POPULATION-BASED FEATURE SELECTION FOR INTRUSION DETECTION

Nikola Stevanović[1]

[1] Faculty of Sciences and Mathematics
University of Niš, Višegradska 33, 18000 Niš
Email: nikola.stevanovic@pmf.edu.rs

Abstract

Computer networks have become an integral part of people's everyday lives. In addition to simplifying and making many of our activities more efficient, their rise of importance has also created an opportunity for fraudsters to obtain illegal profits. To protect networks from intrusions, many network features (measures) are collected and monitored to detect malicious activities. Since intrusion detection has to be fast, it is important to select appropriate features to analyze. In this paper, for the purpose of fast and intelligent intrusion detection, we propose a population-based method for network traffic feature selection, which we combine with various machine learning classifiers. Since different features can influence various classifiers differently, our method is able to select appropriate features for each classifier separately. We prove the effectiveness of our approach by experimental evaluation on the UNSW-NB15 intrusion detection dataset. It was able to improve the accuracy for all the classifiers, while significantly reducing the number of input features at the same time.

Key words: intrusion detection, feature selection, population-based methods

1 Introduction

With the constantly increasing number of malicious cyber activities, it is important to create efficient solutions for protection against them. Any set of actions that attempt to compromise functionality of a system can be termed as an intrusion (attack). An attack can cause changes in network traffic parameters, which intrusion detection systems try to utilize to protect computer networks.

Several datasets have been created to test intrusion detection solutions. Among earlier datasets, the KDDCup'99 [1] dataset is still frequently used for evaluation, even though it is more than two decades old. The NSLKDD [2] dataset was later introduced to mitigate some of the problems that the KDDCup'99 dataset had, such as redundant records, imbalance between normal and malicious records and missing values. Despite the improvement, NSLKDD is not a comprehensive representation of modern low footprint attacks. Guided by those shortcomings, a cybersecurity research group from the Australian Centre for Cyber Security created a new dataset, named UNSW-NB15 [3], which better represents more modern types of attacks. That is why we decided to choose the UNSW-NB15 dataset to evaluate our approach.

Intrusion detection datasets contain a lot of features. To make detection systems faster and more efficient, researchers have utilized various features selection strategies. Janarthanan and Zargari [4] experimented with a few methods for feature selection using the Weka tool. They finally selected five important features, and obtained an accuracy

rate of 81.62% using them. Khan et al. [5] utilized feature importance technique to select a set of 11 input features from the UNSW-NB15 dataset. They achieved the highest accuracy of 75.66% by applying a random forest classifier. More information about current approaches in intrusion detection could be found in survey [6].

2 Methodology

2.1 Dataset

Each sample in the UNSW-NB15 intrusion detection dataset originally contains 42 input features. Detailed explanation of all the features is given in [3]. Three of them are not numerical (protocol that is used, its state and the service that is running), and since most machine learning classifiers expect numerical inputs, we had to transform them. We have achieved that by collecting all possible values that a non-numerical feature can have, and then assigning each value a non-negative auto increment integer value.

The dataset is divided in its training and testing parts. Since we also need one additional part to be used for validation, we randomly selected 30% of samples from the testing dataset to be used for that purpose. The remaining 70% of samples are left to be used for testing. Final sizes of the training, validation and testing datasets used in our experiments are given in Table 1.

Since different features have values that can significantly differ in scale, we normalize each feature by subtracting from it the minimal value of the feature and then dividing by the difference between the maximal and minimal value of the feature. Maximal and minimal values are calculated based on the training dataset.

Dataset	Regular samples	Malicious samples
Training	56000	119341
Validation	11087	13613
Testing	25913	31719

Table 1: Training, validation and testing dataset sizes

2.2 Proposed approach

In this paper, we propose a population-based strategy for selecting appropriate features for intrusion detection. Elements of a population in our case are sets of input features. Each element can be represented with a 0/1 vector of size $N = 42$ (total number of features), where one indicates that the features is selected, and zero that it is not.

Each generation $g \in \{1, 2, \ldots, G\}$ is defined by its vector of probabilities $p^g \in [0, 1]^N$. Value p_i^g indicates the probability that the i-th feature should be selected in the g-th generation. Generation g is created by sampling S instances based on its vector of probabilities p^g. Let us denote with $e_1^g, \ldots, e_S^g$ those instances (elements of the population).

Our strategy does not keep instances from past generations. The way they influence next generations is through the vector of probabilities. For the first generation, all the probabilities p_i^1 are initialized to the same value p_{init}. For any other subsequent generation $g > 1$, its vector of probabilities is calculated based on the scores that instances of the previous $(g - 1)$-th generation obtained. In our case, we define the score of an

instance e_i^g as the accuracy on the validation dataset of the model obtained by training on the training dataset and using input features e_i^g, and we label that value as s_i^g.

To calculate the vector of probabilities p^g, we first calculate contributions of individual elements of the previous generation. Let us denote with c_i^g contribution of the i-th element of generation g to generation $g+1$. We can calculate c_i^g using formula (1).

$$Min^g = \min_j s_j^g, \quad Max^g = \max_j s_j^g, \quad \overline{c}_i^g = \frac{s_i^g - Min^g}{Max^g - Min^g}, \quad c_i^g = \frac{\overline{c}_i^g}{\sum_{j=1}^{S} \overline{c}_j^g} \tag{1}$$

In a theoretical special case when $Min^g = Max^g$, all contributions should be set to $c_i^g = 1/S$. As we can see from the formula, instances with higher accuracies will have higher contribution values. Using these contribution values, we can now calculate the vector of probabilities p^g as shown in formula (2). Please note that in this equation, c_j^{g-1} is a scalar and e_j^{g-1} is a 0/1 vector. In the formula we also use a small hyperparameter α, which is used to give each input feature some positive probability to both be and not be selected in the next generation.

$$\overline{p}^g = \sum_{j=1}^{S} c_j^{g-1} e_j^{g-1}, \qquad p^g = (1-\alpha)\overline{p}^g + 0.5\alpha \tag{2}$$

In our approach, we evaluate $G \cdot S$ input feature subsets in total, and select the subset with the highest accuracy on the validation dataset. That number is significantly smaller than the total number of possible subsets, which is 2^N.

3 Evaluation

To evaluate our method, we have combined our features selection strategy with various machine learning algorithms, often used in intrusion detection systems. We have utilized the following classifiers: decision tree, random forest, AdaBoost, logistic regression, k-nearest neighbors and fully connected neural network. All the models are imported from the Scikit-learn library and used with their default settings. We have run all the experiments for $G = 15$ generations of $S = 15$ instances. Vectors of probabilities were initialized with $p_{init} = 0.1$, and we have set the hyperparameter to $\alpha = 0.1$.

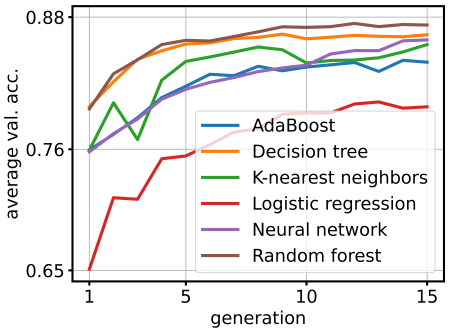

Figure 1: Average validation accuracy per generation

Figure 1 shows how average validation accuracy changes over generations for each classifier. Accuracies are averaged over all instances of one generation. We can see a

continuous increase of average accuracy for all the classifiers, with only some occasional small drops.

Model	Acc. (all features)	Acc. (selected features)	Selected features
Decision tree	0.8630	0.8824	14/42
Random forest	0.8727	0.8812	15/42
AdaBoost	0.8522	0.8604	22/42
Logistic regression	0.8028	0.8113	17/42
K-nearest neighbors	0.8429	0.8762	23/42
Neural network	0.8674	**0.8891**	23/42

Table 2: Classification results on the testing dataset

Classification results on the testing dataset are given in Table 2. As we can see from the table, all the models improved their accuracy using our feature selection method. The biggest improvement was for the k-nearest neighbors classifier, for which classification accuracy increased for more than 3%. The best accuracy of 88.91% was obtained by the neural network model. The decision tree classifier has also obtained good testing accuracy, by selecting only 14 input features (33.3% of all the features).

4 Conclusions

In this paper, we have proposed a network traffic feature selection method for intrusion detection. We have shown its effectiveness by comparing classification accuracies of several machine learning classifiers when using all input features or their subset selected by our method. Our method improved accuracy of all the classifiers, while significantly reducing the number of input features. In the future, we would like to test our approach on more datasets.

References

[1] KDD Cup 1999 Intrusion detection dataset, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[2] Tavallaee M, Bagheri E, Lu W, Ghorbani AA, A detailed analysis of the kdd cup 99 data set. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, 2009.

[3] Moustafa N, Slay J, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 military communications and information systems conference (MilCIS), IEEE, 2015.

[4] Janarthanan T, Zargari S, Feature selection in UNSW-NB15 and KDDCUP'99 datasets. In 2017 IEEE 26th international symposium on industrial electronics (ISIE), IEEE, 2017.

[5] Khan NM, Madhav C N, Negi A, Thaseen IS, Analysis on improving the performance of machine learning models using feature selection technique. In international conference on intelligent systems design and applications, Springer, Cham, 2018.

[6] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2(20):1-22, 2019.