UDC 681.5.004.4

Bohdan Smishchenko, Artem Volokyta.

CREATING PROGRAM CLASSIFYING REQUESTS

In this article is described prototype of application which solves problem of classification requests as secure and dangerous. Neural networks are demonstrated as tool for automation of request processing. Implementation of algorithm is going to be taken from tensorflow and NSL-KDD library as source of training data for neural network.

keyword: neural networks, under-sampling, IPS

Fig.: 5, Tabl.: 1, Bibl.: 13

Actuality. This article demonstrates prototype of application that classifies requests on safe and dangerous. Shows using of neural networks for automatization request processing. To implement neural network will use library tensorflow and as source of data for network training– NSL-KDD

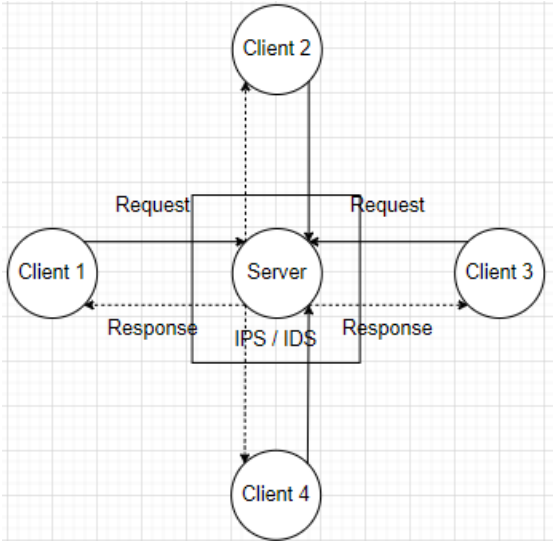

Fig. 1. Intrusion detection system in Client-server architecture)

As we can see on figure 1, Intrusion Detection System (IDS) / Intrusion Protection System (IPS) controls all network activity that reach server and provided by it.

Usage client-side architecture is widely spread in enterprise applications but like any other architecture it has it`s own vulnerabilities.

From client side, danger for server might consist in different software that gains control over client`s requests of can even create own request over designated sites without any notification. Also, client-side vulnerability can occur as a result of exploitation of network socket and establish unauthorized connection.

From server side, vulnerabilities are far more dangerous for application and user`s data, therefore damage after losing control can be far severer. Most common types of attack are [3]:

1. SQL injections
2. Broken authentication
3. Vulnerabilities caused by invalidated forwards and redirects
4. Usage of outdated encrypting algorithms and vulnerabilities in up to date.

For example, fraud can intercept client`s request if it is not properly encrypted or if fraud get access to client`s device and will get access to his profile or modify output from server.

Table 1.

Difference between IDS and IPS

	IDS	IPS
Analyzes all data inside network	+	+
Notifies administrator	+	+
Can block dangerous requests	+	+
Can modify firewall	-	+
Restore network after attack	-	+

According to table 1, difference between systems consists in ability to not only to alert administrator, but in ability to react on attack.

Actual scientific research and issues analysis. In order to the inventions of new methods and algorithms in machine learning that increased accuracy and reduced hardware requirements, topic of cybersecurity has rapidly developed in recent years.

Nowadays defense of server -side protection system is divided into two groups – intrusion prevention system and intrusion detection system.

Examples of Intrusion Prevention Systems [8]:

1. Splunk
2. Sagan
3. OSSEC
4. Open WIPS-NG
5. Fail2Ban
6. Zeek

In this list shown enterprise solution of IPS.

Uninvestigated parts of general matters defining. Despite great number of articles dedicated to using neural networks to classify various attacks on your equipment from computer to network connection. For example, usage of artificial intelligence in antiviruses is widely used in antiviruses but still such technology is still

vulnerable for new types of attacks. Therefore, this work is focused on prototyping of application processing requests on your device.

The research objective Purpose of this article is to create application that classifies requests from external IP address. Also researched approaches to increase accuracy in request classifying and learning possibility of making self-modifying system. As a solution, this article is focused on process of creating model able to classify requests and process of adjusting accuracy.

The statement of base material. As input in our task we have vector of elements $\{X_1, X_2, X_3, X_4, .., X_i, .., X_m\}$ each of them consists of 41 parameter of different types such as numbers and texts. Each vector X_i matches to value Y_i therefore alongside vector of elements we have vector of values $\{Y_1, Y_2, Y_3, Y_4, .., Y_i, .., Y_m\}$, where i – number of request inside file, and m – number of elements inside file. Task of this article is to build model that matches vector X_i to value Y_i. Another problem of this dataset consists in having not equal numbers of examples for each category that results in lower accuracy of classification or in some algorithms reduces accuracy to zero.

0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,1,0,0,150,25,0.17,0.03,0.17,0,0,0,0.05,0,normal,20
0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,0,0,0,0,0.08,0.15,0,255,1,0,0.6,0.88,0,0,0,0,0,normal,15
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,1,1,0,0,0.05,0.07,0,255,26,0.1,0.05,0,0,1,1,0,0,neptune,19
0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5,0.2,0.2,0,0,1,0,0,30,255,1,0,0.03,0.04,0.03,0.01,0,0.01,normal,21
0,tcp,http,SF,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,30,32,0,0,0,0,1,0,0.09,255,255,1,0,0,0,0,0,0,normal,21
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,19,0,0,1,1,0.16,0.06,0,255,19,0.07,0.07,0,0,0,0,1,1,neptune,21
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,166,9,1,1,0,0,0.05,0.06,0,255,9,0.04,0.05,0,0,1,1,0,0,neptune,21
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,117,16,1,1,0,0,0.14,0.06,0,255,15,0.06,0.07,0,0,1,1,0,0,neptune,21
0,tcp,remote_job,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,270,23,1,1,0,0,0.09,0.05,0,255,23,0.09,0.05,0,0,1,1,0,0,neptune,21
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,133,8,1,1,0,0,0.06,0.06,0,255,13,0.05,0.06,0,0,1,1,0,0,neptune,21
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,205,12,0,0,1,1,0.06,0.06,0,255,12,0.05,0.07,0,0,0,0,1,1,neptune,21

Fig. 2. Examples of transformed input [4]

On the Figure above we can see examples of requests, made by users.

General model structure. The structure of model was created to predict value most accurately, moreover such structure makes possible modifying neural network.

As a scenario – user makes request or bunch of requests on your server, system interprets this request into vector X then model makes assumption Y and then this pair X and Y is offered to be analyzed by administrator. Fig 1 illustrates this idea.

Outside users – client of service, that makes requests and can be restricted by our System.

Input request – group of requests made by users, if are secure will be resent to defendable system

Request converter – converts input requests into suitable vector for model.

Model – given transformed request and list of data to be trained on, makes prediction, which can be used either by administrator to improve accuracy of model or prediction can be reason to ban user.

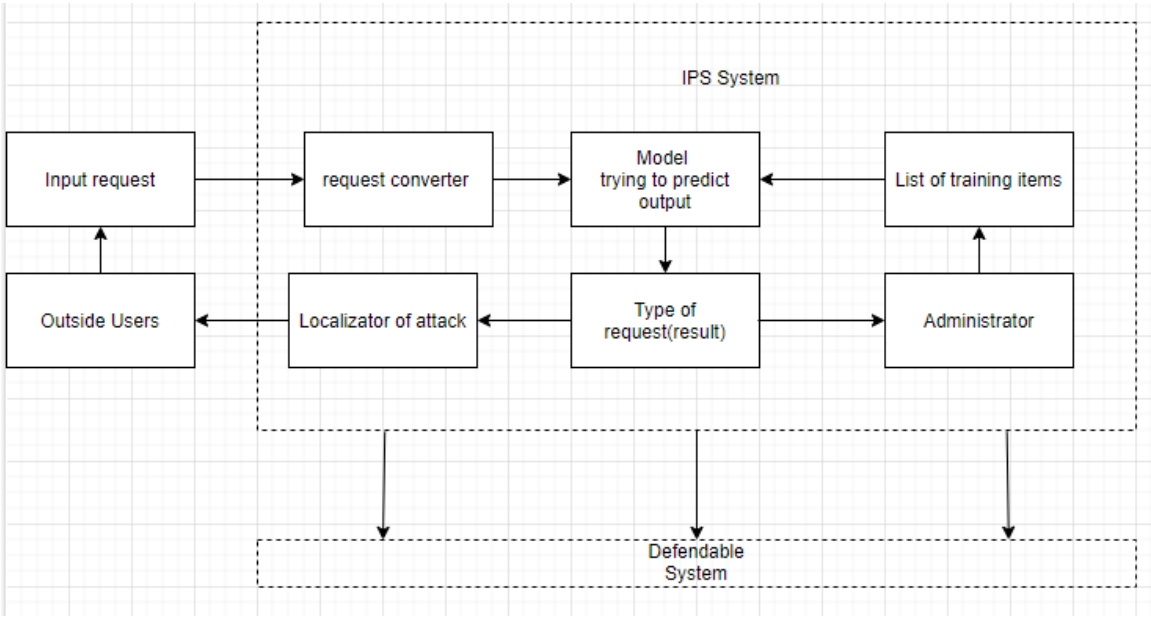

Fig. 3. General model structure

Type of request – main label for request that alongside with transformed request can be used by administrator to improve model and system itself. Also type of request transferred to localization of attack whether request is dangerous.

Administrator – adds new items to list of training items also analyses output of system to manually find new types of attack and improve system`s accuracy.

Localization attack block– blocks dangerous users and considers anomalies of user`s behavior.

As we can see from the image, new attacks are tracked by administrator, so making fully autonomous system is in plans for future.

Possible ways to improve model One of possible solutions of improving accuracy might be generating new examples or reduce amount of the most popular examples. Generating new values based on existing values but another way to get new samples might manual evaluating of prepared examples got during using of application. To create new values it is possible to use library imblearn and function SMOTE(Synthetic Minority Over sample Technique)[6] also it is possible to generate average attack of this type by getting average all of existing values same get maximum and minimum attack. Another possible way to make service more efficient is getting into account not only requests themselves but to include information about sender such as usual time of activity or region where user usually signs in. This technique allows service prevent accounts from stealing and be aware of anomalies appeared on client side.

Experiments Structure of neural network highly affects accuracy so one of the task during making machine learning application is to find out correct structure of model.

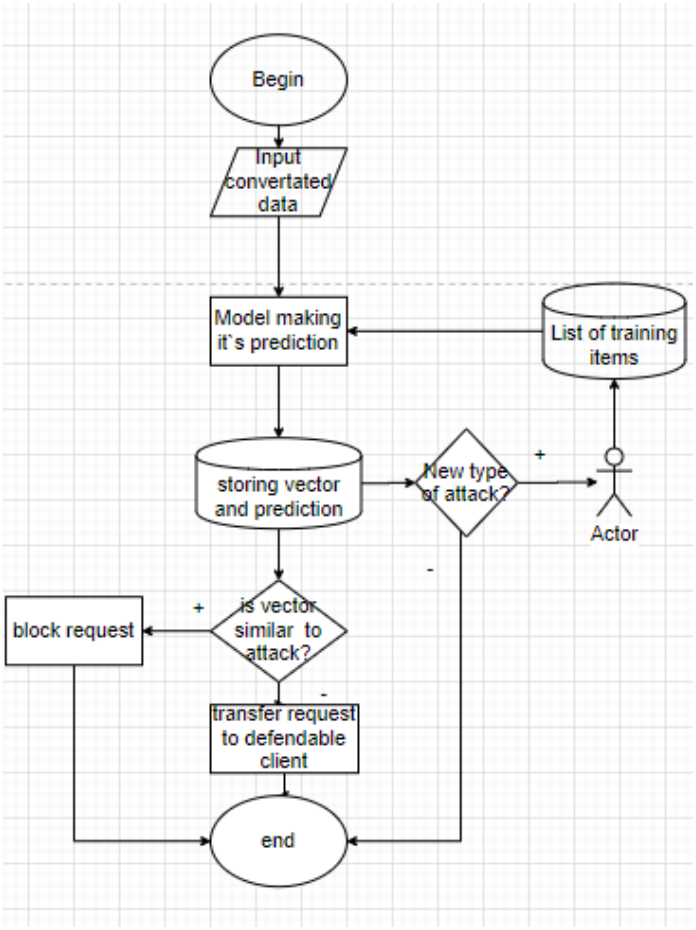

Fig. 4. Algorithm of vector procession

For example, 128 – 128 - 1 – Default scheme, provides accuracy approximately 96%. Accuracy, in future, it is possible to reach better accuracy by changing form of neural network. As we can see, accuracy is almost 100% and further training is going to reduce model`s accuracy and will require more time, but possible way to improve lays in changing form of neural network.

```
1260/1260 [==============================] - 79s 63ms/step - loss: 0.6201 -
accuracy: 0.9532 - val_loss: 0.4803 - val_accuracy: 0.9616
Epoch 2/3
1260/1260 [==============================] - 122s 97ms/step - loss: 0.4979 -
accuracy: 0.9644 - val_loss: 0.4931 - val_accuracy: 0.9642
Epoch 3/3
1260/1260 [==============================] - 166s 132ms/step - loss: 0.4900 -
accuracy: 0.9657 - val_loss: 0.5445 - val_accuracy: 0.9615
394/394 [==============================] - 39s 99ms/step - loss: 0.5335 -
accuracy: 0.9622
Accuracy 0.9621750116348267
```

Fig. 5. Process of model training

Conclusion On example of this task we have demonstrated ways to modify dataset in order to improve accuracy of neural network. In theory described basics of

creating neural networks and provide possible disadvantages from classification algorithm. In future works, we should focus on integration of this system with others in order to customize it under customer`s needs. Also, we should improve fault tolerance of this IPS.

References

1. Matthew K. Thoughtful Machine Learning with python. A test-driven approach. – O`Reilly 2017. – 216 c.

2. Clarence Chio, David Freeman Machine Learning and Security Protecting Systems with data and algorithms. – O`Reilly 2018. – 385

3. Security Architecture Vulnerabilities URL: https://resources. infosecinstitute.com/category/certifications-training/cissp/domains/security-engineering/security-architecture-vulnerabilities (request date 11.05.2020).

4. Rowland, Craig H. "Intrusion detection system." U.S. Patent No. 6,405,318. 11 Jun. 2002.

5. Classify structured data URL: https://www.tensorflow.org/tutorials/ structured_data/feature_columns?hl=ru (request date 11.05.2020).

6. Description of library generating new items URL: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTE.html (request date 7.05.2020)

7. Files as source to neural network URL: https://web.archive.org/web/ 20150205070216/http://nsl.cs.unb.ca/NSL-KDD (request date 6.04.2020)

8. Examples of IPSs URL: https://www.comparitech.com/net-admin/ips-tools-software/(request 11.05.2020)

AUTHORS

Bohdan Smishchenko – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: bogdan.smishenko@gmail.com

Artem Volokyta (supervisor) – associate professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”.

E-mail: artem.volokita@kpi.ua