Combining Oversampling with Recurrent Neural Networks for Intrusion Detection

Jenq-Haur Wang[1](✉) and Tri Wanda Septian[2]

[1] National Taipei University of Technology, Taipei, Taiwan
jhwang@csie.ntut.edu.tw
[2] Sriwijaya University, Palembang, Indonesia

Abstract. Previous studies on intrusion detection focus on analyzing features from existing datasets. With various types of fast-changing attacks, we need to adapt to new features for effective protection. Since the real network traffic is very imbalanced, it's essential to train appropriate classifiers that can deal with rare cases. In this paper, we propose to combine oversampling techniques with deep learning methods for intrusion detection in imbalanced network traffic. First, after preprocessing with data cleaning and normalization, we use feature importance weights generated from ensemble decision trees to select important features. Then, the Synthetic Minority Oversampling Technique (SMOTE) is used for creating synthetic samples from minority class. Finally, we use Recurrent Neural Networks (RNNs) including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) for classification. In our experimental results, oversampling improves the performance of intrusion detection for both machine learning and deep learning methods. The best performance can be obtained for CIC-IDS2017 dataset using LSTM classifier with an F1-score of 98.9%, and for CSE-CIC-IDS2018 dataset using GRU with an F1-score of 98.8%. This shows the potential of our proposed approach in detecting new types of intrusion from imbalanced real network traffic.

Keywords: Class imbalance · Oversampling · Feature selection · Long short-term memory · Gated recurrent unit

1 Introduction

Nowadays, new variants of security threats in the cyber world are massively increasing on the Internet. It is the main focus for the system administrator to protect the network infrastructure from malicious behaviors such as new intrusions and attacks. Therefore, intrusion detection has become an important research area in network security. Intrusion detection systems aim to actively detect attacks and identify the critical illegal behaviors from network traffic. There are some challenges in effective classification for intrusion detection. First, most existing research on analyzing the characteristic of attack patterns use popular datasets such as KDD CUP'99, NSL-KDD, and ISCX2012, which need some improvement since they are out-of-date. With the development of Internet technology, there are increasing amount of new cyber-attacks. To deal with the issues of unreliable datasets that are out of date, we utilize new intrusion detection datasets

C. S. Jensen et al. (Eds.): DASFAA 2021 Workshops, LNCS 12680, pp. 305–320, 2021.
https://doi.org/10.1007/978-3-030-73216-5_21

including CIC-IDS2017 dataset [1], and CSE-CIC-IDS2018 dataset [2] which are developed by Sharafaldin et al. [3]. These new public datasets are based on real-time network traffic captured around the world, which are helpful for intrusion detection research. Second, due to the changing characteristics of new attacks in real data, we need to select features that can capture the most important characteristics. Louppe et al. [4] introduced the variable importance derived from tree-based methods and can be implemented as feature selection methods to improve classification accuracy. Third, we are faced with large-scale imbalanced datasets since only a small percentage of real network traffic are attacks or illegal traffic. To address this issue, we utilize Synthetic Minority Oversampling Technique (SMOTE) [5] to improve the prediction accuracy for the imbalanced dataset. Finally, for the classification algorithm, we compare classical learning methods, such as Random Forest [6], Decision Tree [7], and Naïve Bayes [8], with deep learning methods such as Long Short-Term Memory (LSTM) [9], and Gated Recurrent Unit (GRU) [10] in their classification performances. The contributions of this paper include:

1. We evaluate classification performance of recurrent neural networks (RNNs) for intrusion detection on two new publicly available datasets, which are captured from real work traffic in large scale.
2. We improve the performance of intrusion detection for the imbalanced dataset by using SMOTE oversampling technique as the feature selection method for both classical machine learning and deep learning methods.

The remainder of this paper are as follows. First, related work is reviewed in Sect. 2. Then, the proposed method is described in Sect. 3, and our experimental results are analyzed and discussed in Sect. 4. Finally, we give conclusions in Sect. 5.

2 Related Work

Intrusion detection has been an important research topic in information security. Many conventional machine learning methods have been used for intrusion detection. Albayati et al. [11] discussed the intelligent classifier suitable for automatic detection, such as Naïve Bayes, Random Forest, and decision tree algorithm. The best performance can be obtained for Random Forest classifiers with an accuracy of 99.89% when using all of the features from the NSL-KDD dataset. Almseidin et al. [12] evaluated the intrusion detection using machine learning methods: SVM, Random Forest, and decision tree algorithm. Random forest classifier registered the highest accuracy of 93.77%, with the smallest false positive rate for the KDD CUP'99 dataset. Khuphiran et al. [13] researched on detecting Distributed Denial of Services (DDoS), as the most common attack, using DARPA 2009 DDoS datasets, and implementing a traditional SVM and Deep Feed Forward (DFF) algorithm. Deep Feed Forward got the highest accuracy of 99.63% and F1-score is 0.996 while SVM got an accuracy rate of 81.23% and F1-score is 0.826.

Recently, deep learning methods especially recurrent neural networks, such as Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) have been implemented in intrusion detection research area. Althubiti et al. [14] applied LSTM algorithm for multi-classification using the *rmsprop* parameter in the CIDDS-001 dataset, which specializes in web attacks. The best accuracy of 84.83% can be obtained for LSTM, which is better than SVM and Naïve Bayes. Xu et al. [15] proposed a study in IDS with GRU, which uses softmax function for multiclass classification. The best accuracy of 99.42% using KDD CUP’99 and 99.31% using NSL-KDD can be obtained.

For an IDS to accurately detect unauthorized activities and malicious attacks in network traffic, different features might have different importance to distinguish between attacks and normal traffic. On the one hand, feature selection is needed since it is useful in analyzing complex data, and for removing features excessive or irrelevant. On the other hand, network traffic is extremely imbalanced since normal traffic accounts for most of the traffic, while intrusion or attacking traffic are very rare. For feature selection, Alazzam et al. [16] proposed a pigeon inspired optimizer for feature selection, and achieved good accuracy of 0.883 and 0.917 for NSL-KDD and UNSW-NB15 datasets respectively when reducing the feature size to 5. This shows the importance of feature selection in classification.

Regarding the class imbalance problem, some techniques have been proposed. Wu et al. [17] dealt with imbalanced health-related data with deep learning approaches using RNNs. Shuai et al. [18] devised a multi-source learning approach to extract common latent factors from different sources of imbalanced social media for mental disorders detection. To mitigate the problem of overfitting for the imbalanced class with random oversampling, the technique of SMOTE generates synthetic examples by k-nearest neighbor algorithm rather than simply replicating existing instances. Smiti and Soui [19] explored the idea of employing SMOTE and deep learning to predict bankruptcy. Seo and Kim [20] proposed to handle the class imbalance problem of KDD CUP’99 dataset by finding the best SMOTE ratios in different rare classes for intrusion detection.

Due to the growing types of new attacks, we focus on intrusion detection for the new datasets CIC-IDS2017 and CSE-CICIDS2018. Kurniabudi et al. [21] analyzed the features of CIC-IDS2017 dataset with information gain, and achieved the best accuracy of 99.86% for Random Forest. But they only used 20% of the full dataset, and cannot detect some types of traffic, for example, Infiltration attack. Kim et al. [22] compared the performance of intrusion detection on CSE-CICIDS2018 dataset using Convolutional Neural Networks (CNNs) and RNNs. They only focused on DoS category, and achieved the best accuracy of 91.5% and 65% for CNN and RNN, respectively.

In this paper, we apply deep learning methods to classify imbalanced network traffic for intrusion detection, and compare the performance with conventional machine learning methods using the two new datasets. Specifically, we compare variants of RNNs including LSTM and GRU. Then, we apply SMOTE technique to deal with class imbalance problem. To further improve the classification accuracy, we propose to use variable importance derived from tree-based methods [4] for feature selection, because it has fast calculation and suitable for large data size. We used the full datasets, and the best F1 score of 98.9% and 98.8% can be achieved for CIC-IDS2017 and CSE-CICIDS2018 datasets, respectively.

3 The Proposed Method

In our proposed method for intrusion detection using deep learning approach, there are three stages: data preprocessing, feature selection and oversampling technique, and classification. The proposed framework for intrusion detection is shown in Fig. 1.

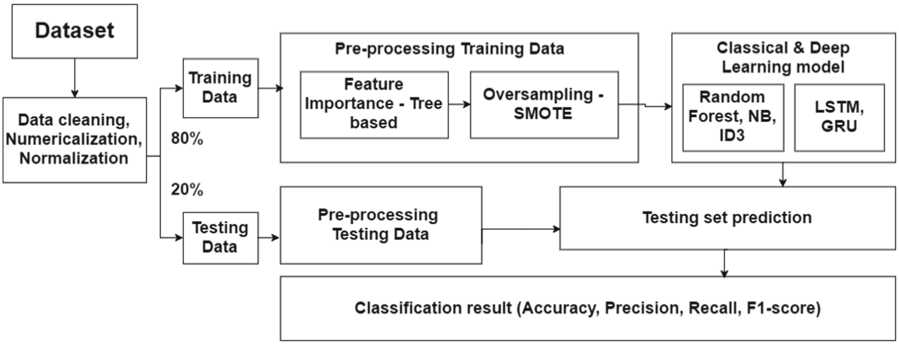
Dataset
Data cleaning, Numericalization, Normalization
80%
20%
Training Data
Testing Data
Pre-processing Training Data
Feature Importance - Tree based
Oversampling - SMOTE
Classical & Deep Learning model
Random Forest, NB, ID3
LSTM, GRU
Pre-processing Testing Data
Testing set prediction
Classification result (Accuracy, Precision, Recall, F1-score)

Fig. 1. The proposed framework for intrusion detection.

As shown in Fig. 1, in the preprocessing stage, missing and undefined values are fixed in data cleaning stage, and separate datasets are combined into a single one. In feature selection stage, we use totally randomized trees to find the important features in the dataset. Then, the SMOTE oversampling technique is used to deal with the class imbalance problem. Finally, we compare the classification performance of classical machine learning methods including Random Forest (RF), Iterative Dichotomiser 3 (ID3), and Naïve Bayes (NB) with recurrent neural networks, including LSTM and GRU on the large scale network traffic data. In the following subsections, we describe each stage in more details.

3.1 Preprocessing

There are several preprocessing tasks needed for the new datasets. First, we remove unnecessary information from the original dataset including the socket information of each data instance, such as source IP address “src_ip”, destination IP address “dst_ip”, Flow ID “flow_id,” and “protocol.” The reason to remove these is to provide unbiased detection. Second, we remove unreadable data which might include some noise in class labels such as: ‘Web Attack Â\x96 Brute Force’, ‘Web Attack Â\x96 XSS’, ‘Web Attack Â\x96 SQL Injection’, which can be replaced to distinct Unicode characters. Then, we also remove invalid numbers, such as Not a Number (NaN) and ‘Infinity’. The missing values and other errors in the dataset are fixed, such as in “flow_bytes_per_s” and “flow_pkts_per_s” features. Regarding the data types, the dataset consists of categorical, strings, and numeric data types such as float64 and int64. The categorical data type in the label consists of benign, and all attack types. In the CSE-CICIDS2018 dataset, the data types of some features are not appropriate, which were changed from int64 to the float64 data type. Finally, for training purpose, the numeric attributes need to be normalized, since the difference of scale in numbers or values can degrade the performance of classification. For example, some of the features with large numeric values, e.g., ‘flow_duration’ can dominate small numeric values such as ‘total_fwd_packets’ and ‘total_fwd_pckts’. Thus, we use min-max normalization to convert values into a normalized range.

3.2 Feature Selection

To select the most important features, we adopt the feature selection method proposed by Louppe et al. [4] to estimate feature importance using Mean Decrease Impurity (MDI) from randomized ensemble trees. Let V = {X_1, X_2, ….X_p} denote categorical input variables, and Y means a categorical output, Shannon entropy is used as impurity measure on totally randomized trees as follows:

$$VarImp(X_m) = \sum_{k=0}^{p-1} \frac{1}{C_p^k} \frac{1}{p-k} \sum_{B \in P_k(V^{-m})} I(X_m; Y|B) \tag{1}$$

$$\sum_{m=1}^{p} VarImp(X_m) = I\left(X_1, X_2, \ldots .X_p; Y\right) \tag{2}$$

Where V^{-m} denotes the subset V\{X_m}, P_k (V^{-m}) denotes subsets of V^{-m} of cardinality k, and I(X_m;Y|B) is the conditional mutual information of X_m and Y given the variables in B.

In this paper, X defines the input features in training data, and Y defines the output class of Benign and Attack. We adopt MDI for feature selection since it calculates each feature importance as the sum over the number of splits (across all trees) that include the features, proportionally to the number of samples it splits. In addition, ensembles of randomized trees are used to select the best subset of features for classification. This reduced feature set is then employed to implement an intrusion detection system.

3.3 Oversampling

In intrusion detection datasets, there is class imbalance problem, where the minority class of attack has much fewer instances than the benign class. The distribution of all classes in CIC-IDS2017 and CSE-CICIDS2018 is shown in Fig. 2.

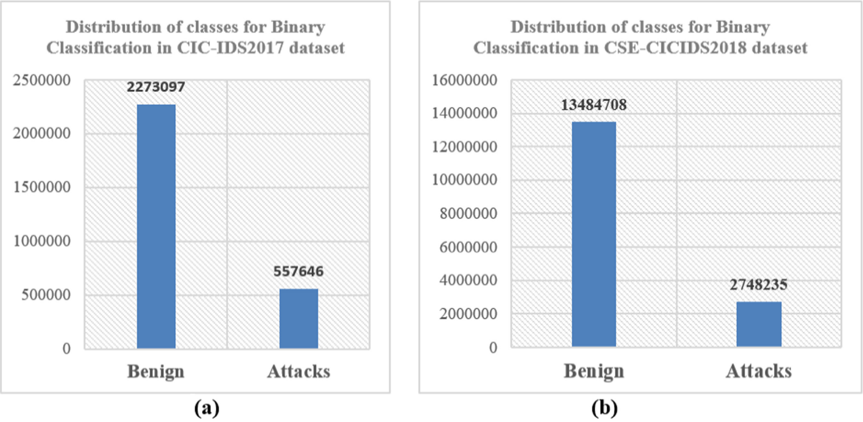

Fig. 2. Distribution of all classes in (a) CIC-IDS2017 dataset (b) CSE-CIC-IDS2018 dataset.

As shown in Fig. 2, the distribution of benign and attack classes show the class imbalance problem in both datasets. To tackle the problem, we adopt SMOTE [5] to improve our prediction of the minority class. The idea is to take each minority class sample, and add synthetic examples on the line segments which join the k-minority class nearest neighbors. This can be done as the following steps:

Step 1: Assume x ∈ A, where A is the set of the minority class. For each x of the k-nearest neighbors, it is obtained from the Euclidean distance calculation between x and samples from the set A.
Step 2: The number of samples N is chosen according to the sample proportion of imbalanced data. For instance, given $x_1, x_2 \ldots, x_N$ $(N \leq k)$ that are randomly selected by k-nearest neighbors, we can build a new set A_1.
Step 3: For each instance $x_k \in A_1$, where k is 1, 2,..., N, the formula is used to create a new instance x_{new} as follows:

$$x_{new} = x + random(0, 1) * \|x - x_k\| \tag{3}$$

The amount of oversampling is influenced by the number of randomly selected samples from the k-nearest neighbors. It has been shown to perform better than simple under-sampling technique because this algorithm creates new instances of the minority class by using convex combinations of neighboring instances.

3.4 Classification

After preprocessing and oversampling the dataset, we use two types of RNNs, including LSTM and GRU, and compare with conventional machine learning classifiers such as Random Forest, ID3, and Naïve Bayes, for intrusion detection.

LSTM is a variation of RNNs to deal with the vanishing gradient problem in sequential data. The architecture of LSTM consists of input gate It, forget gate Ft, output gate Ot, and memory cell Ct, as shown in Fig. 3.

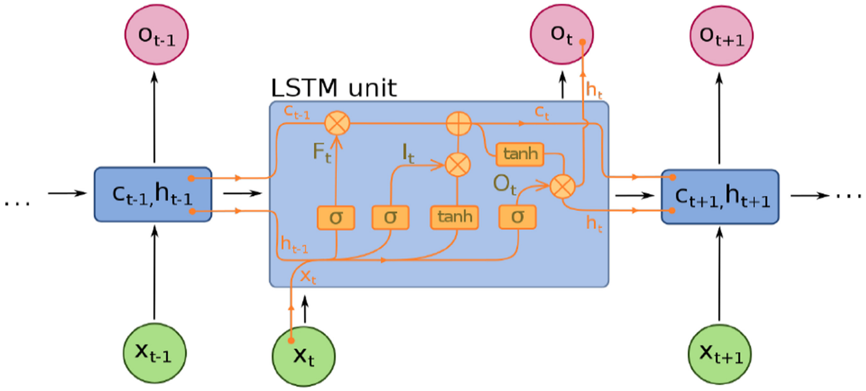

Fig. 3. Long Short Term Memory (LSTM) architecture.

The equations for the operations of LSTM architecture are given below:

$$F_t = \sigma(W_F x_t + U_F h_{h-1} + b_F) \tag{4}$$

$$I_t = \sigma(W_I x_t + U_I h_{h-1} + b_I) \tag{5}$$

$$O_t = \sigma(W_O x_t + U_O h_{h-1} + b_o) \tag{6}$$

$$C_t = F_t \odot c_{t-1} + I_t \odot tanh(W_c x_c + U_c h_{t-1} + b_c) \tag{7}$$

$$h_t = O_t \odot tanh(C_t) \tag{8}$$

$$O_t = f(W_o h_t + b_o) \tag{9}$$

where σ denotes a sigmoid function, x_t means an input vector at time t, h_t denotes a hidden state vector at time t, W denotes the hidden weight matrix from an input, U means the hidden weight matrix from hidden layers, and b means a bias term.

GRU is an LSTM without an output gate, in which the contents are fully written from its memory cell to the output at each time-step. Its internal structure is simpler and therefore considered faster to train as there are fewer computations needed to make updates to its hidden state. GRU has two types of gates: reset gate r, and update gate z. The reset gate determines the new input with the previous memory cell, and the update gate defines how much of the previous memory cell to keep.

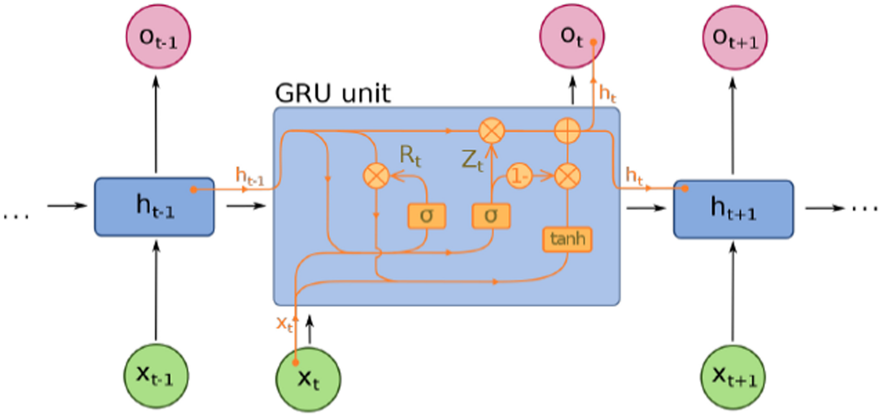

Fig. 4. Gated Recurrent Unit (GRU) architecture

Equations the operations of GRU architecture are given below:

$$Z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{10}$$

$$R_t = \sigma(W_R x_t + U_R h_{t-1} + b_R) \tag{11}$$

$$h_t = (1 - Z_t) \odot h_{t-1} + Z_t \odot tanh(W_h x_t + U_h(R_t \odot h_{t-1}) + b_h) \tag{12}$$

where $\mathrm{Z_t}$ is the update gate, $\mathrm{R_t}$ is the reset gate, and $\mathrm{h_t}$ is the hidden state. $\odot$ is a multiplication element-wise, and σ is the sigmoid activation function. W and U are denoted as learned weight matrices.

4 Experiments

In this paper, we use two new datasets CIC-IDS2017 and CSE-CIC-IDS2018, because they are up-to-date and offer broader attack types and protocols. We want to implement the intrusion detection system using real network traffic data with machine learning and deep learning methods. After the preprocessing stage, we obtained a total of 2,830,743 data instances containing 2,273,097 "benign" and 557,646 "attacks" in CIC-IDS2017. In CSE-CIC-IDS2018 there's a total of 16,232,943 data instances containing 13,484,708 "benign" and 2,748,235 "attacks". The detailed statistics of data distribution in different classes for the two datasets are shown in Figs. 5 and 6.

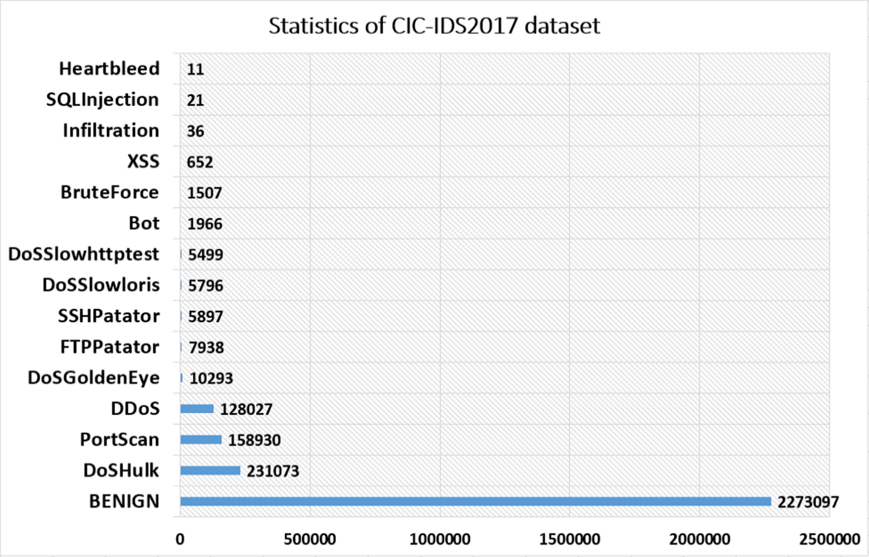

Fig. 5. The statistics of CIC-IDS2017 dataset.

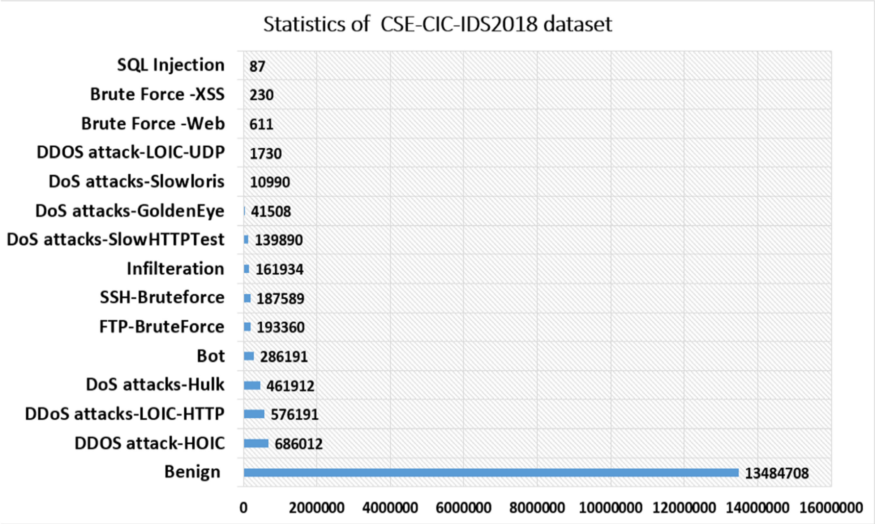

Fig. 6. The statistics of CSE-CIC-IDS2018 dataset.

In this paper, we divided the original 15 categories into two groups: 0 - benign, and 1 - attack. They are further separated into training and test sets as shown in Table 1.

Table 1. Training and test sets from CIC-IDS2017 and CSE-CIC-IDS2018 dataset.

Dataset	Training set	Test set
CIC-IDS2017	2,264,694	566,149
CSE-CIC-IDS2018	12,917,016	3,229,255

Then, we applied feature importance by MDI to select the top features as shown in Table 2.

After applying SMOTE for CIC-IDS2017 dataset, the number of minority instances increases from 445,820 to 1,818,774, and for CSE-CIC-IDS2018 it increases from 2,197,368 to 10,719,648.

In order to implement the LSTM and GRU models, we use the modules from the Keras Python library. Sequential model is a linear stack of layers to initializing the neural network. Dense is a regular layer of neurons in the neural network. A dropout layer is used for implementing regularization technique, which aims to reduce the complexity of the model to prevent overfitting. The architectures of LSTM and GRU both consist of three dimensional input array, one dropout layer, two dense layers, and the output layer which uses softmax function for classification.

The parameters of our model are as follows: Firstly, in the sequential model, and one layer of LSTM or GRU consists of 64 units, which are the dimensionality of the output space. The 3D input shape is the shape of our training set with the format [input

Table 2. Feature importance of CIC-IDS2017 and CSE-CIC-IDS2018 dataset.

Number	CIC-IDS2017		CSE-CIC-IDS2018	
	Features	Weight	Features	Weight
1	init_win_bytes_forward	0.065970	init_fwd_win_bytes	0.158607
2	psh_flag_count	0.061660	fwd_seg_size_min	0.140294
3	bwd_packet_length_mean	0.046262	ack_flag_cnt	0.048100
4	avg_bwd_segment_size	0.042485	init_bwd_win_bytes	0.044378
5	bwd_packet_length_std	0.040715	bwd_pkts_per_s	0.037927
6	packet_length_std	0.034778	flow_pkts_per_s	0.035585
7	bwd_packet_length_max	0.031926	fwd_pkts_per_s	0.032158
8	average_packet_size	0.030894	fwd_pkt_len_max	0.023756
9	bwd_packet_length_min	0.030410	bwd_pkt_len_max	0.019874
10	fwd_iat_max	0.028837	fwd_iat_tot	0.019413
11	min_seg_size_forward	0.027920	fwd_iat_mean	0.018725
12	flow_iat_max	0.026415	flow_iat_min	0.018556
13	packet_length_mean	0.025863	fwd_iat_max	0.018508
14	packet_length_variance	0.022699	flow_duration	0.017454
15	ack_flag_count	0.022489	flow_iat_mean	0.016863

samples, time steps, features]. Secondly, we add a dropout layer with a dropout rate of 0.2, meaning that 20% of the layers will be dropped. Next, the dense layer specifies the output of 2 units (number of classes), and activated with softmax function which normalizes the output to a probability distribution over each output class.

Next, we compile our model using the Adaptive moment estimation (Adam), and sparse categorical cross-entropy loss function to obtain the output. Adam optimizer is implemented for maintaining a learning rate for updating each network weight separately, which can automatically decrease the gradient size steps towards minima based on the exponential moving average of gradients and squared gradients. Sparse categorical cross-entropy loss function is used for our classification since its efficiency and the use of integers as our class labels. Finally, a fitting function is used to fit the model on the data, and we ran the model for ten epochs, with the batch size of 1,000.

To evaluate the performance of intrusion detection, we use evaluation metrics including: Accuracy, Precision, Recall (sensitivity), and F1-score, as shown below.

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

$$Precision = \frac{TP}{FP + TP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1 - Score = \frac{2 * precision * recall}{precision + recall} \quad (15)$$

First, the evaluation results of the classification performance using the full dataset of CIC-IDS2017 are shown in Table 3.

Table 3. Evaluation results using full data (72 features) – CIC-IDS2017.

Classifier	Precision	Recall	F1-score	Accuracy
RF	92.7%	92.7%	92.7%	92.7%
ID3	93.1%	93.1%	**93.1%**	**93.1%**
NB	78.2%	68.6%	71.5%	68.6%
LSTM	89.3%	88.4%	86.5%	**95.7%**
GRU	93.3%	93.2%	**92.8%**	95.1%

As shown in Table 3, for classical learning methods, ID3 gives better performance than Random Forest and Naïve Bayes with an accuracy of 93.1% and an F1-Score of 93.1%. For deep learning methods, better performance can be obtained for LSTM with an accuracy of 95.7%, and GRU with an F1-score of 86.5%.

Next, the evaluation results of CIC-IDS2017 dataset using SMOTE are shown in Table 4.

Table 4. Evaluation results using oversampling (72 features) – CIC-IDS2017.

Classifier	Precision	Recall	F1-score	Accuracy
RF	93.8%	93.7%	93.7%	**93.7%**
ID3	93.8%	93.8%	**93.8%**	**93.7%**
NB	78.6%	77.8%	78.2%	77.8%
LSTM	96.6%	96.2%	96.3%	96.2%
GRU	96.9%	96.5%	**96.6%**	**96.5%**

As shown in Table 4, in classical learning, ID3 gives better performance, with an accuracy of 93.7% and an F1-Score of 93.8%. In deep learning, better performance can be obtained for GRU with an accuracy of 96.6% and an F1-score of 96.5%.

If we applied feature selection by MDI, the evaluation results using 20 selected features are shown in Table 5.

Table 5. Evaluation results using 20 selected features, SMOTE – CIC-IDS2017.

Classifier	Precision	Recall	F1-score	Accuracy
RF	95.2%	94.2%	**94.4%**	**94.2%**
ID3	95.1%	94.1%	94.1%	94.1%
NB	88.8%	82.5%	83.4%	82.5%
LSTM	98.9%	98.9%	**98.9%**	**98.9%**
GRU	98.4%	98.4%	98.4%	98.4%

As shown in Table 5, in classical learning, Random Forest gives better performance, with an accuracy of 94.2% and an F1-score of 94.4%. In deep learning, LSTM gives better performance than GRU, with an accuracy of 98.9% and an F1-score of 98.4%.

If we further reduce the number of selected features, the evaluation results are shown in Table 6.

Table 6. Evaluation results using 10 selected features, SMOTE – CIC-IDS2017.

Classifier	Precision	Recall	F1-score	Accuracy
RF	95.1%	94.1%	**94.1%**	**94.1%**
ID3	94.8%	93.9%	93.9%	94.0%
NB	88.5%	80.9%	81.9%	80.9%
LSTM	98.6%	98.6%	**98.6%**	**98.6%**
GRU	98.1%	98.1%	98.1%	98.1%

As shown in Table 6, in classical learning, Random Forest gives better result, with an accuracy of 94.1% and an F1-score of 94.1%. In deep learning, we found LSTM gives better performance than GRU with an accuracy of 98.6% and an F1-score of 98.6%.

From the performance comparison of results from Tables 5 and 6, we found in classical learning, Random Forest gives the best result with an accuracy of 94.4% and

Table 7. Evaluation results full data (72 features) – CSE-CIC-IDS2018.

Classifier	Precision	Recall	F1-score	Accuracy
RF	89.0%	88.0%	89.0%	89.1%
ID3	93.3%	93.2%	**93.2%**	**93.2%**
NB	62.9%	50.1%	55.0%	49.1%
LSTM	81.1%	81.0%	**89.9%**	**85.0%**
GRU	87.1%	86.8%	83.7%	84.7%

an F1-score of 94.2%. In deep learning, LSTM gives the best result, with an accuracy of 98.9% from 20 selected features, and an F1 score of 98.9%. This shows the effectiveness of the proposed feature selection and deep learning methods. Next, we do the same for the CSE-CIC-IDS2018 dataset as shown in Table 7.

As shown in Table 7, in classical learning, the best performance can be obtained for ID3 with an accuracy of 93.2% and an F1-score of 93.2%. In deep learning, the best performance can be obtained for LSTM with an accuracy of 85.0%, and an F1-score of 89.9%. Then, evaluation results using SMOTE oversampling are shown in Table 8.

Table 8. Evaluation results of using oversampling (72 features) – CSE-CIC-IDS2018.

Classifier	Precision	Recall	F1-score	Accuracy
RF	90.0%	88.9%	86.6%	88.9%
ID3	93.9%	93.7%	**93.8%**	**93.7%**
NB	85.7%	50.7%	55.0%	50.7%
LSTM	91.1%	91.0%	**89.9%**	**95.2%**
GRU	87.1%	86.8%	83.7%	94.7%

As shown in Table 8, in classical learning, ID3 gives better performance, with an accuracy of 93.7% and an F1-score of 93.8%. It's better than deep learning methods in F1-score, where LSTM gives better accuracy of 95.2%. Then, the evaluation results using 20 selected features are shown in Table 9.

Table 9. Evaluation results of using 20 selected features, SMOTE – CSE-CIC-IDS2018.

Classifier	Precision	Recall	F1-score	Accuracy
RF	92.1%	92.1%	92.1%	92.1%
ID3	94.6%	94.6%	**94.7%**	**94.6%**
NB	82.9%	75.1%	75.1%	75.1%
LSTM	98.0%	97.9%	97.9%	97.9%
GRU	98.9%	98.8%	**98.8%**	**98.8%**

As shown in Table 9, in classical learning, ID3 gives better performance, with an accuracy of 94.6% and an F1-score of 94.7%. In deep learning, GRU gives better performance, with an accuracy of 98.8% and an F1-score of 98.8%.

Finally, the evaluation results using 10 selected features are shown in Table 10.

Table 10. Evaluation results using 10 selected features, SMOTE – CSE-CIC-IDS2018.

Classifier	Precision	Recall	F1-score	Accuracy
RF	91.5%	91.5%	91.5%	91.4%
ID3	94.2%	94.2%	**94.2%**	**94.2%**
NB	82.5%	74.5%	74.5%	74.5%
LSTM	97.7%	97.6%	97.6%	97.6%
GRU	98.1%	98.1%	**98.1%**	**98.1%**

As shown in Table 10, in classical learning, ID3 gives better result with an accuracy of 94.2% and an F1-score of 94.2%. In deep learning, we found GRU gives better performance with an accuracy of 98.1% and an F1-score of 98.1%.

When comparing Tables 9 and 10, GRU shows the best F1-score and accuracy of 98.8%. In classical learning, ID3 gives the best performance with an accuracy of 94.6% and an F1-score of 94.7%.

In summary, when we compare the evaluation results for the two datasets, the best performance can be obtained using different methods: LSTM and RF for CIC-IDS2017 dataset, and GRU and ID3 for CSE-CICIDS2018 dataset. There's only slight difference between the best performance of LSTM and GRU. Also, we can see comparable performance when using only 10 selected features. This shows the effectiveness of combining the MDI feature selection method, and SMOTE oversampling method in recurrent neural networks.

5 Conclusions

In this paper, we aimed at intrusion detection using deep learning methods. In this context, the CIC-IDS2017 and CSE-CICIDS2018 datasets were used since they are up-to-date with wide attack diversity, and various network protocols (e.g., Mail services, SSH, FTP, HTTP, and HTTPS). First, by using a feature selection method, we can determine the most important features in both datasets. Then, it is combined with oversampling technique to deal with imbalanced data. The experimental results show that our results are better than existing works to classify and detect intrusions. In CIC-IDS2017 dataset, the best performance obtained for the proposed method is an accuracy of 98.9 and an F1-score of 98.9% by LSTM. Second, in CSE-CIC-IDS2018 dataset, the best performance can be obtained for GRU with an accuracy of 98.8% and an F1-score of 98.8%. Third, by using the top 10 selected features, the performance is better than using all features. This shows the effectiveness of our proposed method for using feature selection and oversampling for intrusion detection in large scale network traffic.

In future, we plan to use other datasets which include new variants of attacks like malware and backdoor activity in real network traffics. Besides, we want to compare with other feature selection methods. For under-sampling or over-sampling technique, to adjust the class distribution in the dataset, we can use the weight of distribution of minority class, to generate more synthetic data for the minority class. Finally, we plan to combine deep learning with other classification methods for improving the performance.

Acknowledgements. This work was partially supported by research grants from Ministry of Science and Technology, Taiwan, under the grant number MOST109–2221-E-027–090, and partially supported by the National Applied Research Laboratories, Taiwan under the grant number of NARL- ISIM-109–002.

References

1. IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. www.unb.ca, 2017. https://www.unb.ca/cic/datasets/ids-2017.html. Accessed 15 June 2019
2. A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018) (2018). https://registry.opendata.aws/cse-cic-ids2018/. Accessed 15 June 2019
3. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., Habibi Lashkari, A., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: Proceedings of 4th International Conference. Information System Security Privacy, pp. 108–116 (2018)
4. Louppe, G., Wehenkel, L., Sutera, A., Geurts, P.: Understanding variable importances in forests of randomized trees. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1 (NIPS 2013), pp. 431–439 (2013)
5. Chawla, K.W., Bowyer, L., Hall, O., Kegelmeyer, W.P.: SMOTE: synthetic minority oversampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
6. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
7. Chen, J., Luo, D., Mu, F.: An improved ID3 decision tree algorithm. In: 2009 4th International Conference on Computer Science & Education, pp. 127–130 (2009)
8. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn **29**, 131–163 (1997)
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
10. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on Deep Learning (2014)
11. Albayati, M., Issac, B.: Analysis of intelligent classifiers and enhancing the detection accuracy for intrusion detection system. Int. J. Comput. Intell. Syst. 841–853 (2015)
12. Almseidin, M., Alzubi, M., Kovacs, S., Alkasassbeh, M.: Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th International Symposium Intelligent System Informatics (SISY), pp. 277–282 (2017)
13. Khuphiran, P., Leelaprute, P., Uthayopas, P., Ichikawa, K., Watanakeesuntorn, W.: Performance comparison of machine learning models for DDoS attacks detection. In: 2018 22nd International Computer Science and Engineering Conference (ICSEC), pp. 1–4 (2018)
14. Althubiti, S.A., Jones, E.M., Roy, K.: LSTM for anomaly-based network intrusion detection. In: 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), pp.1–3 (2018)
15. Xu, C., Shen, J., Du, X., Zhang, F.: An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access **6**, 48697–48707 (2018)

16. Alazzam, H., Sharieh, A., Sabri, K.E.: A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer. Expert Syst. Appl. **148** (2020)
17. Wu, M.Y., Shen, C.-Y., Wang, E.T., Chen, A.L.P.: A deep architecture for depression detection using posting, behavior, and living environment data. J. Intell. Inf. Syst. **54**(2), 225–244 (2018). https://doi.org/10.1007/s10844-018-0533-4
18. Shuai, H.-H., et al.: A comprehensive study on social network mental disorders detection via online social media mining. IEEE Trans. Knowl. Data Eng. (TKDE) **30**(7), 1212–1225 (2018)
19. Smiti, S., Soui, M.: Bankruptcy prediction using deep learning approach based on borderline SMOTE. Inf. Syst. Front. **22**(5), 1067–1083 (2020). https://doi.org/10.1007/s10796-020-10031-6
20. Seo, J.-H., Kim, Y.-H.: Machine-learning approach to optimize SMOTE ratio in class imbalance dataset for intrusion detection. Comput. Intell. Neurosci. 1–11 (2018)
21. Kurniabudi, D.S., Darmawijoyo, M.Y.B.I., Bamhdi, A.M., Budiarto, R.: CICIDS-2017 dataset feature analysis with information gain for anomaly detection. IEEE Access **8**, 132911–132921 (2020)
22. Kim, J., Kim, J., Kim, H., Shim, M., Choi, E.: CNN-based network intrusion detection against denial-of-service attacks. Electronics **9**(6) (2020)