**0.572**/0.5	**0.609** /0.525
		R2L	0.749/0.98	0.383/0.089	**0.506**/0.164	0.617/0.911	0.039/0.001	**0.672**/0.544	
		Probe	0.753/0.849	0.811/0.659	**0.781**/0.742	0.189/0.341	0.068/0.03	**0.872**/0.814	
		DoS	0.901/0.921	0.557/0.557	0.689/**0.694**	0.443/0.443	0.035/0.028	**0.761**/0.765	
		Normal	0.346/0.278	0.823/0.917	**0.487**/0.426	0.177/0.083	0.346/0.53	**0.739**/0.694	
UNSW-NB15		Fuzzers	0.663/0.492	0.617/0.5	**0.619**/0.496	0.383/0.5	0.03/0.041	**0.794**/0.73	**0.882** /0.797
		Analysis	0.918/0.286	0.497/0.038	**0.645**/0.068	0.5/0.962	0.0007/0.001	**0.748**/0.519	
		Backdoor	0.539/0.212	0.816/0.861	**0.663**/0.34	0.139/0.139	0.005/0.023	**0.928**/0.919	
		DoS	0.573/0.325	0.744/0.294	**0.648**/0.309	0.256/0.706	0.028/0.032	**0.858**/0.631	
		Exploits	0.851/0.66	0.731/0.46	**0.787**/0.542	0.269/0.54	0.019/0.037	**0.856**/0.711	
		Generic	0.962/0.914	0.964/0.964	**0.963**/0.939	0.036/0.036	0.011/0.027	**0.977**/0.969	
		Recon	0.735/0.48	0.645/0.579	**0.687**/0.525	0.354/0.421	0.01/0.028	**0.817**/0.776	
		Shellcode	0.947/0.985	0.894/0.849	**0.92**/0.912	0.11/0.151	0.0002/0	**0.947**/0.925	
		Worms	1/0	0.614/0	**0.761**/0	0.386/1	0/1	**0.807**/0.5	
		Normal	0.952/0.931	0.98/0.951	**0.965**/0.941	0.02/0.049	0.04/0.058	**0.97**/0.947	

After augmentation / Before augmentation

the classification model to better learn the knowledge of them during the training process.

From a global view of all metrics of UNSW-NB15, the augmentation effectiveness is shown more obviously on the "Analysis" and "Worms" attack detection. For "Analysis", it corresponds with the nearly-least F1 and the nearly-biggest FNR before augmentation, and in comparison, F1 has been improved by nearly 60 percent and FNR has been decreased by nearly 50 percent after augmentation. For "Worms", it corresponds with the least F1 and the biggest FNR before augmentation, and in comparison, F1 value has been improved by 76 percent, and FNR value has been decreased by nearly 61 percent after augmentation.

In all, data augmentation of NADS-RA is necessary for detecting rare anomalies, and it is promising to alleviate the influence of imbalance on FNR of rare anomalies detection.

3) COMPARING WITH DIFFERENT DATA SYNTHESIS METHODS

The effectiveness of data augmentation depends highly on the quality of synthetic data, so we evaluate the quality of data produced by different data synthesis methods. Imbalanced-learn is a common python package offering a number of resampling techniques commonly used in datasets showing strong between-class imbalance [35]. We set these re-sampling techniques as the baseline methods. It contains three categories: Over-sampling, Under-sampling and Hybrid methods.

- Over-sampling technique tends to generate more samples that are similar to the minority data to increase the proportion of the minority class(es), and it includes five classical methods: Random minority over-sampling (ROS), Synthetic Minority Over-sampling (SMOTE) [12], Borderline SMOTE (bSMOTE) [50], SMOTE for Nominal Continuous (SMOTE-NC) and Adaptive synthetic sampling (ADASYN) [51].
- Under-sampling technique tends to discard partial data of the majority class(es) to decrease the proportion of the majority class(es), and it includes ten classical methods: Random under-sampling (RUS), Repeated Edited Nearest Neighbors (RENN) [52], One-Sided Selection (OSS), Neighborhood Cleaning Rule (NCR), Instance Hardness Threshold (IHT), Condensed Nearest Neighbor (CNN), Edited Nearest Neighbors (ENN) [53], NearMiss, Extraction of majority-minority Tomek links (TL) and AllKNN [52].
- Hybrid-sampling technique tends to combine the over-sampling and under-sampling technique to generate more samples that are similar to the minority class of data and discard partial data of the majority class(es) simultaneously to balance the proportion of both, and it includes two classical methods [54]: SMOTEtomek - SMOTE + Tomek and SMOTEenn - SMOTE + ENN.

The quality of synthetic data is measured by seven metrics, Precision, Recall, F1, FNR, FPR, AUC and Accuracy. Comparison results on $test^{+}$ and $test^{-21}$ sets of NSL-KDD are shown in Table. 10 where the methods' names begin with ''O_'', ''U_'' and ''H_'' indicate the over-sampling methods, under-sampling methods, and hybrid-sampling methods, respectively. For each group of comparison results, the best metric values are marked bold in each column. Generally, the higher true rate values and lower false rate values are, the better quality of generated data is. Analyzing all the metric values globally, most of the data re-sampling methods cannot maintain the high true rates and low false rates simultaneously, since the imbalance problem is not well solved.

Most of the Over-sampling methods aim at duplicating the original minority samples or producing new samples according to the distance [12], which ignores the data distribution so that the generated data will confuse the inter-class margin. In contrast, we produce the data with the help of LSGAN that can learn the distribution of minority samples and then generate the similar samples that obey the same or similar distribution. Therefore, the augmented training dataset constructed by our augmentation strategy is more effective than other over-sampling methods.

Most of the Under-sampling and Hybrid-sampling methods solve the imbalanced training set by randomly discarding partial majority class samples. They reduce the training set, and thus decrease the training time and consume less resources, but they ignore the distribution and might lose the characteristics information that is useful to the majority class [8]. On the contrary, we keep all the original samples of training set and avoid information loss. Furthermore, we insert the similar samples in the raw training set to enhance the characteristics of rare samples. Though the FPR of U_IHT method is less than that of ours, its FNR is the worst, and the over-fitting presents itself. FPR value of ours is 0.126 and 0.098 for two test sets which are both the second best of all methods. Hence, our augmentation strategy can be approximately seen as the best method.

Overall, the metric values of $test^{-21}$ set are less than those of $test^{+}$ set, because the $test^{-21}$ set contains many unknown attacks that do not occur in the $test^{+}$ set, so the difficulty of anomaly detection is increased. In conclusion, NADS-RA outperforms other data re-sampling method.

4) COMPARING DIFFERENT MIXING RATIOS OF AUGMENTED DATASET

After producing the high-quality samples, how to control the proportion of each class in the augmented training set is validated. According to the Table.3, there are five classes in the NSL-KDD dataset, and U2R and R2L account 0.04% and 0.79% in the raw training set, respectively. The main challenge faced by many experiments of the state-of-the-art works [18], [42], [47] is the low detection rates of U2R and R2L. In an ideal balanced set, each class accounts the same proportion that is 20% for each class in NSL-KDD. We control the proportions of U2R and R2L the same, and increase them from the original proportion that is less than 1% to 30%. AUC and ROC curves have been proved effective to evaluate the overall classification effectiveness of the imbalanced dataset [40], so we use the average AUC obtained from 100 groups of experiments.

Figure. 10 shows the ROC curves of NSL-KDD $test^{+}$ and $test^{-21}$ set. Obviously, the AUC is the least in the raw imbalanced training set, and the AUC improvement is achieved on all augmented training sets. A general trend appears that with the increasing proportion, the AUC value tends to be bigger. By comparing the AUC metric for different proportions of U2R and R2L, it can be found that accounting 20% for each class contributes to a more stable and effective classification performance. The classification details of each class is shown in Figs.11 and 12. For U2R and R2L detection, AUC is improved on the augmented training set, and the other classes remain almost unchanged or increased. It can be deduced that the produced rare data not only enhances their characteristics, but also helps to improve the classifier's

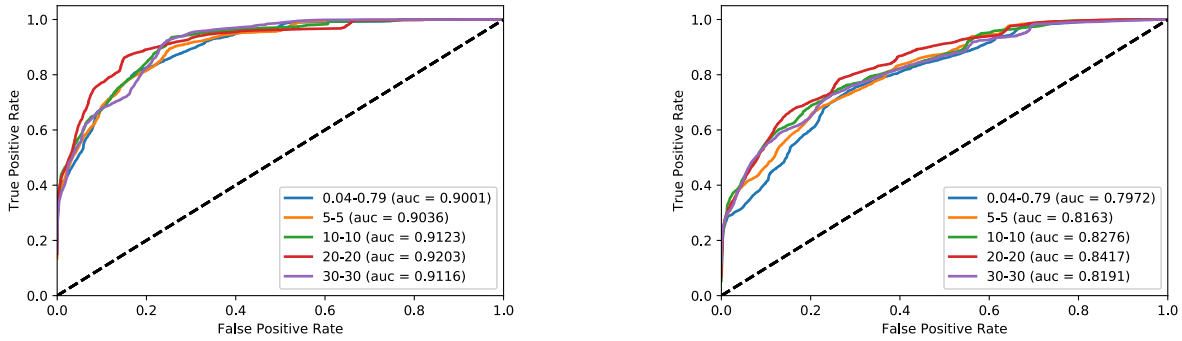
TABLE 10. The Comparison Results of Different Data Re-Sampling Methods of NSL-KDD.

TestSet	Theoretics	Methods	Precision	Recall	F1	FNR	FPR	AUC	Acc
$test^{+}$	Over-sampling	**NADS-RA**	**0.79535**	**0.783**	**0.77247**	**0.217**	0.12639	**0.82831**	**0.783**
		O_ADASYN	0.58898	0.63015	0.58849	0.36985	0.21314	0.7085	0.63015
		O_bSMOTE	0.64644	0.64776	0.56792	0.35224	0.24845	0.69965	0.64776
		O_ROS	0.58566	0.62194	0.57101	0.37806	0.23598	0.69298	0.62194
		O_SMOTE	0.74437	0.68275	0.6621	0.31725	0.16108	0.76084	0.68275
		O_SMOTEnc	0.69313	0.63933	0.58514	0.36067	0.25734	0.69099	0.63933
	Under-sampling	U_ALLKNN	0.57692	0.63662	0.57997	0.36338	0.23053	0.70305	0.63662
		U_CNN	0.48156	0.57971	0.51214	0.42029	0.26643	0.65664	0.57971
		U_ENN	0.62634	0.59772	0.52702	0.40228	0.3002	0.64876	0.59772
		U_IHT	0.34698	0.27524	0.30022	0.72476	**0.08372**	0.59576	0.27524
		U_NCR	0.61533	0.63432	0.56887	0.36568	0.26484	0.68474	0.63432
		U_nearMiss	0.66346	0.6211	0.57422	0.3789	0.22954	0.69578	0.6211
		U_OSS	0.63725	0.6156	0.54735	0.3844	0.26616	0.67472	0.6156
		U_RENN	0.63366	0.63232	0.57409	0.36768	0.26929	0.68151	0.63232
		U_RUS	0.71198	0.6887	0.65131	0.3113	0.17997	0.75436	0.6887
		U_TL	0.67639	0.66386	0.60903	0.33614	0.22879	0.71753	0.66386
	Hybrid-sampling	H_SMOTEenn	0.68205	0.67916	0.63168	0.32084	0.20597	0.7366	0.67916
		H_SMOTEtomek	0.737	0.67242	0.63095	0.32758	0.21363	0.72939	0.67242
$test^{-21}$	Over-sampling	**NADS-RA**	**0.73001**	**0.60945**	**0.62072**	**0.39055**	0.09848	**0.75548**	**0.60945**
		O_ADASYN	0.37544	0.33291	0.30155	0.66709	0.18898	0.57197	0.33291
		O_bSMOTE	0.51735	0.34186	0.27357	0.65814	0.18777	0.57704	0.34186
		O_ROS	0.3703	0.30338	0.27048	0.69662	0.17263	0.56537	0.30338
		O_SMOTE	0.63597	0.42582	0.41185	0.57418	0.14326	0.64128	0.42582
		O_SMOTEnc	0.63448	0.34734	0.31965	0.65266	0.16016	0.59359	0.34734
	Under-sampling	U_ALLKNN	0.35216	0.33367	0.28912	0.66633	0.17447	0.5796	0.33367
		U_CNN	0.27977	0.30793	0.25278	0.69207	0.22134	0.5433	0.30793
		U_ENN	0.52732	0.31924	0.27904	0.68076	0.15701	0.58111	0.31924
		U_IHT	0.19683	0.20852	0.18725	0.79148	**0.06239**	0.57307	0.20852
		U_NCR	0.4529	0.33958	0.29585	0.66042	0.17408	0.58275	0.33958
		U_nearMiss	0.55691	0.30439	0.27242	0.69561	0.14232	0.58103	0.30439
		U_OSS	0.54401	0.29705	0.25956	0.70295	0.17824	0.5594	0.29705
		U_RENN	0.5131	0.34253	0.30938	0.65747	0.15287	0.59483	0.34253
		U_RUS	0.60519	0.43013	0.4011	0.56987	0.13787	0.64613	0.43013
		U_TL	0.55304	0.37519	0.34143	0.62481	0.15585	0.60967	0.37519
	Hybrid-sampling	H_SMOTEenn	0.56703	0.40506	0.38344	0.59494	0.16133	0.62187	0.40506
		H_SMOTEtomek	0.67443	0.39603	0.37924	0.60397	0.14719	0.62442	0.39603

global learning ability. Therefore, the augmented training set is helpful to improve the detection rate of rare anomalies, and the balanced training set seems to be more promising to train a globally effective classifier than the imbalanced training set.

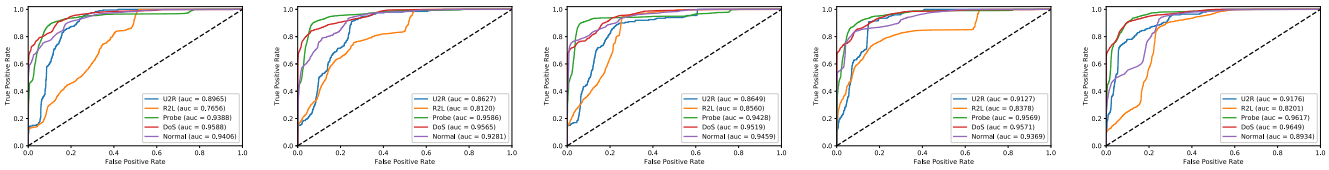
D. GENERAL APPLICABILITY ANALYSIS

To prove the general applicability of NADS-RA, we also implement it on another two scenarios including the credit card fraud detection and software defect detection. Three publicly benchmark datasets are used: one credit dataset [22] and two software defect datasets (JM1 and PC5) from NASA MDP project [23]. 30% of the dataset are selected as test samples and the remaining are used as training samples. To reflect the average classification effect, we take 100 groups of experiments by randomly sampling test samples, then present the average results.



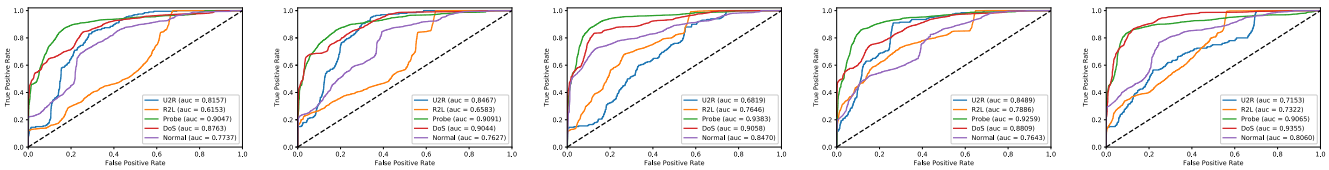
(a) test^{+} set (b) test^{-21} set

FIGURE 10. ROC curves of different proportions (U2R-R2L) on NSL-KDD.



(a) 0.04-0.79 (a) 5-5 (b) 10-10 (c) 20-20 (d) 30-30

FIGURE 11. ROC curves of different proportions (U2R-R2L) on NSL-KDD test^{+} set.



(a) 0.04-0.79 (a) 5-5 (b) 10-10 (c) 20-20 (d) 30-30

FIGURE 12. ROC curves of different proportions (U2R-R2L) on NSL-KDD test^{-21} set.

1) CASE STUDY 1: CREDIT CARD FRAUD DETECTION

In this case, to identify the fraud transactions from the legitimate ones, a binary-classification task is abstracted. Since Recall, F1, Gmean and FPR are used in the state-of-the-art methods, these four metrics are used to evaluate the effectiveness of NADS-RA. An ideal fraud detection system should identify precisely the fraudulent transactions, prevent financial loss, and at the same time reduce the number of false positive transactions that require control of human source with significant costs. Table. 11 lists the comparison results obtained before and after augmentation of different detection methods (They are marked as "Method" and "Method+", respectively), and they express the superior performance of augmentation. Compared with the over-sampling method used in research [55], we additionally represent the original feature vectors into the images, which contributes to the better learning of hidden spatial knowledge. Therefore, NADS-RA provides promising support in the credit card fraud detection.

TABLE 11. Comparison With State-of-the-Art Methods on Credit Card Fraud Detection.

Methods	Precision	Recall	F1	FPR
SVM	0.794	0.794	0.891	0.0002
SVM+	0.847	0.659	0.92	0.0008
RF	0.756	0.805	0.869	0.0001
RF+	0.847	0.776	0.92	0.0003
DT	0.794	0.782	0.891	0.0002
DT+	0.855	0.663	0.925	0.0008
[55]	**0.93204**	0.73282	0.82051	-
Ours	0.858268	**0.832061**	**0.844961**	**0.0001**

2) CASE STUDY 2: SOFTWARE DEFECT DETECTION

Since Recall, F1, Gmean and FPR are used in the state-of-the-art methods, we present these metric values obtained from JM1 dataset and PC5 dataset in Table. 12. The detection results obtained before and after augmentation are marked as "Method" and "Method+", respectively. As we all know, a high Recall can maintain an accurate detection of defects, and a low FPR involves less human investigators. Combining these metric values together, our method takes on an advantage over others. Though the research [56]

TABLE 12. Comparison With State-of-the-Art Methods on Software Defect Detection.

Test sets	Methods	Recall	F1	Gmean	FPR
JM1	SVM	0.007	0.013	0.072	**0**
	SVM+	1	0.367	0	1
	RF	0.202	0.288	0.402	0.057
	RF+	0.24	0.32	0.44	0.075
	DT	0.002	0.038	0.125	0.009
	DT+	0.225	0.291	0.425	0.094
	[56]	0.587	0.411	0.616	0.301
	Ours	**0.83**	**0.613**	**0.882**	**0.254**
PC5	SVM	0.097	0.172	0.271	**0.012**
	SVM+	0.93	0.46	0.92	0.757
	RF	0.382	0.468	0.554	0.089
	RF+	0.549	0.539	0.678	0.174
	DT	0.201	0.301	0.394	0.05
	DT+	0.653	0.551	0.748	0.256
	[56]	**0.95**	**0.846**	0.458	0.764
	Ours	0.93	0.78	**0.951**	**0.163**

obtains the high Recall and F1, its FPR is the too high to require many human resources. According to the statistics of research [57], the accuracy value is less than 50% when there is no repeated data in JM1. In contrast, under the same dataset without repeated modules, the accuracy value of NADS-RA is about 67% on JM1 dataset. In all, the comparison results suggest that NADS-RA has a general adaptability and applicability in different scenarios. Hence, NADS-RA has a great potentiality to be applied in other security fields.

E. SIGNIFICANCE TEST ANALYSIS

To strengthen our approach, the statistical significance tests are conducted to compare the performances of various approaches on multiple datasets. Friedman test and post-hoc Nemenyi test are used to further analyze whether our approach is statistically significant compared with others. As shown in Table.13, the AUC values of SVM, RF, DT and ours over the NSL-KDD, Credit card, JM1 and PC5 datasets are demonstrated. After Friedman hypothesis testing, the null hypothesis (the performances of all approaches are equivalent) is rejected at $\alpha = 0.05$ since the p-value is 0.0194. This result indicates that our approach is significantly different with other approaches.

Afterwards, it needs to conduct the post-hoc test to further measure how significant are the performance differences among the considered approaches. The post-hoc Nemenyi test is adopted. The critical difference (CD) of 2.3452 is computed at $p\text{-}value = 0.05$. For the AUC metric, the Friedman average ranks of SVM, RF, DT and ours are 3.75, 2.25, 3 and 1, respectively. Generally, the lower the rank, the better performance of the approach is. In Table.13, the best value is indicated in bold. Ours appears as the best of the benchmark approaches, so it is picked as a control algorithm for being compared with the remaining approaches. The rank differences among SVM-ours, RF-ours and DT-ours, the first one is bigger than the CD value and the latter two ones are lower than the CD value, so it can be accepted at the confidence degree of 0.95 that SVM is statistically different from ours, and RF and DT have no statically significant difference in terms of AUC, despite our method wins on most of the datasets. This proves that the deep learning classifiers or ensembling classifiers are more powerful in the big data network anomaly detection.

F. DISCUSSION

Compared with the state-of-the-art works, we obtain a better result in the imbalanced network anomaly detection. The AUC of our presentation strategy is improved by an average of 10 percent compared with 12 detection methods. Since we absorb the advantages of the feature representation and data augmentation together, and then propose an image-based data augmentation strategy for network data. The existing feature representation methods either disrupt the feature unity or lose the spatial knowledge of partial adjacent features, so that the classifier trained on the obtained images do not perform well due to the information loss. On the other hand, our AUC is improved by at least 10 percent compared with 17 data generation methods. Since the conventional data generation methods produce data according to the distance or density, which will disrupt the distribution of original data, and even confuse the margin between the classes. In contrast, we utilize the Re-circulation Pixel Permutation (RPP) strategy which retains the feature unity through bundling the discrete features and keeping the original continuous features. It not only maintains the spatial structure of raw features, but also enhances the spatial knowledge of adjacent features. Furthermore, with the help of LSGAN's ability to learn the data distribution, we produce the augmented image data to enrich the rare class, and then improve the detection rate of rare classes and avoid over-fitting. Therefore, our superior performance can be explained as that we not only maintain the spatial features within each sample, but also keep the distribution characteristics of rare class, and then the augmented training set is used to train an effective classifier.

Meanwhile, we cannot ignore the limitation in this article. The larger represented image size and enlarged training set might cost more training time and resources. Though we have explored various training sets with different proportions of rare classes, it is just for finding the optimal mixing ratios, so a refined training set and fast training process is still essential to be incorporated with the incremental learning online in the big data environment. We will take this problem as our future work.

TABLE 13. Performance Results of all Classifiers w.r.t AUC Metric Along With Friedman Rank.

	NSL-KDD	Credit	JM1	PC5	Average	Friedman Rank	Friedman p-value
SVM	0.759	0.829	0.5	0.587	0.669	3.75	0.0194
RF	0.801	0.888	0.583	0.688	0.74	2.25	
DT	0.747	0.831	0.566	0.699	0.71	3	
Ours	**0.834**	**0.916**	**0.788**	**0.884**	**0.855**	**1**	

VI. CONCLUSION

A. SUMMARY

In summary, we study how to represent the network traffic features as images and balance the imbalanced training dataset to improve the classification accuracy of rare anomalies. The proposed NADS-RA produces augmented data based on feature images, which maintains the spatial knowledge between features and also keeps the data distribution of each class. Through the experiments conducted on five public benchmark datasets including NSL-KDD and UNSW-NB15, and so on, NADS-RA is in good agreement with experimental observations, and the advantages of feature representation and data augmentation are explained. They contribute to learning the high-level characteristics and the hidden knowledge of data, making the classifier more powerful. Overall, NADS-RA opens opportunities for improving the imbalanced classification in the non-image-processing area, and also provides a general deep-learning-based detection scheme for the imbalanced classification in different scenarios.

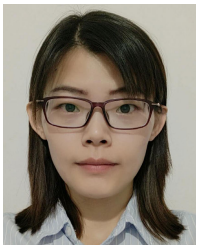
B. FUTURE WORK

Our current work focuses on over-sampling each class of rare data to balance the imbalanced training dataset. In the future, we will study a more intelligent data generation method to maintain the intra-class distribution and inter-class margin and to further produce multiple classes of data simultaneously, as well as a fast training model to deal with the problem of more training time and more computing resources brought by the enlarged training set.

REFERENCES

[1] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *J. Netw. Comput. Appl.*, vol. 60, pp. 19–31, Jan. 2016.
[2] X. Luo, X. Di, X. Liu, H. Qi, J. Li, L. Cong, and H. Yang, *Anomaly Detection for Application Layer User Browsing Behavior Based on Attributes and Features*. vol. 1069, no. 1. Amsterdam, The Netherlands: Elsevier, 2018, pp. 1–9.
[3] F. J. Castellanos, J. J. Valero-Mas, J. Calvo-Zaragoza, and J. R. Rico-Juan, "Oversampling imbalanced data in the string space," *Pattern Recognit. Lett.*, vol. 103, pp. 32–38, Feb. 2018.
[4] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. for Secur. Defense Appl.*, Jul. 2009, pp. 1–6. [Online]. Available: https://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html
[5] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic," *IEEE Sensors Lett.*, vol. 3, no. 1, pp. 1–4, Jan. 2019.
[6] C. Thomas, "Improving intrusion detection for imbalanced network traffic," *Secur. Commun. Netw.*, vol. 6, no. 3, pp. 309–324, Mar. 2013.
[7] S. Phetlasy, S. Ohzahata, C. Wu, and T. Kato, "A sequential classifiers combination method to reduce false negative for intrusion detection system," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 5, pp. 888–897, 2019.
[8] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.
[9] H. Liu, B. Lang, M. Liu, and H. Yan, "CNN and RNN based payload classification methods for attack detection," *Knowl.-Based Syst.*, vol. 163, pp. 332–341, Jan. 2019.
[10] K. Wu, Z. Chen, and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018.
[11] A. Ali-Gombe and E. Elyan, "MFC-GAN: Class-imbalanced dataset classification using multiple fake class generative adversarial network," *Neurocomputing*, vol. 361, pp. 212–221, Oct. 2019.
[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[13] T. Zhu, Y. Lin, Y. Liu, W. Zhang, and J. Zhang, "Minority oversampling for imbalanced ordinal regression," *Knowl.-Based Syst.*, vol. 166, pp. 140–155, Feb. 2019.
[14] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, 2017, pp. 2813–2821, doi: 10.1109/ICCV.2017.304.
[15] D. Sun, K. Yang, B. Lv, and Z. Shi, "Could we beat a new mimicking attack?" in *Proc. 19th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2017, pp. 247–250.
[16] Z. Li, Z. Qin, K. Huang, X. Yang, and S. Ye, "Intrusion detection using convolutional neural networks for representation learning," in *Neural Information Processing*. Cham, Switzerland: Springer, 2017, pp. 858–866.
[17] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, "An empirical study on network anomaly detection using convolutional neural networks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1595–1598.
[18] S. Potluri, S. Ahmed, and C. Diedrich, "Convolutional neural networks for multi-class intrusion detection system," in *Proc. Int. Conf. Mining Intell. Knowl. Explor.* Cham, Switzerland: Springer, 2018, pp. 225–238.
[19] D. Gümüşbaş, T. Yldrm, A. Genovese, and F. Scotti, "A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems," *IEEE Syst. J.*, early access, May 26, 2020, doi: 10.1109/JSYST.2020.2992966.
[20] X. Liu, X. Di, W. Liu, X. Zhang, H. Qi, J. Li, J. Zhao, and H. Yang, "NADSR: A network anomaly detection scheme based on representation," in *Knowledge Science, Engineering and Management*. Cham, Switzerland: Springer, 2020, pp. 380–387.
[21] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.
[22] (2013). *Credit Card Fraud Dataset*. [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud
[23] NASA. (2004). *Metric Data Program MDP*. [Online]. Available: http://mdp. ivv.nasa.gov
[24] N. Macek and M. Milosavljevic, "Reducing U2R and R2L category false negative rates with support vector machines," *Serbian J. Electr. Eng.*, vol. 11, no. 1, pp. 175–188, 2014.
[25] S. Z. Lin, Y. Shi, and Z. Xue, "Character-level intrusion detection based on convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–8.

[26] Y. Kim, J. Sa, S. Kim, and S. Lee, "Shapelets-based intrusion detection for protection traffic flooding attacks," in *Database Systems for Advanced Applications*. Cham, Switzerland: Springer, 2018, pp. 227–238.
[27] W. W. Nsunza, A.-Q.-R. Tetteh, and X. Hei, "Accelerating a secure programmable edge network system for smart classroom," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 1384–1389.
[28] R. Blanco, P. Malagon, J. J. Cilla, and J. M. Moya, "Multiclass network attack classifier using CNN tuned with genetic algorithms," in *Proc. 28th Int. Symp. Power Timing Modeling, Optim. Simulation (PATMOS)*, Jul. 2018, pp. 177–182.
[29] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1222–1228.
[30] V. L. Cao, M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3074–3087, Aug. 2019.
[31] S.-Y. Ji, B.-K. Jeong, and D. H. Jeong, "Evaluating visualization approaches to detect abnormal activities in network traffic data," *Int. J. Inf. Secur.*, 2020, doi: 10.1007/s10207-020-00504-9.
[32] D. Jackle, F. Fischer, T. Schreck, and D. A. Keim, "Temporal MDS plots for analysis of multivariate data," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 141–150, Jan. 2016.
[33] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 445–458, Jun. 2019.
[34] E. Min, J. Long, Q. Liu, J. Cui, and W. Chen, "TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest," *Secur. Commun. Netw.*, vol. 2018, Jul. 2018, Art. no. 4943509.
[35] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5, 2017.
[36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[37] D. Zhang, Q. Niu, and X. Qiu, "Detecting anomalies in communication packet streams based on generative adversarial networks," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Cham, Switzerland: Springer, 2019, pp. 470–481.
[38] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," *Comput. Secur.*, vol. 75, pp. 36–58, Jun. 2018.
[39] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manage.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.
[40] J. Hu, H. Yang, M. R. Lyu, I. King, and A. Man-Cho So, "Online nonlinear AUC maximization for imbalanced data sets," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 882–895, Apr. 2018.
[41] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[42] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *Networked Digital Technologies*. Berlin, Germany: Springer, 2012, pp. 135–145.
[43] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.
[44] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484–497, Feb. 2017.
[45] A.-H. Muna, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial Internet of Things based on deep learning models," *J. Inf. Secur. Appl.*, vol. 41, pp. 1–11, Aug. 2018.
[46] N. M. Khan, A. Negi, and I. S. Thaseen, "Analysis on improving the performance of machine learning models using feature selection technique," in *Proc. Int. Conf. Intell. Syst. Design Appl.* Cham, Switzerland: Springer, 2018, pp. 69–77.
[47] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.
[48] N. Moustafa, G. Creech, and J. Slay, "Anomaly detection system using beta mixture models and outlier detection," in *Progress in Computing, Analytics and Networking*, P. K. Pattnaik, S. S. Rautaray, H. Das, and J. Nayak, Eds. Singapore: Springer, 2018, pp. 125–135.
[49] N. Moustafa, G. Creech, and J. Slay, "Big data analytics for intrusion detection system: Statistical decision-making using finite Dirichlet mixture models," in *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*. Cham, Switzerland: Springer, 2017, pp. 127–156.
[50] H. Han, W. Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Advances in Intelligent Computing*. Berlin, Germany: Springer, 2005, pp. 878–887.
[51] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 1322–1328.
[52] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 6, pp. 448–452, Jun. 1976.
[53] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-2, no. 3, pp. 408–421, Jul. 1972.
[54] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004.
[55] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Inf. Sci.*, vol. 479, pp. 448–455, Apr. 2019.
[56] T. Choeikiwong and P. Vateekul, "Software defect prediction in imbalanced data sets using unbiased support vector machine," in *Information Science and Applications* (Lecture Notes in Electrical Engineering), vol. 339. Berlin, Germany: Springer, doi: 10.1007/978-3-662-46578-3_110.
[57] D. Gray, D. Bowes, N. Davey, Y. Sun, and B. Christianson, "Reflections on the NASA MDP data sets," *IET Softw.*, vol. 6, no. 6, pp. 549–558, Dec. 2012.



XU LIU received the B.E. degree in network engineering from Qingdao University, in 2013, and the M.E. degree in computer software theory from the Changchun University of Science and Technology, China, in 2017, where she is currently the Ph.D. degree in computer science and technology. Her research interests include network security, information security, anomaly detection, artificial intelligence, big data, game theory, and cloud computing.

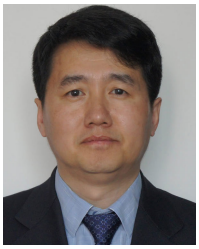
XIAOQIANG DI received the B.S. degree in computer science and technology from the Changchun University of Science and Technology, in 2002, and the M.S. and Ph.D. degrees in communication and information systems from the Changchun University of Science and Technology, in 2007 and 2014, respectively. From August 2012 to August 2013, he was a Visiting Scholar with the Norwegian University of Science and Technology, Norway. He is currently a Professor and a Supervisor of Ph.D. with the Changchun University of Science and Technology. His current research interests include network information security and integrated networks.

QIANG DING received the B.E. degree in computer and practical application, and the M.S. degree in computer technology from the Changchun University of Science and Technology, China, in 1997 and 2007, respectively. He is currently a mid-level Engineer in information center with the Changchun University of Science and Technology. His research interests include network management, information security, and big data.

WEIYOU LIU received the B.E. degree in communication engineering from the Changchun University of Science and Technology, China, in 2018, where he is currently pursuing the master's degree in computer science and technology. His research interests include network security, big data, information security, anomaly detection, and artificial intelligence.

HUI QI received the Ph.D. degree from the College of Computer Science and Technology, Jilin University, in 2015. He is currently an Associate Professor and a Master Student Supervisor with the Changchun University of Science and Technology. His research interests include network security, access control, and vehicular networks.

JINQING LI received the B.S. degree from the Changchun University of Technology, in 2002, and the M.S. and Ph.D. degrees from the Changchun University of Science and Technology, in 2007 and 2014, respectively. She is currently an Associate Professor and a Master Student Supervisor with the Changchun University of Science and Technology. Her research interests include network security, information security, and image encryption.



HUAMIN YANG received the B.E. degree from the Dalian University of Technology, in 1985, China, the M.E. degree from Jilin University, and the Ph.D. degree from the Changchun University of Science and Technology, in 2002. He is currently a Professor and a Supervisor of Ph.D. with the Changchun University of Science and Technology. His current research interests include information security, computer simulation modeling, and virtual reality technology.