COMPARATIVE ANALYSIS OF INFORMATIVE FEATURES QUANTITY AND COMPOSITION SELECTION METHODS FOR THE COMPUTER ATTACKS CLASSIFICATION USING THE UNSW-NB15 DATASET

DOI: 10.36724/2072-8735-2020-14-10-53-60

Oleg I. Sheluhin,
Moscow Technical University of Communication and
Informatics, Moscow, Russia, sheluhin@mail.ru

Manuscript received 05 August 2020;
Accepted 21 September 2020

Valentina P. Ivannikova,
Moscow Technical University of Communication and
Informatics, Moscow, Russia, iv8post@gmail.com

Keywords: Feature selection, machine learning, binary
classification, network attacks, UNSW-NB15 dataset

A comparative analysis of statistical and model-based methods for selecting the quantity and the composition of informative features was performed using the UNSW-NB15 database for machine learning models training for attack detection. Feature selection is one of the most important steps in data preparation for machine learning tasks. It allows to increase a quality of machine learning models: it reduces sizes of the fitted models, training time and probability of overfitting. The research was conducted using Python programming language libraries: scikit-learn, which includes various machine learning models and functions for data preparation and models estimation, and FeatureSelector, which contains functions for statistical data analysis. Numerical results of experimental research of application of both statistical methods of features selection and machine learning models-based methods are provided. As the result, the reduced set of features is obtained, which allows improving the quality of classification by removing noise features that have little effect on the final result and reducing the quantity of informative features of the data set from 41 to 17. It is shown that the most effective among the analyzed methods for feature selection is the statistical method SelectKBest with the function chi2, which allows to obtain a reduced set of features providing an accuracy of classification as high as 90% in comparison with 74% provided with the full set.

Information about authors:

Oleg I. Sheluhin, doctor of technical sciences, professor, head of the Department of Information Security, Moscow Technical University of Communication and Informatics, Moscow, Russia

Valentina P. Ivannikova, undergraduate, Moscow Technical University of Communication and Informatics, Moscow, Russia

Для цитирования:

Шелухин О.И., Иванникова В.П. Сравнительный анализ способов отбора количества и состава информативных признаков компьютерных атак на примере базы данных UNSW-NB15 // T-Comm: Телекоммуникации и транспорт. 2020. Том 14. №10. С. 53-60.

For citation:

Sheluhin O.I., Ivannikova V.P. (2020) Comparative analysis of informative features quantity and composition selection methods for the computer attacks classification using the UNSW-NB15 dataset. *T-Comm*, vol. 14, no.10, pp. 53-60. (in Russian)

Problem Formulation

The detection of network computer attacks and timely response to them are some of the most important things in a field of information security and these problems can be resolved by machine learning algorithms. Implementing computer attacks detection algorithms, demands to make them as fast and reliable as possible. At the same time, datasets used for training computer attacks classifiers may contain features, that are redundant or non-informative and can be removed without quality loss [1]. Feature selection allows simplifying classifiers models, reducing the learning time and the probability of overfitting, which in turn allows to increase a quality of software.

The similar problem formulation has been reviewed in a number of publications. An approach to the feature selection in NSL-KDD dataset which is based on feature importance estimation calculated using the Random Forest algorithm was reviewed in [2]. By recursively removing two least important features at each iteration, authors have succeeded in reducing the initial set of parameters from 41 to 19, improving the quality of the model.

Another approach which uses Random Forest algorithm was demonstrated in [3], where the one-time estimate of feature importance was used and the selection of features, whose importance estimate, according the algorithm, was greater than the median value, was made. As the result, 12 of 24 features were selected. A different approach using the statistical features' parameters, such as redundancy relative to other features, feature correlation with the result and computation time of the feature, as the basis, was developed by authors for the feature importance calculation in [4] and the subsequent selection was performed.

In [5] approaches to the feature selection, based on Extra Tree machine learning algorithm and SelectKBest class from scikit-learn library demonstrated good results in reducing the quantity of features on the example of three different datasets.

In contrast to those works, in this article results of comparative analysis of different methods of feature selection are shown.

The aim of the article is the comparative analysis of the most widespread approaches to the selection of a reduced set of informative features for classification task, which allows increasing the quality of the final model.

Dataset description

The research of approaches to the feature selection was performed on modern UNSW-NB15 dataset, which was created in 2015 in the Cyber Range Lab of the Australian Centre for Cyber Security «for generating a hybrid of real modern normal activities and synthetic contemporary attack behaviors» [6]. The dataset contains records about 2.5 million network flows, which are characterized by 49 informative features, 41 from them represent useful information about connection (these features are shown in Table 1). All network flows have a label, that indicates whether the flow is normal network traffic or contains an attack. The traffic with attacks is additionally labeled by 9 attack categories.

The network flows' features in the dataset represent the following information: base flow features (number of packets, time to live, bits per second, etc.), content features (mean of the flow packet size, TCP window advertisement, etc.), time features (jitter, inter-packet arrival time, etc.) and additional generated statistical features (for example, how often flows with the same features were encountered).

Features of the UNSW-NB15 dataset

Table 1

№	Feature	Description	№	Feature	Description
1	du	Record total duration	22	dwin	Destination TCP window advertisement value
2	p_oto	Transaction protocol	23	tcprt	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'
3	servic	Used service	24	synack	TCP connection setup time, the time between the SYN and the SYN_ACK packets
4	state	Indicates to the state and its dependent protocol	25	ackdat	TCP connection setup time, the time between the SYN_ACK and the ACK packets
5	spkt	Source to destination packet count	26	meansz	Mean of the flow packet size transmitted by the source
6	dpkt	Destination to source packet count	27	meansz	Mean of the flow packet size transmitted by the destination
7	sb_tes	Source to destination transaction bytes	28	trans_dept_h	Represents the pipelined depth into the connection of http request/response transaction
8	db_tes	Destination to source transaction bytes	29	re-sponse_body_len	Actual uncompressed content size of the data transferred from the server's http service
9	sttl	Source to destination time to live value	30	ct_srv_src	No. of connections that contain the same service and source address in 100 connections according to the last time
10	dtl	Destination to source time to live value	31	ct_state_ttl	No. for each state according to specific range of values for source/destination time to live
11	sload	Source bits per second	32	ct_dst_ltm	No. of connections of the same destination address in 100 connections according to the last time
12	dload	Destination bits per second	33	ct_src_dp_port_ltm	No of connections of the same source address and the destination port in 100 connections according to the last time
13	sloss	Source packets retransmitted or dropped	34	ct_dst_sport_ltm	No of connections of the same destination address and the source port in 100 connections according to the last time
14	dloss	Destination packets retransmitted or dropped	35	ct_dst_src_ltm	No of connections of the same source and the destination address in 100 connections according to the last time
15	sintpkt	Source interpacket arrival time (mSec)	36	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0
16	dintpkt	Destination interpacket arrival time (mSec)	37	ct_ftp_cmd	No of flows that has a command in ftp session
17	sji	Source jitter (mSec)	38	ct_flw_http_mthd	No. of flows that has methods such as Get and Post in http service
18	dji	Destination jitter (mSec)	39	ct_src_ltm	No. of connections of the same source address in 100 connections according to the last time

19	swin	Source TCP window advertisement value	40	ct_srv_dst	No. of connections that contain the same service and destination address in 100 connections according to the last time
20	stepb	Source TCP base sequence number	41	is_sm_ips_ports	If source and destination IP addresses equal and port numbers equal then, this variable takes value 1 else 0
21	dtpcb	Destination TCP base sequence number			

Methods used for feature selection

There are several approaches that can be used for feature selection.

Model-based approach uses simple machine learning models which can be used to estimate the importance of features. Decision Tree ensembles or linear models that reset the weight of redundant features are usually used as a simple model. There are six models which has been used in this research: Decision Tree and Extra Tree as the unit models and their ensembles - Random Forest, AdaBoost, Gradient Boosting (all of these ensembles use Decision Tree as a basic classifier) and Extra Trees (it uses Extra Tree as a basic classifier).

Within the scikit-learn library's models feature importance calculates as the (normalized) total reduction of the criterion brought by that feature. This approach is known as Gini importance or Mean Decrease in Impurity as well.

Statistical approach uses statistical methods for feature selection. For example, it is possible to remove features whose values in the dataset do not change or change insignificantly.

The following statistical features' selection tools are reviewed: FeatureSelector, which contains functions for statistical data analysis, and SelectKBest from scikit-learn Python library for machine learning tasks.

FeatureSelector tool allows to select features by the following characteristics:

- Collinearity – identification of the feature pairs with strong correlation among themselves. The correlation is calculated by formula 1:

$$r = \frac{n(\sum f c) - (\sum f)(\sum c)}{\sqrt{[n\sum f^2 - (\sum f)^2][n\sum c^2 - (\sum c)^2]}} \quad (1)$$

where n – quantity of records in the dataset, f – feature values, c – class values. While working with strongly correlated features, it is necessary to combine them into a single feature or remove one of them.

- Zero/low importance – the importance of features is estimated using the gradient boosting algorithm from the LightGBM library.

- Features with a single value – identification of features whose values do not change, and therefore do not affect the result.

- Percentage of missed features – calculation of percentage of missed values among all values of the feature.

SelectKBest algorithm uses three functions to determine the importance of features in classification tasks:

- chi2 – the function for the chi-square characteristic calculation that shows the dependence between the values of each feature and the class and calculates by the formula 2:

$$\chi^2 = \sum_{c=1}^C \sum_{f=1}^F \frac{(O_{cf} - E_{cf})^2}{E_{cf}}, \quad (2)$$

where C – quantity of classes, F – quantity of features value, O_{cf} and E_{cf} – observed and expected frequencies of the feature with a value f in class c . The expected frequency is calculated as the probability of two independent events (formula 3):

$$E_{ij} = N * P(i = c \cap f) = N * (P(c) * P(f)), \quad (3)$$

where N – quantity of all record in dataset, $P(c)$ and $P(f)$ – probabilities of having a record with a class label c or with a feature value f among all records.

- f_classif – the function for calculating the F-criterion using dispersion analysis based on the examination of significance of differences in mean values of features and aimed to find dependencies in the data. The function is calculated by formula 4:

$$F = \frac{\frac{1}{C-1} \sum_{i=1}^C N_i (\bar{x}_i - \bar{x})^2}{\frac{1}{N-C} \sum_{i=1}^C \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)^2}, \quad (4)$$

where C – quantity of classes, N – quantity of records in dataset, N_i – quantity of records with class label i , $x_{i,j}$ – value of j -th feature of class i , $\bar{x}_i$ – mean feature value in class i , $\bar{x}$ – mean feature value in dataset.

- mutual_info_classif – the function based on the evaluation of the entropy by distances of k -nearest neighbors for an estimation of mutual information between features. The function is calculated by formulae 5-8:

$$MI(F, C) = \psi(k) + \psi(N) - \frac{1}{N} \sum_{n=1}^N [\psi(\tau_f(n)) + \psi(\tau_c(n))], \quad (5)$$

$$\psi(t) = \frac{\Gamma'(t)}{\Gamma(t)} = \frac{d}{dt} \ln \Gamma(t), \quad (6)$$

$$\Gamma(t) = \int_0^\infty u^{t-1} e^{-u} du, \quad (7)$$

$$\epsilon_n = \max(\|f_n - f_{k(n)}\|, \|c_n - c_{k(n)}\|). \quad (8)$$

where F – feature values, C – class values, k – quantity of nearest neighbors, N – quantity of records in dataset, $\tau_f(n)$ – quantity of values $f \in F$ whose distance to f_n strictly less than ϵ_n , $\tau_c(n)$ – quantity of values $c \in C$ whose distance to c_n strictly less than ϵ_n .

Brute-force approach involves the process of the model training with all possible subsets of features and selecting a subset that demonstrates the best overall quality. This approach is the simplest and can guarantee the best result, but usually it is not appropriate when the quantity of features is large as the run time of the search algorithm for n features grows exponentially as $O(2^n)$ [7].

Evaluation metrics of classification algorithms

Evaluation metrics that are used for classification [8]:

- accuracy – the proportion of correctly classified flows;
- recall – the fraction of correctly classified flows with attacks in the whole set of flows with attacks;
- precision – the fraction of correctly classified flows with attacks in the whole set of flows classified as flows with attacks;
- F-score – harmonic mean of precision and recall metrics;
- ROC-AUC – area under ROC-curve.

The most representative metric in the binary classification quality evaluation is ROC-AUC.

Model-based feature selection

The model-based feature selection algorithm implements recursive removal of the least important feature following with the subsequent model performance evaluation with the remaining set until the minimal feature quantity is reached. This algorithm can be described as follows:

Algorithm 1 – Model-based feature selection

Designations:

trainX – features of the training set; trainY – class labels of the training set; testX – features of the testing set; testY – class labels of the testing set; n – minimum quantity of features; model – a model used for feature selection; F – the list of used features; f – the reduced list of used features; acc – accuracy metric value for the model built on the reduced set of features; auc – ROC-AUC metric value for the model built on the reduced set of features; result – the result of the trained model; resultAcc – accuracy metric value for the trained model; resultAuc – ROC-AUC metric value for the trained model; E – the list of featured sorted by estimated importance.

Algorithm:

```
function FeatureSelectionWithModel(trainX, trainY, testX,
testY, n, model)
    F = GetFeatures(trainX)
    (f, acc, auc) = InitialValues()
    while Length(F) < n do
        model = ModelTrain(model, trainX[F], trainY)
        result = ModelPredict(model, testX)
        (resultAcc, resultAuc) = GetMetrics(testX, re-
sult)
        if resultAuc > auc then
            (f, acc, auc) = (F, resultAcc,
resultAuc)
        end if
        E = GetFeaturesSortByEstimate(model)
        F = DropLastFeature(E)
    end while
end function
```

In order to obtain the most reliable data, training the model and calculating the features importance at each stage were performed ten times and then the average values were calculated. It should be mentioned that all algorithms were trained with default hyperparameters.

Results of feature selection using this model-based algorithm for six machine learning algorithms (Random Forest, Extra Trees, AdaBoost, Gradient Boosting, Decision Tree и Extra Tree) are shown in the Table 2.

Table 2

Classification results of model-based algorithms

Algorithm	Features count	Accuracy	Precision	Recall	F-score	ROC-AUC
Decision Tree	20	0.706	0.753	0.694	0.722	0.707
Extra Tree	30	0.830	0.882	0.798	0.838	0.833
Random Forest	32	0.8313	0.787	0.949	0.861	0.818
Extra Trees	27	0.840	0.787	0.971	0.870	0.825
AdaBoost	11	0.762	0.749	0.854	0.798	0.752
Gradient Boosting	13	0.769	0.757	0.854	0.803	0.759

As a modeling outcome, the best results were shown for the set of 27 features achieved by the Extra Trees algorithm. For this reduced set results reach a good level of flows classification (accuracy metric) – 84% and a high level of attack detection (recall metric) – 97%. According to ROC-AUC and precision metrics, the best result was shown by the set of 30 attributes achieved by the Extra Tree algorithm, but it is significantly worse than the ensemble version of this algorithm (Extra Trees) by recall metric, which indicates a relatively low quality of valid attack detection.

Statistical approach to the feature selection

FeatureSelector tool. As the consequence of using FeatureSelector tool the following results were achieved:

- 11 pairs of features have a correlation over 95%. The full heat map of features correlation is shown in Figure 1. Based on this, it is proposed to remove following 8 features (some features have high correlation with more than one another feature) from the set:
 - sbytes (7) – correlates by 96.5% with spkts (5);
 - dbytes (8) – correlates by 97.6% with dpkts (6);
 - loss (13) – correlates by 97.3% with spkts (5) and by 99.5% with sbytes (7);
 - dloss (14) – correlates by 98.1% with dpkts (6) and by 99.7% with dbytes (8);
 - dwin (22) – correlates by 95.6% with state and by 96.0% with swin (19);
 - ct_src_dport_ltm (33) – correlates by 96.0% with ct_dst_ltm (32);
 - ctftp_cmd (37) – correlates by 99.4% with isftp_login (36);
 - ct_srv_dst (40) – correlates by 97.7% with ct_srv_src (30).
- 31 features have a 99% effect on the result and 10 features have a low influence (features state (4), dtl (10), swin (19), dwin(22), trans_depth(28), ct_state_ttl(31), isftp_login (36), ctftp_cmd (37), ctflw_http_mthd (38), ct_srv_dst (40)).
- Feature dwin (22) have a zero importance.
- There are no features with single value or missed values.

Table 4

Classification results of the statistical approach using FeatureSelector tool with the least important features removal

Algorithm	Decision Tree	Extra Tree	Random Forest	Extra Trees	AdaBoost	Gradient Boosting	Average result
Accuracy	0.892	0.860	0.898	0.893	0.897	0.899	0.890
Precision	0.979	0.966	0.989	0.988	0.983	0.987	0.982
Recall		0.824	0.859	0.853	0.862	0.864	0.854
F-score	0.915	0.889	0.919	0.915	0.919	0.921	0.913
ROC-AUC	0.910	0.881	0.919	0.915	0.916	0.920	0.910

The removal of both collinear and least important features made possible to reduce the quantity of features in data set from 41 to 25 while preserving high quality of classifier training, as shown by results in Table 5.

Table 5

Classification results of statistical approach using FeatureSelector tool with both collinear and least important features removal

Algorithm	Decision Tree	Extra Tree	Random Forest	Extra Trees	AdaBoost	Gradient Boosting	Average result
Accuracy	0.891	0.888	0.896	0.887	0.896	0.901	0.893
Precision	0.980	0.977578	0.988	0.987	0.980	0.985	0.983
Recall		0.855	0.857	0.845	0.866	0.867	0.858
F-score	0.915	0.912	0.918	0.910	0.919	0.923	0.916
ROC-AUC	0.910622	0.906	0.918	0.911	0.914	0.920	0.913

As can be seen, by removing both collinear and least important features, the average accuracy metric value gets as high as 89.3%. Average results of all other metrics also demonstrate high quality of classification on the set of features reduced from 41 to 25 values.

The best of three reduced feature sets' results obtained using FeatureSelector was obtained with the collinear features only removal from the initial set: the accuracy metric hit 89.4% and the ROC-AUC metric hit 91.4% within this strategy.

With both methods combined i.e. both collinear features and least importance features removal, the overall result have become worse, than with the use of collinear features only removal. This deterioration is due to the fact that the least important features removal affects some of the features, whose collinear pairs were deleted and therefore some flow characteristic information were lost. At the same time least importance features only removal shows the worst result of all exploited combinations. This could be consequence from the fact that collinear features' pairs that stayed in the reduced set have more negative effect on the result than the positive effect from least importance features removal.

SelectKBest tool. The selection of features for this tool is similar to one that had been used for the model-based method and was performed as shown in Algorithm 2. In contradiction with the model-based approach, this one assumes that the features importance evaluates only once, and then the features that have less than a certain importance value got removed.

Since this algorithm do not assumes repetition in feature importance evaluation on every step within the reducing dataset, the

Figure 1. The heat map for UNSW-NB15 dataset features

Classification results obtained by removing collinear features and training for six machine learning algorithms are shown in Table 3.

Table 3

Classification results of the statistical approach using FeatureSelector tool with the collinear features removal

Algorithm	Decision Tree	Extra Tree	Random Forest	Extra Trees	AdaBoost	Gradient Boosting	Average result
Accuracy		0.890	0.898	0.886	0.899	0.903	0.894
Precision	0.979	0.973	0.988	0.986	0.980	0.985	0.982
Recall		0.863	0.860	0.844	0.870	0.871	0.860
F-score	0.913	0.915	0.920	0.910	0.922	0.924	0.917
ROC-AUC	0.908	0.906	0.919	0.910	0.916	0.921	0.914

From the results shown in table 3 succeeds that the removal of 8 collinear features from the dataset and reducing their overall quantity from 41 to 33 allows results improvement in comparison with the model-based approach. The removal of features which have lesser importance in model-based approach gives accuracy value in the range from 70.6% to 84.0%, while the collinear features removal allows to increase accuracy metric value from 88.6% to almost 90.3% depending on classification machine learning algorithm being used. In comparison with the model-based approach, the average result of six algorithms presented in table 3 shows improvement for all metrics except recall.

Removal of least important features results for six machine learning algorithms and their training are presented in Table 4. In this case feature set is reduced from 41 to 31 features.

As can be seen from the presented results, algorithms' metrics have decreased with the removing of least important features in comparison with the removing of collinear features, but they are still better than metrics obtained by using model-based approach.

main way to define clipping boundary of the feature importance is direct search among the numbers of the features being clipped.

Algorithm 2 – SelectKBest feature selection

Designations:

trainX – features of the training set; trainY – class labels of the training set; testX – features of the testing set; testY – class labels of the testing set; n – minimum quantity of features; model – a model used feature selection; func – a function name for SelectKBest algorithm to be used to perform evaluation; F – the list of used features; f – the reduced list of used features; acc – accuracy metric value for the model built on the reduced set of features; auc – ROC-AUC metric value for the model built on the reduced set of features;; result – the result of the trained model; resultAcc – accuracy metric value for the trained model; resultAuc – ROC-AUC metric value for the trained model.

Algorithm:

```
function FeatureSelectionWithSelectKBest(trainX, trainY,
testX, testY, n, model, func)
    F = GetFeaturesEstimateFromSelectKBest(trainX,
trainY, func)
    (f, acc, auc) = InitialValues()
    while Length(F) < n do
        model = ModelTrain(model, trainX[F], trainY)
        result = ModelPredict(model, testX)
        (resultAcc, resultAuc) = GetMetrics(testX, result)
        if resultAuc > auc then
            (f, acc, auc) = (F, resultAcc, resultAuc)
        end if
        F = DropLastFeature(F)
    end while
end function
```

There were experiments performed with every exploited function that gave subsequent results for the decreased feature quantity of both accuracy and ROC-AUC metrics and following that the clipping boundary were set at such a level that produces the highest values of the mentioned metrics. If the next feature in importance order have been cut off from the reduced dataset, the metrics' values was starting to decrease.

Numerical values of the feature importance evaluation results using SelectKBest with the chi2 function were in range $0 \dots 10^{14}$. Table 6 represents the best result which was obtained by using the set of 17 from 41 features whose importance value is greater than 10^6 .

Table 6

Classification results of the statistical approach using SelectKBest tool with the chi2 function with the features having importance value lower than 10^6 removal

Algorithm	Decision Tree	Extra Tree	Random Forest	Extra Trees	AdaBoost	Gradient Boosting	Average result
Accuracy		0.896	0.902	0.897	0.896	0.904	0.900
Precision	0.971	0.970	0.979	0.980	0.977	0.979	0.976
Recall		0.874	0.875	0.867	0.867	0.877	0.874
F-score	0.926	0.91	0.924	0.920	0.919	0.925	0.922
ROC-AUC	0.915	0.908	0.918	0.915	0.912	0.919	0.914

As can be seen from Table 6, the average results of algorithms with the set of 17 features obtained using SelectKBest with the chi2 function are higher than similar results obtained by other approaches. The only metric that shows result worse than the one obtained by using FeatureSelector tool is the precision metric. However, high values of all other metrics somewhat compensate the observed insignificant loss (about 0.01%) in the terms of classification quality.

Use of the `f_classif` function gave the range of feature importance numerical values of $1 \dots 10^5$. The selection of the feature set was based on the feature importance numerical value greater than 100 that left the set of 30 features from 41. Table 7 presents the making use of the obtained feature set results for six machine learning algorithms.

Table 7

Classification results of the statistical approach using SelectKBest tool with the `f_classif` function with the features having importance value lower than 100 removal

Algorithm	Decision Tree	Extra Tree	Random Forest	Extra Trees	AdaBoost	Gradient Boosting	Average result
Accuracy	0.895	0.874	0.894	0.888	0.897	0.901	0.892
Precision	0.980	0.974	0.990	0.987	0.980	0.986	0.983
Recall		0.838	0.853	0.847	0.866	0.867	0.855
F-score	0.918	0.901	0.917	0.912	0.919	0.922	0.915
ROC-AUC	0.913	0.895	0.918	0.912	0.915	0.920	0.912

For the `mutual_info_classif` function the feature importance numerical values were in the range of $0 \dots 0.5$. Importance value greater than 0.15 features were used to reduce the initial set from 41 feature to 20. Table 8 demonstrates results of the classification given by the set reduced by this method.

Table 8

Classification results of the statistical approach using SelectKBest tool with the `mutual_info_classif` function with the features having importance value lower than 0.15 removal

Алгоритм	Decision Tree	Extra Tree	Random Forest	Extra Trees	AdaBoost	Gradient Boosting	Средний результат
Accuracy	0.901	0.888	0.901	0.894	0.877	0.893	0.892
Precision	0.971	0.971	0.978	0.980	0.976	0.977	0.976
Recall		0.861	0.873	0.861	0.839	0.863	0.863
F-score	0.923	0.912	0.923	0.917	0.903	0.916	0.916
ROC-AUC	0.913	0.903	0.916	0.912	0.898	0.910	0.909

As can be seen from the tables 7 and 8, the set of 30 attributes selected by using SelectKBest algorithm with the `f_classif` function and the set of 20 features selected with `mutual_info_classif` function shown worse average results in comparison with the set selected using the chi2 function, but they are still better than model-based approach results.

Thus, among the results obtained by algorithm SelectKBest, the best result was achieved using function chi2.

Comparative analysis of classification results with different approaches to the feature selection

Summary results of the classifiers application using reviewed approaches to the feature selection for UNSW-NB15 dataset are shown in Table 9.

Table 9

Classification results for the different feature selections

№	Algorithm	Features count	Features	Accuracy	ROC-AUC
0	—	41	Full feature set	0.740	0.748
Model-based approach					
1	Decision Tree	20	1, 7, 9, 11, 13, 15, 16, 18, 20, 21, 23, 24, 25, 26, 27, 30, 31, 35, 39, 40	0.706	0.707
2	Extra Tree	30	1, 2, 3, 4, 7, 9, 10, 11, 12, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 39, 40, 41	0.830	0.833
3	Random Forest	32	1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 39, 40	0.831	0.818
4	Extra Trees	27	1, 3, 4, 7, 9, 10, 11, 12, 15, 19, 20, 21, 22, 23, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 39, 40, 41	0.840	0.825
5	AdaBoost	11	2, 3, 6, 7, 9, 24, 26, 27, 31, 34, 40	0.762	0.752
6	Gradient Boosting	13	2, 3, 7, 9, 13, 15, 24, 26, 27, 30, 35, 40, 41	0.769	0.759
Statistical approach: FeatureSelector					
7	Removing collinear features	33	1, 2, 3, 4, 5, 6, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 34, 35, 36, 38, 39, 41	0.894	0.914
8	Removing least important features	31	1, 2, 3, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 29, 30, 32, 33, 34, 35, 39, 40	0.890	0.910
9	Removing collinear and least important features	25	1, 2, 3, 5, 6, 9, 11, 12, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 29, 30, 32, 34, 35, 39	0.893	0.913
Statistical approach: SelectKBest					
10	chi2	17	3, 4, 7, 8, 9, 11, 12, 15, 16, 17, 18, 19, 20, 21, 22, 27, 29	0.900	0.914
11	f_classif	30	2, 3, 4, 6, 9, 10, 11, 12, 14, 15, 16, 19, 20, 21, 22, 27, 30, 24, 25, 26, 27, 30, 31, 32, 33, 34, 35, 39, 40, 41	0.892	0.912
12	mutual_info_classif	20	1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 15, 16, 17, 23, 24, 25, 26, 27, 31, 34	0.892	0.909

Following the data from both Table 9 and Figure 2, the best result among the analyzed approaches was demonstrated by using the SelectKBest statistical algorithm with the chi2 characteristic function.

Figure 2. Classifiers comparison results using different approaches to the feature selection

The final reduced feature set obtained using the SelectKBest algorithm with the chi2 characteristic function is presented in Table 10.

Table 10

Reduced feature set for the UNSW-NB15 dataset

№	Feature	№	Feature	№	Feature
1	dtpcb	7	sinpkt	13	swin
2	stcpb	8	sjit	14	dinpkt
3	sload	9	response_body_len	15	djit
4	dload	10	state	16	dwin
5	dbytes	11	service	17	sttl
6	sbytes	12	dmean		

Among the selected features 3 from whose characterize the network flow in general (response_body_len, state and service), 7 from whose characterize the direction from the source to the destination (dtpcb, dload, dbytes, dmean, dinpkt, djit, dwin) and the rest 7 from whose characterize the direction from the destination to the source (stcpb, sload, sbytes, sinpkt, sjit, swin, sttl).

Conclusion

The effect of features' count and their composition within the dataset on the quality of classification was researched and the selection of reduced feature sets for classification tasks was performed using different approaches: statistical and model-based.

On the performed research basis there could be concluded that the noise features that have no significant effect on the result could be harmlessly removed. Furthermore, the informative features quantity reduction of the initial dataset from 41 to 17 makes possible to improve the overall binary classification quality.

Statistical feature selection methods shown vastly better overall results over model-based methods among the approaches reviewed and are recommended to use in similar to attack detection case binary classification issues.

The most effective among the analyzed approaches was the statistical approach using SelectKBest with chi2 characteristic function, which demonstrated accuracy value of 0.9, and ROC-AUC value of 0.914.

References

1. Bermingham, M. L., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., ... & Haley, C. S. (2015). Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific reports*, 5, 10312.
2. Hota, H. S., Shrivastava, A. K., & Singhai, S. K. (2011). An Ensemble Classification Model for Intrusion Detection System with Feature Selection. *International Journal of Decision Science of Information Technology*, 3(1), 13-24.
3. Sheluhin O.I., Simonyan A.G., Vanyushina A.V. (2017). Influence of training sample structure on traffic application efficiency classification using machine-learning methods. *T-Comm*, vol. 11, no. 2, pp. 25-31.
4. Varlamov, A., & Sharapov, R. (2012). Machine learning of visually similar images search. In *CEUR Workshop Proceedings* (Vol. 934, pp. 113-120).
5. Powell, A., Bates, D., Van Wyk, C., & de Abreu, D. (2019). A cross-comparison of feature selection algorithms on multiple cyber security data-sets. In *FAIR* (pp. 196-207).
6. Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pp. 1-6. IEEE.
7. Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(3), 131-156.
8. Sheluhin, O., Erokhin, S., and Vanyushina, A. (2018). *IP-Traffic Classification by Machine Learning Methods*, Moscow: Hotline-Telekom, 2018, 284 p.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СПОСОБОВ ОТБОРА КОЛИЧЕСТВА И СОСТАВА ИНФОРМАТИВНЫХ ПРИЗНАКОВ КОМПЬЮТЕРНЫХ АТАК НА ПРИМЕРЕ БАЗЫ ДАННЫХ UNSW-NB15

Шелухин Олег Иванович, Московский технический университет связи и информатики, Москва, Россия, sheluhin@mail.ru

Иванникова Валентина Павловна, Московский технический университет связи и информатики, Москва, Россия, iv8post@gmail.com

Аннотация

На примере базы данных UNSW-NB15, используемой для обучения моделей машинного обучения, нацеленных на поиск атак в потоках трафика, проведен сравнительный анализ статистических и с использованием моделей способов отбора количества и состава информативных признаков. Отбор признаков одна из важных задач при подготовке данных, так как он позволяет улучшить качество моделей машинного обучения: сократить размеры итоговых моделей, уменьшить время обучения и снизить вероятность переобучения. Исследование проводилось с использованием библиотек языка программирования Python: scikit-learn, включающей в себя модели машинного обучения и функции для отбора признаков и оценки работы моделей, и FeatureSelector, включающей в себя набор функций для статистического анализа признаков. Приведены численные результаты экспериментальных исследований применения как статистических признаков, так и на основе моделей машинного обучения. В результате получен сокращенный набор параметров, позволяющий повысить качество классификации за счёт удаления шумовых признаков, оказывающих незначительное влияние на итоговый результат и сократить количество информативных признаков набора данных с 41 до 17. Показано, что самым эффективным среди анализируемых способов отбора оказался статистический SelectKBest с функцией chi2 обеспечивающий высокую точность классификации - около 90%, по сравнению с 74%, полученными на полном наборе.

Ключевые слова: отбор признаков, машинное обучение, бинарная классификация, сетевые атаки, база данных UNSW-NB15.

Литература

1. Bermingham M. L. et al. Application of high-dimensional feature selection: evaluation for genomic prediction in man // *Scientific reports*. 2015. Т. 5. С. 10312.
2. Hota H. S., Shrivastava A. K., Singhai S. K. An Ensemble Classification Model for Intrusion Detection System with Feature Selection // *International Journal of Decision Science of Information Technology*. 2011. Т. 3. №. 1. С. 13-24.
3. Шелухин О. И., Симоныч А. Г., Ванюшина А. В. Влияние структуры обучающей выборки на эффективность классификации приложений трафика методами машинного обучения // *Т-Comm: Телекоммуникации и Транспорт*. 2017. Т. 11. №. 2.
4. Варламов А.Д., Шарипов Р.В. Поиск визуально подобных изображений на основе машинного обучения // *Электронные библиотеки: перспективные методы и технологии, электронные коллекции: XIV Всероссийская научная конференция "RCDL'2012"*. Переславль-Залесский, 15-18 октября 2012 г.: труды конференции Переславль-Залесский: Изд-во Университет города Переславля, 2012. С. 152-159.
5. Powell A. et al. A cross-comparison of feature selection algorithms on multiple cyber security data-sets // *FAIR*. 2019. С. 196-207.
6. Moustafa N., Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) // *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015. С. 1-6.
7. Dash M., Liu H. Feature selection for classification // *Intelligent data analysis*. 1997. Т. 1. №. 3. С. 131-156.
8. Шелухин О.И., Ерохин С.Д., Ванюшина А.В. Под ред. Шелухина О.И. *Классификация IP-трафика методами машинного обучения*. М.: Горячая линия – Телеком, 2018. 284 с

Информация об авторах:

Шелухин Олег Иванович, доктор технических наук, профессор, заведующий кафедрой информационной безопасности, Московский технический университет связи и информатики, Москва, Россия

Иванникова Валентина Павловна, магистрант, Московский технический университет связи и информатики, Москва, Россия