Anomaly-based Network Intrusion Detection Methods

Pavel NEVLUD, Miroslav BURES, Lukas KAPICAK, Jaroslav ZDRALEK

Department of Telecommunications, Faculty of Electrical Engineering and Computer Science,
VSB–Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic

pavel.nevlud@vsb.cz, miroslav.bures@vsb.cz, lukas.kapicak@vsb.cz, jaroslav.zdralek@vsb.cz

Abstract. *The article deals with detection of network anomalies. Network anomalies include everything that is quite different from the normal operation. For detection of anomalies were used machine learning systems. Machine learning can be considered as a support or a limited type of artificial intelligence. A machine learning system usually starts with some knowledge and a corresponding knowledge organization so that it can interpret, analyse, and test the knowledge acquired. There are several machine learning techniques available. We tested Decision tree learning and Bayesian networks. The open source data-mining framework WEKA was the tool we used for testing the classify, cluster, association algorithms and for visualization of our results. The WEKA is a collection of machine learning algorithms for data mining tasks.*

Keywords

Anomaly-based detection, attack, bayesian networks, WEKA.

1. Introduction

Nowadays, computer network is a frequent target of attacks in order to obtain confidential data, or unavailability of network services. To detect and prevent these attacks, there are a large number of software or hardware solutions such as IDS (Intrusion Detection Systems), firewalls and monitoring systems.

These attacks increased normal network traffic that appears as something undesirable, what would not occur in the network. Such deviations from normal operation are called as network anomalies. Between network anomalies include everything that is quite different from the normal operation of the network [1].

Anomalies are values in a statistical sample which does not fit a pattern that describes most other data points. Figure 1 illustrates anomalies in a simple 2-dimensional data set. The data has one normal regions, since most observations lie in this region. Three points that are sufficiently far away from the regions are anomalies. One of these points is border point that can be detected as anomaly.

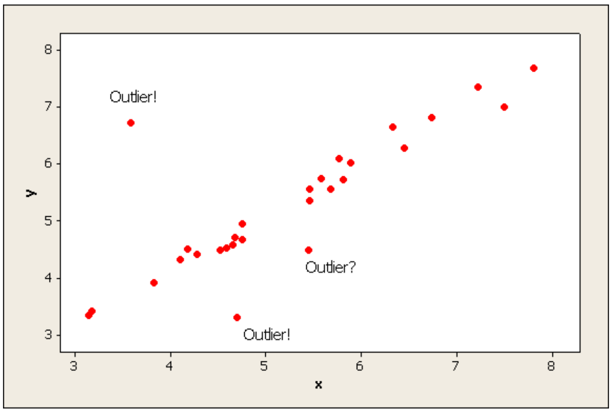


Fig. 1: A simple example of anomalies.

2. Detection of Network Anomalies

Network anomalies can be detected in several ways. Each method has its advantages and disadvantages, but in practice there are three commonly used methods. Them together they can develop systems such as IDS software.

2.1. Comparing Signatures

The principle of this method is the comparison of network data with a database of signatures. Signature database contains patterns of data anomalies. Data anomaly pattern is actually a description of a typical data sequence that characterizes the anomaly. The principle can be seen in Figure 2. It used the same principle as in the anti-virus programs.

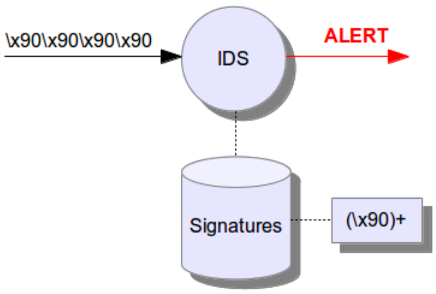


Fig. 2: Comparing signatures.

The effectiveness of anomaly detection using signature recognition is highly dependent on the quality of the database of signatures. The big disadvantage is almost no detection of new types of attacks called Zero day attack, because it is not in the database signature pattern for this type of anomaly [4].

2.2. Stateful Protocol Analysis

Stateful protocol analysis assumes that each protocol used for network communication is specified, such as RFC. Thanks to precise specifications, all connections using protocols defined state. Each event must occur at the right moment, the state. This makes it possible to describe the protocol as a state machine. Figure 3 illustrates an example of stateful machine.

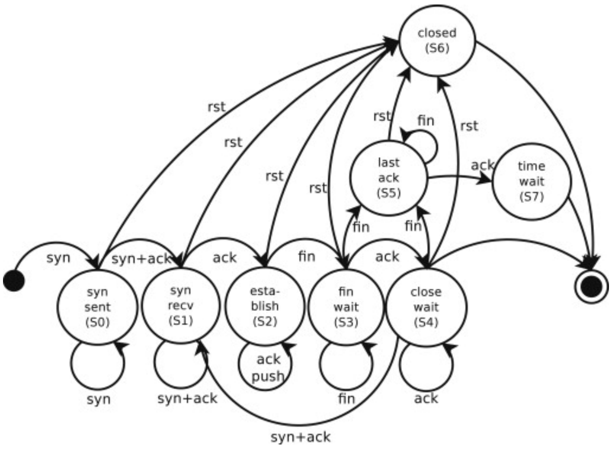


Fig. 3: Stateful protocol analysis.

The advantage of this method is less frequent updates. The stateful analysis needs update only after the change of protocol or the installation of a new one [3].

2.3. Behavioral Analysis

The method of behavioral analysis is based on the assumption that the emergence of anomalies can be detected by the deviation from the normal or expected network behavior. Model of normally or anticipated behavior of the network is created based on network monitoring and collecting reference information.

The reference information is compiled model normal behavior and network traffic is subsequently compared with this model. Any deviation from such a learned model is automatically considered an anomaly. The principle can be seen in Fig. 4. For behavioral analysis and create network's model can be used MLS (Machine Learning Systems).

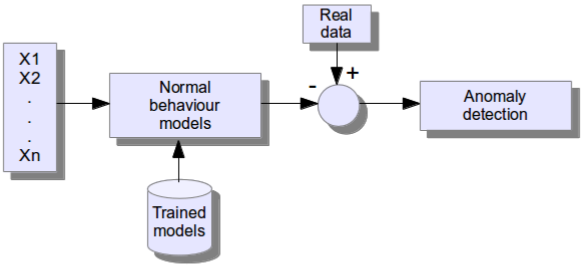


Fig. 4: Behavioral analysis.

The disadvantage of this method is precisely the fine detection. Any deviation from the normal model is detected even though it is not an attack or threat. It is due to the fact that the creation of the model can not capture all types of network traffic and user activity on the network. This model is created to some extent distorted.

On the other hand, behavioral analysis provides an advantage in terms of detection of completely new types of threats, for example, by comparing detection signatures did not react at all.

3. Machine Learning Systems

If we want to be able to solve the computer problem, some intelligence is needed. Machine learning can be considered as a support or a limited type of artificial intelligence. Algorithms MLS can move on with the development of computers. This means that computers are no longer just a database comparing sets of data.

A machine learning system usually starts with some knowledge and a corresponding knowledge organization so that it can interpret, analyze, and test the knowledge acquired. The principle can be seen in Fig. 5.

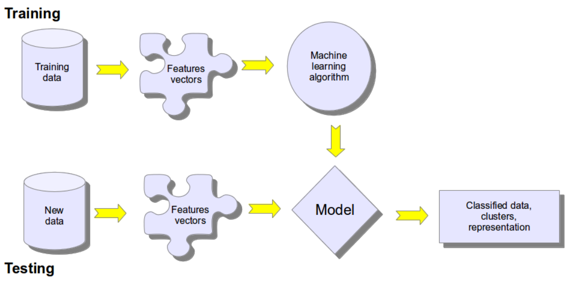


Fig. 5: Principle of machine learning system.

Training is the process of making the system able to learn. It may consist of randomly selected examples that include a variety of facts and details including irrelevant data. The learning techniques can be characterized as a search through a space of possible hypotheses or solutions. Background knowledge can be used to make learning more efficient by reducing the search space.

The success of machine learning system also depends on the algorithms. These algorithms control the search to find and build the knowledge structures. The algorithms should extract useful information from training examples. There are several machine learning techniques available [2].

Among the best-known machine learning algorithms include:

- Decision tree learning.
- Artificial neural networks.
- Genetic programming.
- Clustering.
- Bayesian networks.
- Representation learning.

4. Decision Tree Learning

Decision tree learning is 'a method for approximating discrete valued functions that is robust to noisy data and capable of learning disjunctive expressions' according to [5].

Ross Quinlan has produced several working decision tree induction methods that have been implemented in his programs, ID3, C4.5 and C5. Decision tree induction takes a set of known data and induces a decision tree from that data. The tree can then be used as a rule set for predicting the outcome from known attributes. The initial data set from which the tree is induced is known as the training set. The decision tree takes the top-down form. At the top is the first attribute and its values, from this next branch leads to either an attribute or an outcome. Every possible leaf of the tree eventually leads to an outcome.

4.1. Decision Trees – C4.5

C4.5 is an algorithm developed by Ross Quinlan that generates Decision Trees (DT), which can be used for classification problems. It improves (extends) the ID3 algorithm by dealing with both continuous and discrete attributes, missing values and pruning trees after construction. Its commercial successor is C5.0/See5, a lot faster that C4.5, more memory efficient and used for building smaller decision trees. J48 is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool.

Algorithm 1 C4.5(D)

```
Input: an attribute-valued dataset $D$
Tree = {}
if $D$ is "pure" OR other stopping criteria met then
    terminate
end if
for all atribute a $\in$ $D$ do
    Compute information-theoretic criteria if we split $a$
end for
$a_{best}$ = Best attribute according to above computed criteria
Tree = Create a decision node that tests $a_{best}$ in the root
$D_v$ = Induced sub-datasets from $D$ based on $a_{best}$
for all $D_v$ do
    Tree$_v$ = C4.5($D_v$)
    Attache Tree$_v$ to the corresponding branch of Tree
end for
return Tree
```

The generic description of how C4.5 works is shown in Algorithm 1. A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). Decision tree algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one. The entropy of class random variable that takes on c values with probabilities $p_1, p_2, \ldots, p_c$ is given by:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i. \tag{1}$$

Figure 6 shows the form of the entropy function relative to a binary classification.

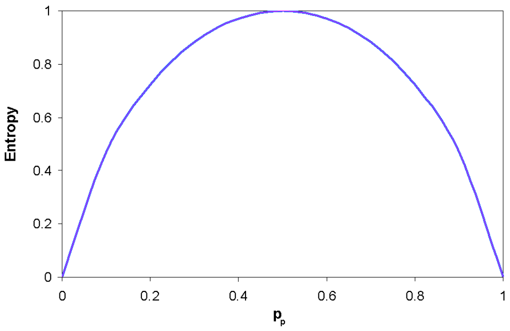
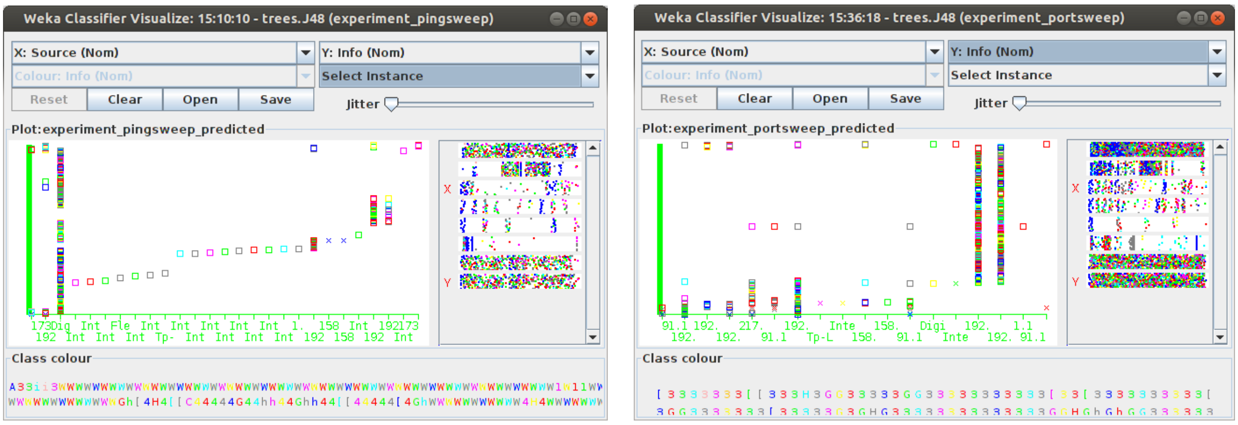


Fig. 6: Entropy function.



Weka Classifier Visualize: 15:10:10 - trees.J48 (experiment_pingsweep)

X: Source (Nom)

Y: Info (Nom)

Colour: Info (Nom)

Select Instance

Plot:experiment_pingsweep_predicted

Weka Classifier Visualize: 15:36:18 - trees.J48 (experiment_portsweep)

X: Source (Nom)

Y: Info (Nom)

Colour: Info (Nom)

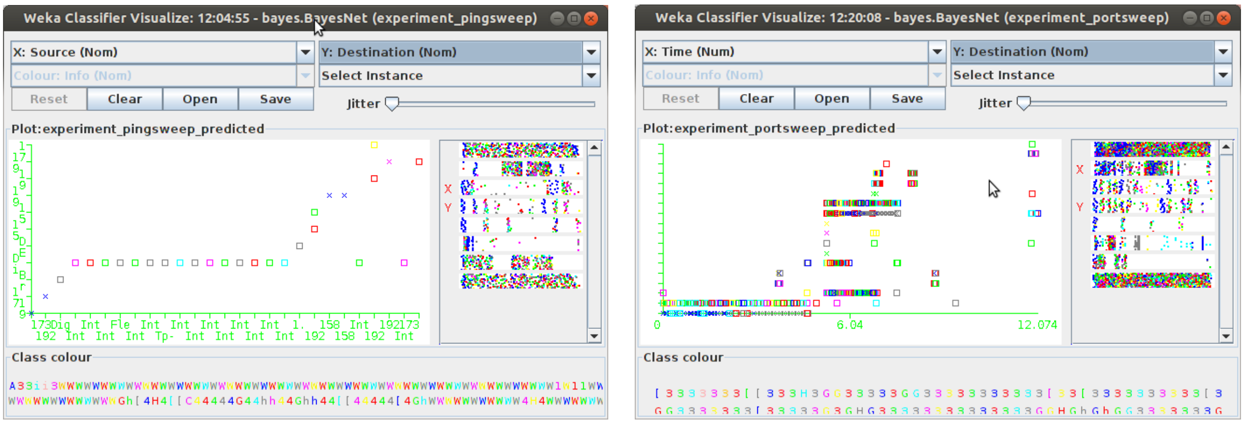
Select Instance

Plot:experiment_portsweep_predicted

(a) pingsweep

(b) portsweep

Fig. 7: J48 classifier results.



Weka Classifier Visualize: 12:04:55 - bayes.BayesNet (experiment_pingsweep)

X: Source (Nom)

Y: Destination (Nom)

Colour: Info (Nom)

Select Instance

Plot:experiment_pingsweep_predicted

Weka Classifier Visualize: 12:20:08 - bayes.BayesNet (experiment_portsweep)

X: Time (Num)

Y: Destination (Nom)

Colour: Info (Nom)

Select Instance

Plot:experiment_portsweep_predicted

(a) pingsweep

(b) portsweep

Fig. 8: BayesNet classifier results.

The estimation criterion in the decision tree algorithm is the selection of an attribute to test at each decision node in the tree. The goal is to select the attribute that is most useful for classifying examples. A good quantitative measure of the worth of an attribute is a statistical property called information gain that measures how well a given attribute separates the training examples according to their target classification. This measure is used to select among the candidate attributes at each step while growing the tree. The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e. the most homogeneous branches).

$$Gain(S,A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v), \quad (2)$$

where $Values(A)$ is the set of all possible values for attribute A, and S_v is the subset of S for which attribute A has value v (i.e. $S_v = \{s \hat{I}\, S | A(s) = v\}$).

The first term in the equation for information gain is just the entropy of the original collection S and the second term is the expected value of the entropy after S is partitioned using attribute A. The expected entropy described by this second term is simply the sum of the entropies of each subset S_v, weighted by the fraction of examples $|S_v|/|S|$ that belong to S_v. $Gain(S,A)$ is therefore the expected reduction in entropy caused by knowing the value of attribute A. Put another way, $Gain(S,A)$ is the information provided about the target attribute value, given the value of some other attribute A. The value of $Gain(S,A)$ is the number of bits saved when encoding the target value of an arbitrary member of S, by knowing the value of attribute A.

5. Bayesian Networks

Bayesian networks are graphical representation of the relationship between variables. Graphical representation of Bayesian networks are directed acyclic graphs with nodes and edges. Nodes represent variables, parameters or hypotheses and edges represent conditional dependencies.

5.1. Algorithm of Naive Bayesian

The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. Bayes theorem provides a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$. Naive Bayes classifier assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. This assumption is called class conditional independence.

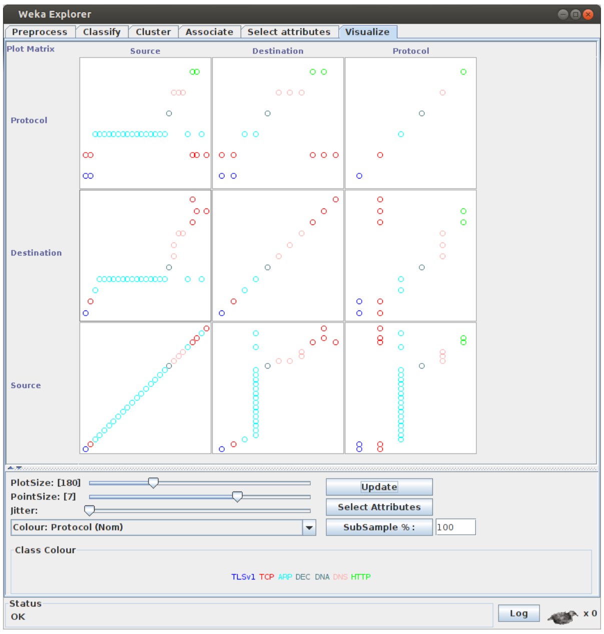
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}, \tag{3}$$

$$P(c|x) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c), \tag{4}$$

where $P(c|x)$ is the posterior probability of class (target) given predictor (attribute), $P(c)$ is the prior probability of class, $P(x|c)$ is the likelihood which is the probability of predictor given class and $P(x)$ is the prior probability of predictor.

6. Experimental Results

For data mining platform was chosen open source project WEKA [6]. WEKA is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code.



PlotSize: [180]
PointSize: [7]
Jitter:
Colour: Protocol (Nom)
SubSample % : 100
Status
OK

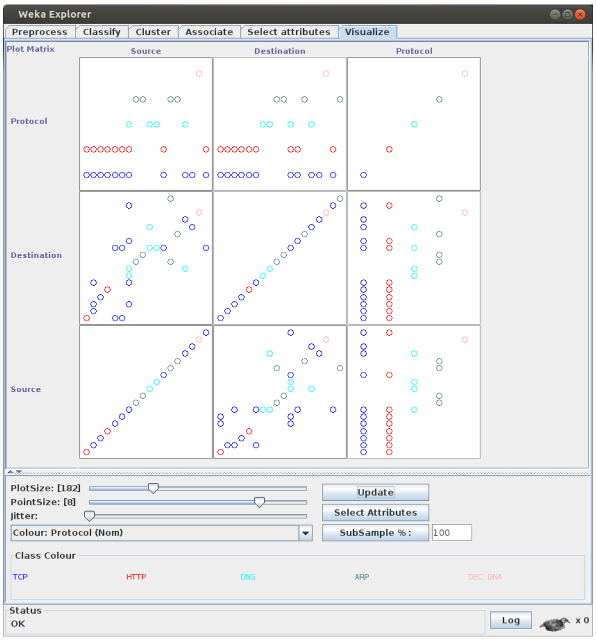
Fig. 9: Visualize results of pingsweep.

WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

Tested activities were captured by the network traffic collector and saved as pcap files. These pcap files are binary files and can't be read directly into most data mining applications.

These pcap files were converted into csv files with scripts using tshark [7]. WEKA accepts *.csv files, *.arff files or a connection to a database. For this research were used converted csv files that was opened in WEKA.

As part of the preprocessing step, information data were injected into the data which were useful for training of the data mining algorithm. Additionally insignificant data were eliminated if it were not serving the overall process. To begin running this data through the algorithms it was opened in WEKA explorer. For the purpose of these initial runs we selected all of the attributes.



PlotSize: [182]
PointSize: [8]
Jitter:
Colour: Protocol (Nom)
SubSample % : 100
Status
OK

Fig. 10: Visualize results of portsweep.

Figure 7(a) shows the results in WEKA of a J48 Classifier training run on data acquired during ping sweep of the target network. The results described here were derived from single nmap target scan, where an attack computer scanned a victim network, probing for active IP addresses.

Also Fig. 7(b) shows the results in WEKA of a J48 Classifier training run on data acquired during port sweep of the target computer. The results described here were derived from single nmap target scan, where an attack computer scanned a victim computer, probing for open ports.

There were chosen only 3 attributes to visualize experimental results. These attributes were IP source address, IP destination address and Protocol. Figure 9 and Fig. 10 show relation between these attributes.

Figure 8(a) shows the results in WEKA of a NaiveBayes Classifier training run on data acquired during ping sweep of the target network. Also Fig. 8(b) shows the results in WEKA of a NaiveBayes Classifier training run on data acquired during port sweep of the target computer.

7. Conclusion and Future Work

In this paper, we have presented detection of network anomalies by using machine learning systems. Machine learning system usually starts with some knowledge and during the rounds can improve its knowledge.

There were tested some attacks in regular network traffic. As the first attack was used ping sweep to sub network target to get information about active IP addresses. The port sweep was used as second attack to scanning open ports at the target victim computer.

First, we captured network traffic by the network collector and saved data as pcap format file. Next, we converted collected data from pcap file into csv format file. We used some scripts by means of tshark to convert data from pcap to csv file format. Next converted csv data was inserted in the WEKA software to use classification of data. Finally classified data was visualize by WEKA software.

Future work expects to use more attributes that will be get from pcap files. We also assume the use of other classification methods and other data mining algorithms.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

References

[1] FOWLER, Ch. A. and R. J. HAMMELL II. Building Baseline Preprocessed Common Data Sets for Multiple Follow-on Data Mining Algorithms. In: *Proceedings of the Conference on Information Systems Applied Research 2012*. New Orleans: EDSIG, 2012, pp. 1–17. ISSN 2167-1508.

[2] FARRAPOSO, F., P OWEZARSKI and E. MONTEIRO. NADA–Network Anomaly Detection Algorithm. In: *18th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, DSOM 2007*. San Jose: Springer Verlag, 2007, vol. 4785, pp 191—194, ISBN 978-3-540-75694-1.

[3] DAS, K. Protocol Anomaly Detection for Network-based Intrusion Detection. *The SANS Institute* [online]. 2002. Available at: `http://www.sans.org/reading_room/whitepapers/detection/protocol_anomaly_detection_for_networkbased_intrusion_detection_349?show=349.php&cat=detection`.

[4] RICHARD, M. Intrusion Detection FAQ: Are there limitations of Intrusion Signatures?. *The SANS Institute* [online]. 2001. Available at: `http://www.sans.org/resources/idfaq/limitations.php`.

[5] MITCHELL, Tom M. *Machine learning*. Boston: McGraw-Hill, 1997. ISBN 00-704-2807-7.

[6] WEKA 3. *Data Mining Software in Java* [online]. 2013. Available at: `http://www.cs.waikato.ac.nz/ml/weka/`

[7] TShark [online]. 2013. Available at: `http://www.wireshark.org/docs/man-pages/tshark.html`

About Authors

Pavel NEVLUD received his M.Sc. degree in telecommunication engineering from VSB–Technical University of Ostrava, Czech Republic in 1995. Since this year he has been holding position as an assistant professor at the Department of Telecommunications, VSB–Technical University of Ostrava. The topics of his research interests are communication technologies, networking and security.

Miroslav BURES received his M.Sc. degree in telecommunications from VSB–Technical University of Ostrava, Czech Republic in 2011. Since 2011 has been studying Ph.D. degree at the same university. His research is focused on networking, analysis of network's data and security.

Lukas KAPICAK received his M.Sc. degree in telecommunications from VSB–Technical University of Ostrava, Czech Republic in 2007. Since 2007 has been studying Ph.D. degree at the same university. His research is focused on wireless transmission and data flow analysis, simulation and optimization.

Jaroslav ZDRALEK holds position as an associate professor with Department of Telecommunications, VSB–Technical University of Ostrava, Czech Republic. He received his M.Sc. degree in Computer Science from Slovak Technical University of Bratislava, Slovakia in 1977. He received his Ph.D. degree from VSB–Technical University of Ostrava in 2002, dissertation thesis "Diagnostic system without dismantling of locomotive controller". His research is focused on fault tolerant system and communication technologies.