Journal of Physics: Conference Series

PAPER • OPEN ACCESS

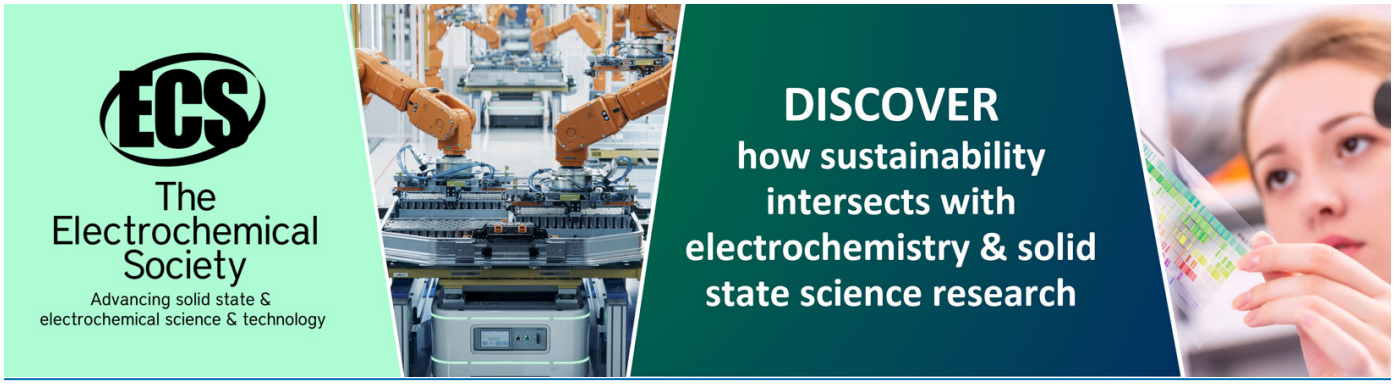
Network traffic analysis based on machine learning methods

To cite this article: A M Vulfin *et al* 2021 *J. Phys.: Conf. Ser.* **2001** 012017

View the article online for updates and enhancements.

You may also like

- Peer Review Statement
- Peer review statement
- Peer review statement



This content was downloaded from IP address 3.134.90.44 on 02/05/2024 at 12:11

Network traffic analysis based on machine learning methods

A M Vulfin, V I Vasilyev, V E Gvozdev, K V Mironov and O E Churkin

Ufa State Aviation Technical University, 12, K. Marks st., Ufa, 450008, Russian Federation

E-mail: vulfin.alexey@gmail.com

Abstract. Comparison of mining algorithms in the problem of detecting malicious network activity based on machine learning models is performed. A structural diagram of a system for analyzing network traffic in an industrial network based on machine learning methods has been developed. On one of the known datasets (CICIDS17), a series of experiments was carried out on preliminary analysis and preprocessing of features, highlighting the most significant features and building final models of classifiers. The f1-measure score for the committee of classifiers on the test sample is 0.967.

1. Introduction

Today, the trend [1] towards combining or even replacing traditional SCADA systems with devices of the Internet of Things (IoT) and the Industrial Internet of Things (IIoT) is becoming more and more obvious. The deep penetration of the IIoT into critical infrastructure and the industrial sector has already led to an increase in the likelihood and number of potential cyberattacks against such structures. Damage from cyberattacks to the energy and utilities industries averages $ 13.2 million annually. The increase in risks is forcing the development of common approaches to ensuring cybersecurity [2].

To solve this kind of tasks, cybersecurity monitoring centers are created that collect, store and analyze traffic [3] both corporate (public servers, client terminals, traffic routing and switching devices) and industrial network (SCADA systems, hubs and hubs of IoT devices). This allows identify patterns of attacks or exploitation of vulnerabilities.

The purpose of the work is to compare the algorithms of intelligent analysis in the task of detecting malicious network activity based on machine learning models.

2. Development of a system for analyzing network traffic in an industrial network based on machine learning methods

The block diagram of the proposed system is shown in figure 1, where 1 – the transfer of the analysis results to the SIEM/SOC system [4]; 2 – network security specialist (DevOps engineer); 3 – data mining specialist; 4 – base of trained ML models for network traffic analysis.

The network session collector collects traffic parameters from agents installed at key points of the network infrastructure: aggregation switches, edge firewall, from access points in the format of the xFlow protocol family (netFlow or OpenFlow) [5]. Modules for preprocessing and extracting features and storing network traffic statistics allow to capture a compact description of network sessions in long-term storage, which allows IDS to conduct retrospective analysis of accumulated data and prompt update of indicators of compromise when interacting with external Threat Intelligence platforms [6]. The module for analyzing and generating features is used to prepare labeled data for building and

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

training machine learning models (ML-models) that are stored in a database (4) for further use in the operational analysis of incoming and internal network traffic. The module for enrichment, testing and verification of ML-models allows additional marking of network traffic by associating certain information security events with the corresponding network sessions. The final decision block for attack detection interacts with a network security specialist and visualizes the results of the analysis of an ensemble of ML-models. The operational interaction of the system is performed with the SOC, which allows to transfer metrics and additional information about the parameters of the current state of the network for subsequent aggregation and analysis. The data mining specialist manages the work of the ensemble of ML-models, performs the tasks of adjusting the parameters of its work and timely updating the bank of models.

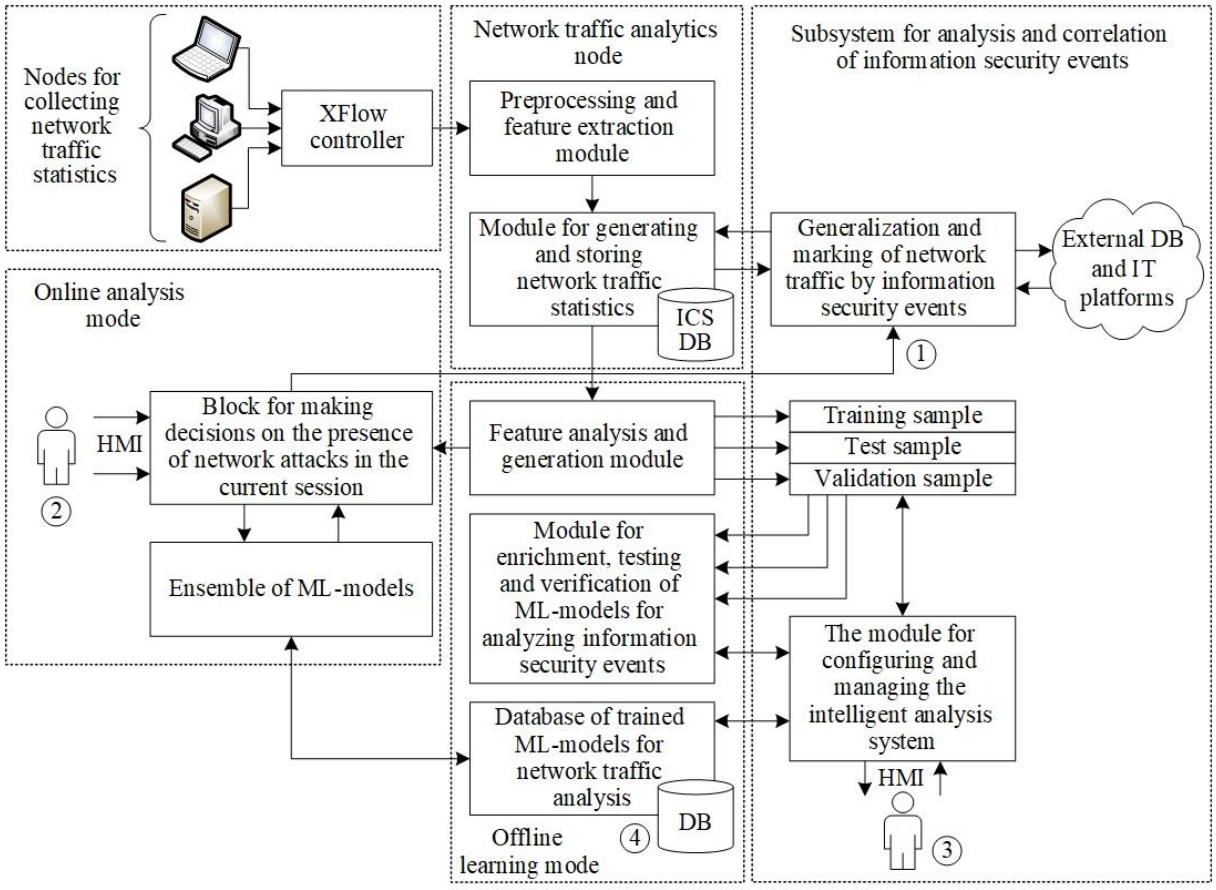


Figure 1. Diagram of a system for analyzing network traffic in an industrial network based on machine learning methods.

In general, the mining algorithm in the problem of detecting anomalies is shown in figure 2.

The CICIDS2017 [7] dataset contains the traffic of the most common network attacks (in PCAP format) [8] and includes the results of network traffic analysis using CICFlowMeter with tagged flows based on time stamp, source and destination IP addresses, source and destination ports, protocols and attacks. The work of 25 users was simulated using the protocols HTTP, HTTPS, FTP, SSH and e-mail. The total number of examples is 3119345, and the number of selected features is 84.

At the preprocessing stage, identical characteristics were deleted, the characteristics in the records containing non-numerical values of NaN and Infinity were filled in correctly. The values of categorical features (“Flow ID”, “Source IP”, “Destination IP” and “Timestamp”) are converted to numerical values using the appropriate encoder (Label Encoder).

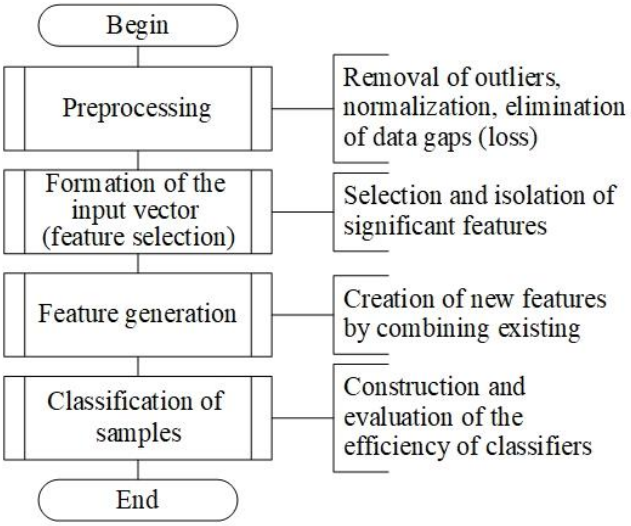


Figure 2. Generalized mining algorithm in the problem of detecting network attacks.

In the original CICIDS2017 set, the number of examples classified as normal is 2273097. At the same time, the number of examples attributed to different classes of attacks is 557646 instances (table 1).

Table 1. Features selected for creating a dataset and describing network sessions.

Label	Attack type	Number of examples in the sample by class	Number of examples after class balancing
BENIGN	Normal traffic	2273097	10500
DoS Hulk	DoS/DDoS	231073	1500
PortScan	Port scan	158930	1500
DDoS	DoS/DDoS	128027	1500
DoS GoldenEye	DoS/DDoS	10293	1500
FTP-Patator	Bruteforce	7938	1500
SSH-Patator	Bruteforce	5897	1500
DoS slowloris	DoS/DDoS	5796	1500
DoS Slowhttptest	DoS/DDoS	5499	1500
Bot	DoS/DDoS	1966	1500
Web Attack – Brute Force	Web attack	1507	1500
Web Attack – XSS	Web attack	652	0
Infiltration	Infiltration	36	0
Web Attack – SQL Injection	Web attack	21	0
Heartbleed	Heartbleed	11	0

Because the dataset is unbalanced, classes with very few examples are removed, such as “Heartbleed”, “Web Attack – Sql Injection”, “Infiltration”, “Web Attack – XSS”, and “Bot”.

From each remaining class of attacks, 1,500 examples are randomly selected, and 10,500 entries are selected from examples of normal operation. The attack class label is encoded with a value between 0 and 9.

Pronounced signature features, according to [9], are removed: “Flow ID”, “Source IP”, “Source Port”, “Destination IP”, “Destination Port”, “Protocol” and “Timestamp”.

This will allow building ML models that are focused on detecting statistical features of network sessions correlated with network attacks, and not with signature parameters that can be changed or tampered with by an attacker, and which traditional network attack detection systems do well.

Feature significance was assessed by a committee (k = 250) of Random forest (RF) using a cross-validation procedure (Validation Score = 0.98).

Next, the significance of the features was assessed using the permutation method. For this, a Logistic Regression model was used. The methods of feature selection used make it possible to reduce their number to 20.

The degree of pairwise correlation of features is estimated and features with a correlation coefficient of more than 0.8 are removed. The final heat map of the pairwise correlation matrix obtained agree with [9].

To reduce the dimension of the feature space and visualize the distribution of examples by classes t-Distributed Stochastic Neighbor Embedding (t-SNE) was applied.

Visualization of classes of attacks and normal operation allows to conclude that there is a data structure with a reduced set of features and the possibility of further constructing a classifier.

The resulting reduced dataset includes the following features: “Packet Length Std”, “Bwd Packet Length Min”, “min_seg_size_forward”, “Flow IAT Mean”, “Total Length of Fwd Packets”, “Flow IAT Max”, “Max Packet Length”, “Fwd Packet Length Max”, “Bwd Packets/s”, “Min Packet Length”.

3. Building classifiers of examples of network sessions

To solve the problem of detecting network attacks based on a vector of features extracted from the description of a network session, it is necessary to create and select the parameters of a ML-model [10]. The classifiers used: XGBClassifier, Random Forest (RF), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Logistic Regression (LR), Classification and Regression Trees (CART), Naive Bayes (NB), AdaBoost, Linear discriminant analysis (LDA), Quadratic Discriminant Analysis (QDA). The procedure for optimizing hyperparameters over a grid is applied.

The dataset was divided into training and test samples - 16800 and 7200 examples, respectively.

Using cross-validation with 5 partitions of the training dataset, the classification procedure for the training and test dataset was carried out by these classifiers with the above parameters (table 3).

In the second experiment, convolutional neural networks with one-dimensional and two-dimensional input layers (CNN1D and CNN2D, respectively) were used.

The dataset was divided into training, test, and validation samples (15120, 7200, and 1680 examples, respectively).

The CNN1D architecture is shown in table 2.

Table 2. The CNN1D architecture.

Layer (type)	Output Shape	Parameters	Filters	Kernel_size	Activation function
Conv1D	(1, 10, 16)	64	16	3	
Batch normalization	(1, 10, 16)	40			
Activation	(1, 10, 16)	0			relu
Conv1D	(1, 8, 32)	1568	32	3	
Batch normalization	(1, 8, 32)	32			
Activation	(1, 8, 32)	0			relu
Flatten	(1, 256)	0			
Dense	(1, 64)	16448			relu
Dropout	(1, 64)	0			
Dense	(1, 10)	650			
Activation	(1, 10)	0			softmax

Training took 50 epochs, the estimate of the f1-measure of the model on the training set was 0.916, on the test set – 0.922.

Next, a classifier based on a deep neural network (DNN) was used (6 dense layers with dropout coeffiecient = 0.2).

Next, a classifier was used based on a convolutional neural network with a two-dimensional input CNN2D feature layer. However, since the dataset did not have a two-dimensional structure, one had to be created. For this, the examples of the set were transformed into graphical primitives with a dimension of 5x2 in shades of gray.

The layered network architecture has 3 conv2D layer, flatten layer with dropout coefficient = 0.2 and 2 dense layers.

The dataset was subdivided into samples similar to the previous model. The training took 100 epochs, as a result of which the f1-measure of the model on the training set was 0.935, on the test set – 0.943.

At the final stage, a committee of classifiers was used, including the Random Forest, the AdaBoost Algorithm and the ExtraTreesClassifier. The latter implements a meta-estimator corresponding to a series of randomized decision trees, or complementary trees, on different subsamples of the dataset, and uses averaging to improve prediction accuracy and control overfitting.

Committee parameters: voting type – "soft" (full voting and weighting of model predictions for each class); the weights are distributed as [1-3].

The dataset was subdivided into samples similar to the previous model. After training, the f1-measure of the model on the training set was 0.981, on the test set – 0.967 (table 4).

4. Results

The classifiers are located in table 4 in descending order of their f1-measure values on the test sample.

Table 3. Results of the first stage of testing classifiers.

Classifier	CV Fit Time, seconds	CV mean F1	Test F1
XGBClassifier	10.35923	0.96816	0.96653
RF	1.15881	0.96637	0.96597
KNN	0.04588	0.94542	0.94639
MLP	44.32636	0.92185	0.92000
SVM	2.96680	0.75893	0.75444
LR	1.86500	0.71601	0.73319
CART	0.03520	0.67369	0.66528
NB	0.00738	0.61369	0.60153
AdaBoost	0.83967	0.53167	0.56597
LDA	0.02663	0.56911	0.55667
QDA	0.01414	0.54542	0.52333

Table 4. Results of the second stage of testing classifiers.

Classifier	Accuracy on the training sample	f1-measure on the training set	Accuracy on the test sample	f1-measure on the test sample
VotingClassifier	0.980	0.981	0.967	0.967
RF	0.976	0.977	0.967	0.967
KNN	0.982	0.982	0.954	0.955
CNN2D	0.936	0.935	0.944	0.943
CNN1D	0.917	0.916	0.924	0.922
DNN	0.873	0.872	0.879	0.878

5. Discussion

From the pivot tables presented earlier, several conclusions can be drawn:

- Some of the best results are shown by the classifiers XGBClassifier, Random Forest, and k-Nearest Neighbors. The estimates of accuracy and f1-measure in both cases differ insignificantly.
- The Quadratic Discriminant Analysis classifier showed the worst result compared to the others used in the first table.
- At the second stage of the experiment, the VotingClassifier (committee) and the Random forest showed the best results. Considering that the first one of the voting classifiers also included the Random Forest, such results are quite understandable.
- In absolute terms, the VotingClassifier showed the best efficiency, reaching a record f1-measure of 0.981 on the training set and 0.967 on the test set.

6. Conclusion

A structural diagram of a system for monitoring, collecting and correlating information security events in an industrial network has been developed and described.

Algorithms for intelligent analysis of network traffic parameters in the task of detecting malicious network activity have been developed. The general scheme of the algorithm is presented. At the end of the experiment and the cumulative analysis of all the results, the most effective was the committee of classifiers based on a random forest model, randomized decision trees and Adaboost, which has the highest f1-measure value on the training set of all the others, equal to 0.981, and on the test set 0.967.

Acknowledgments

The reported study was funded by RFBR according to the research project No. 19-07-00972 A

References

[1] Moore B 2018 Gartner's top 10 IoT tech trends. *IT Brief*
[2] Cybersecurity threatscape: Q4 2020. *Positive Technologies*
[3] Cecil A 2006 A summary of network traffic monitoring and analysis techniques. *Computer Systems Analysis* 4-7
[4] Paganini P 2016 What is a SOC (Security Operations Center)? *Security Affairs*
[5] Trammell B, Wagner A and Claise B 2013 Flow aggregation for the ip flow information export (IPFIX) protocol. *Internet Requests for Comments, RFC Editor, RFC 7015*
[6] What is Cyber Threat Intelligence? *Cisco Systems, Inc.*
[7] Kostas K 2018 Anomaly Detection in Networks Using Machine Learning. *Research Proposal* **23** 343
[8] Gharib A et al. 2016 An evaluation framework for intrusion detection dataset. *International Conference on Information Science and Security (ICISS). IEEE* 1-6
[9] Goryunov M N et al. 2020 Synthesis of a Machine Learning Model for Detecting Computer Attacks Based on the CICIDS2017 Database. *Trudy ISP RAN* **32(5)** 81-94
[10] Priya V et al. 2021 Robust attack detection approach for IIoT using ensemble classifier. *Computers, Materials & Continua* **66(3)** 1-14