Rafał Kozik, Ph.D. Eng.
Faculty of Telecommunication, Information Technology and Electrical Engineering
UTP University of Sciences and Technology in Bydgoszcz

Summary of scientific achievements

1. Name and surname: Rafał Kozik

2. Diplomas and degrees.

Year	Title	Details
2013	Ph.D. Eng.	Faculty of Telecommunications, Information Technology and Electrical Engineering UTP University of Sciences and Technology in Bydgoszcz Thesis: „Complex algorithms of computer vision for blind people support” Supervisor: prof. dr. hab. Ryszard S. Choraś
2008	M.Sc.	Faculty of Telecommunications, Computer Science and Electrical Engineering UTP University of Sciences and Technology in Bydgoszcz

3. Information about employment

Year	Position	Place
2013-obecnie	Assistant professor	Faculty of Telecommunications, Computer Science and Electrical Engineering UTP University of Sciences and Technology in Bydgoszcz
2008-2013	Research Assistant	Faculty of Telecommunications, Computer Science and Electrical Engineering UTP University of Sciences and Technology in Bydgoszcz

4. Scientific achievement

A. Title of the scientific achievement:

Data analysis and classification techniques for improving cyber security of applications and computer networks.

B. Publications comprising a scientific achievement (monothematic series of publications):

Number of publications comprising the achievement	**11**
Number of publications on JCR list	**7**
Scoring of the ministry (including the percentage share)	**151**
Scoring of the ministry	**220**
Total IF of the achievement	**7,223**

1. **[SCN17] Rafał Kozik [80%],** Choraś M.[20%], Pattern Extraction Algorithm for NetFlow-Based Botnet Activities Detection, Security and Communication Networks, vol. 2017, Article ID 6047053, 10 pages, https://doi.org/10.1155/2017/6047053, 2017 ***IF=0.904***

   My contribution is:
   - *Proposal of the approach presented in the paper, in particular, proposal of a system architecture, development of an algorithm for time windows-based analysis of network flows and training methods of the Random Forest classifier based on an imbalanced data*
   - *Planning and performing the described experiments, in particular, selection of metrics for assessing the effectiveness of the proposed solution*
   - *Analysis of the obtained results*
   - *A key participation in writing the manuscript*

   My contribution is estimated at: 80%

2. **[PRL18] Kozik Rafał [100%],** Distributing extreme learning machines with Apache Spark for NetFlow-based malware activity detection, Pattern Recognition Letters, Volume 101, 14-20, 2018 ***IF=1.952***

   I am the only author of this work.

3. **[JPDC18] Rafał Kozik [50%],** Choraś M.[20%], Massimo F.[20%], Francesco P.[10%], A scalable distributed machine learning approach for attack detection in edge computing environments, Journal of Parallel and Distributed Computing, Volume 119, 18-26, 2018 **IF=1.815**.

   My contribution is:
   - *Definition and implementation of the method proposed in the paper, in particular realisation of ELM classifier learning schema, which is based on cloud resources and dedicated for edge computing applications*
   - *Participation in experiments definition, in particular, selection of demonstration scenarios related to IoT and edge computing.*
   - *Analysis of obtained results*

- *Final version proofreading and coordination of the publishing process*

My contribution is estimated at: 50%

4. **[CISIM14] Kozik Rafał [65%],** Choraś M. [15%], Renk R.[10%], Hołubowicz W.[10%], A Proposal of Algorithm for Web Applications Cyber Attacks Detection , in Saeed K. and Snasel V. (Eds.): Computer Information Systems and Industrial Management, CISIM 2014, November, Vietnam, Lecture Notes in Computer Science, vol. 8838, 680-687, Springer, 2014 **(publication indexed in WoS)**

   My contribution is:
   - *Proposal and implementation of the method presented int the paper, in particular, realisation of algorithm for HTTP content encoding based on compression technique*
   - *Definition and execution of the experiments, in particular comparison of the proposed algorithm with other method*
   - *Analysis of the obtained results*
   - *Coordination of publishing process*

   My contribution is estimated at: 65%

5. **[CISIS15] Kozik Rafał [60%],** Choraś M. [20%], Renk R. [10%], Hołubowicz W. [10%], Patterns Extraction Method for Anomaly Detection in HTTP Traffic, in: Herrero A., Baruque B., Sedano J., Quintan H., Corchado E. (Eds), International Joint Conference CISIS'15 and ICEUTE'15, Advances in Intelligent Systems and Computing, 227-236, 2015 **(publication indexed in WoS)**

   My contribution is:
   - *Proposal and implementation of the method presented in the paper, in particular, realisation of algorithm for request structure extraction for improving the effectiveness of attacks detection in the application layer*
   - *Definition and execution of the experiments, in particular, comparison of techniques for features extraction*
   - *Analysis of the obtained results*
   - *Coordination of the publishing process*

   My contribution is estimated at: 60%

6. **[IGPL15]** Choraś Michał [50%], **Kozik Rafał [50%],** Machine learning techniques applied to detect cyber attacks on web applications, Logic Journal of the IGPL, vol. 23(1): 45-56, 2015 ***IF=0.461***

   My contribution is:
   - *Key participation in the definition of the method proposed int the paper and its implementation; in particular proposal of the graph-based method for request segmentation.*
   - *Participation in experiments definition*
   - *Paper writing and proofreading*

   My contribution is estimated at: 50%

7. **[3PGCIC15] Kozik Rafał [70%],** Choraś M.[30%], Adapting an Ensemble of One-Class Classifiers for a Web-Layer Anomaly Detection System , in Proc. Of 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC, November, IEEE CPS, Cracow, 724-729, 2015 **(publication indexed in WoS)**

My contribution is:

- Proposal of the method presented in the paper and its implementation, in particular, realisation of method for request structure extraction and techniques to combine it with the classification algorithms
- Definition and execution of the experiments
- Analysis and description of the obtained results

My contribution is estimated at: 70%

8. **[HAIS16] Kozik Rafał [80%],** Choraś M.[20%], Solution to Data Imbalance Problem in Application Layer Anomaly Detection systems, in Martinez-Alvarez F., Troncoso A., Quintian H., Corchado E. (Eds.): Hybrid Artificial Intelligent Systems, HAIS 2016, LNAI, Springer, vol. 9648, 441-450, 2016 **(publication indexed in WoS)**

   My contribution is:

   - Proposal of the approach presented in the paper and its implementation, in particular, proposal of method for classifier learning based on imbalanced data, as well as a technique for request structure extraction by means of genetic algorihtm
   - Definition and execution of the experiments scenarios
   - Coordination of the publishing process of the paper

   My contribution is estimated at: 80%

9. **[SCN16] Kozik Rafał [80%],** Choraś M.[10%], Hołubowicz W.[10%], Evolutionary-based packets classification for anomaly detection in web layer, Security and Communication Networks, Wiley, vol. 9, Issue 15, 2901-2910, 2016 ***IF=1.067***

   My contribution is:

   - Proposal of the approach described in the paper, including its implementation
   - Definition and execution of the experiments scenarios, in particular comparison of methods for structure extraction and techniques for classification
   - Participation in results description and analysis

   My contribution is estimated at: 80%

10. **[IGPL17] Kozik Rafał [70%]**, Choraś M. [20%] and Hołubowicz W.[10%], Packets tokenization methods for web layer cyber security, Logic Journal of the IGPL - 2017, 25, 1, 103-113, 2017 ***IF=0. 575***

   My contribution is:

   - Proposal of the approach presented in the paper and its implementation, in particular realisation of algorithms for request grouping and structure extraction
   - Participation in experiments scenarios definition, in particular comparison of methods for application layer attacks detection
   - Participation in analysis and description of the obtained results

   My contribution is estimated at: 70%

11. **[IGPL18] Kozik Rafał [75%]**, Choraś M. [25%], Protecting the application layer in the public domain with machine learning methods, Logic Journal of the IGPL, 2018, DOI: https://doi.org/10.1093/jigpal/jzy029 ***IF=0. 449***

   My contribution is:

- Proposal of the approach presented in the paper and its implementation, in particular the realisation of an algorithm for extracting the request structure based on suffix tables, as well as combining this method with the Random Forest classifier
- Participation in experiments scenarios definition, in particular comparison of methods for application layer attacks detection
- Participation in analysis and description of the obtained results

My contribution is estimated at: 75%

C. Detailed description of the above-mentioned scientific achievement.

Introduction

The cyber security topic covers a wide spectrum of aspects including purely technical (e.g. development of new tools for cyber threats countering), organisational (e.g. increasing competences of employees in the area of cyber security), and strategic ones (e.g. cyber security strategy).
However, in my research I focus on two key problems of protecting the computer networks against cyber attacks:

- **P1.** Analysis of aggregated network traffic for anomaly and infected hosts detection
- **P2.** Analysis of requests in application layer for protecting the application against cyber attacks

As for the problem of aggregated traffic analysis (P1), my research focuses mainly on analysis of computers building the network. Basing on the traffic generated by the computers (hosts) it is possible to build mechanism for detecting anomalies, cyber attacks and symptoms indicating that specific host has been infected with malicious software (e.g. Ransomware, SPAM-bot, etc.).
When it comes to application layer (P2), the significance of this topic is supported by numerous reports[1] and information from the press[2]. Currently, vulnerabilities in web systems are easier to detect by cybercriminals due to a number of available tools (e.g. vulnerability scanners), openness and availability of the application code (or their parts, e.g. a framework for content management) and easy access to vulnerable applications.
In both areas (network traffic analysis and application layer protection), my research focuses largely on the development of feature extraction techniques (based on the analysed data) and their classification for the detection of network threats.
The presented set of scientific papers is an achievement that forms the basis of this application and presents key aspects of my research. These are presented thematically with the distinction into the aforementioned areas of interest.

[1] Internet Security Threat Report https://www.websecurity.symantec.com/security-topics/istr-2017-infographic
[2] Cyber attack on polish election committee's servers http://www.locos.pl/index.php/incydenty-epidemie/377-wlamanie-na-serwery-polskiej-komisji-wyborczej

C.1. Analysis of aggregated network traffic for anomaly and infected hosts detection

The analysis of aggregated network traffic can be made on the basis of network flows. A network flow is defined as a packet stream flowing in one direction between a given source and destination. Flows are described by a series of parameters such as IP address, sender's and recipient's port number (services), duration of the connection, number of bytes transmitted, and protocol type. Analysis of network flows is one of the popular techniques for auditing and monitoring ICT networks. However, data collected for small IT networks in a few-hour period can exhibit large volume. In addition, a single flow usually does not contain enough information to detect the anomalous behaviour of machines composing an IT network. Usually, an additional element of analysis is required, which allows for analysing the collected data (from a different perspectives). Therefore, in this area, solutions supporting the analysis of large data sets are desired. In particular, in cybersecurity applications they allow for more effective detection of malicious software. At the same time, they are expanding the range of administrator tools to combat cyber threats.
As part of my research [SCN17, PRL18], a network traffic analysis system was developed to detect network anomalies. In order to enable a large scale of data collection, analysis and classification, the concept and architecture of the proposed system was based on the Apache Spark system. It is a free and highly scalable solution supporting the creation of a flexible computing cluster. The created system and developed algorithms analyse network traffic collected in the form of network flows (called NetFlow). The calculated detection model (using the computing cluster) is transferred directly to the NIDS (Network Intrusion Detection System). The general schematic diagram is shown in Fig. 1.

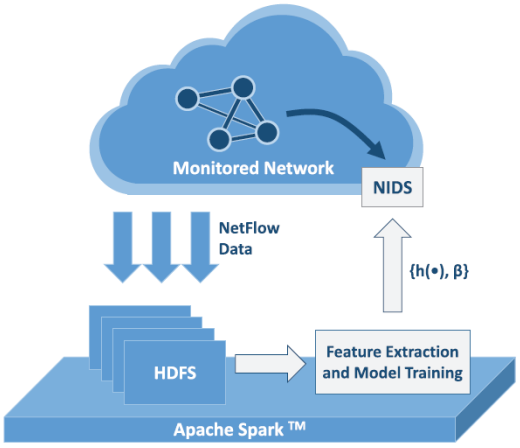

Fig. 1 The general outline of the proposed system based on the Apache Spark platform [JDPC18].

The research presented in **[SCN17]** was mainly focused on proposing scalable and innovative method for features extraction and network traffic analysis. In order to fully utilise the resources of the cluster I have proposed to use distributed file system HDFS[3] (Hadoop Distributed File System) and scalable data processing pattern called MapReduce[4]. This approach allowed for parallel analysis of the network elements (composing computer network) behaviours based on collected network flows. In order to build the behavioural model of particular network element a dedicated feature extraction method has been adapted. First, all network flows are collected in so called fixed-length time windows. Next, data are grouped based on key which is formed

[3] HDFS https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[4] MapReduce https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

out of sender IP address and time-window number. Foreach group of the network flows feature vectors are extracted such as number of flows sum of bytes being sent, number of unique destination IP addresses (and ports), and average duration of the flow. Extracted feature vectors are used to train the model representing the behaviour of the computer network. For that purpose the Random Forest (RF). Again, to utilise cluster resources the distributed implementation of RF classifier based on MapReduce data processing pattern is used. In the research the problem of data imbalance is also addressed. In order to verify the effectiveness of the proposed method the CTU[5] evaluation dataset has been used. Experiments showed that the proposed method exhibits better effectiveness than other methods compared on the same dataset (e.g. BotHuner, CCD, BClus).

The research presented in **[PRL18]** proposes modification of the method described in [SCN17] which uses innovative approach based on ELM classifier that has been implemented in above mentioned Apache Spark architecture. In the proposed approach the original feature vectors are transformed using single layer of neutrons. In the next step the response of this later is used to train linear classifier. Similarly as in [SCN17], the feature extraction algorithm uses statistical features calculated for network flows in so called time windows. The feature vectors are distributed among cluster nodes. This allows for to train different models in parallel. The ELM classifier uses artificial neural network architecture with single hidden layer (Fig. 2). In contrast to traditional neural network, only the neurons in the visible layer are trained. Additionally, the training process is not iterative and weights in visible layer are calculated using classical linear algebra tool.

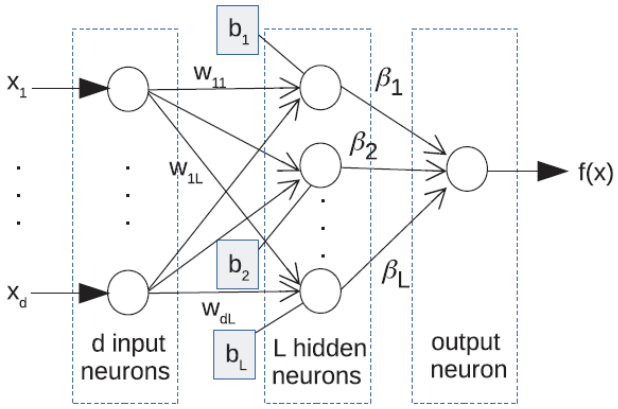

Fig. 2 The architecture of adapted ELM classifier.

In this method I exploited the fact that the learning data obtained from network flows result in a matrix with a large number of rows (a large number of measurements) and a small number of columns (12 attributes describing the flow). This allows to parallelise the process of determining the values of the output layer weights. The results obtained on the CTU dataset showed that the proposed method allows for achieving high efficiency in detecting anomalous behaviour of hosts producing at the same time a relatively low rate of false positives.

In **[JPDC18]** ELM classifier (mentioned above) was proposed to detect network attacks in the IoT (Internet of Things) environment. In this approach, the innovative Security-as-a-Service model is proposed, so that computationally complex operations are processed in the cloud. In the proposed approach, the detection model, which is deployed in the IoT environment, is trained in the cloud (using the Apache Spark platform) based on the provided learning data. A schematic diagram showing the classifier learning algorithm is shown in Fig. 3

[5] CTU Dataset https://mcfp.weebly.com/

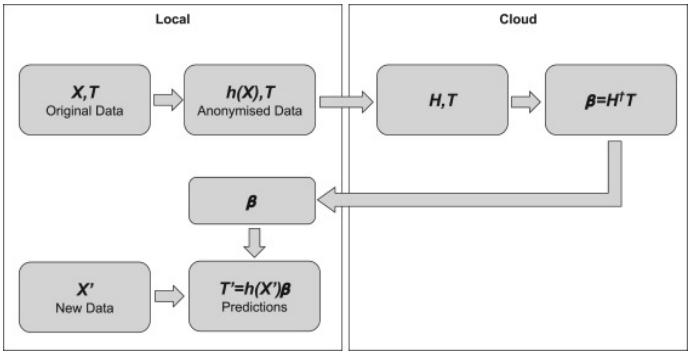

Fig. 3 Schematic overview of ELM classifier training process, which is exploiting cloud resources.

C.2. Analysis of requests in application layer for protecting the application against cyber attacks

Currently, the problem of protecting the web applications against hackers is one of the key challenges for cybersecurity[6]. It is estimated that nowadays it is much easier to find and use the vulnerability of the web application than to attack another component in the network (e.g. server's operating system). In addition, the functional and technological diversity of web applications is an obstacle for security mechanisms. This is a big challenge when creating rules for firewalls, which are designed to filter out malicious requests. In addition, each application can have a specific communication protocol between the client and the server (e.g. HTTP, JSON-RPC, RESTFul API, SOAP). Therefore, in this area there are many challenges associated with the extraction of features and the construction of an appropriate detection mechanisms.
In a typical web application, there are two key elements of its architecture, namely client providing the front-end interface and the server-side (back-end) providing services. Usually, via the font-end interface, the client sends a request to the remote server and receives the result in response. The idea of protecting the application layer in such architecture is based on intercepting requests sent by clients and filtering out malicious ones (e.g. containing injected code). The general outline of the method is shown in Fig. 4.

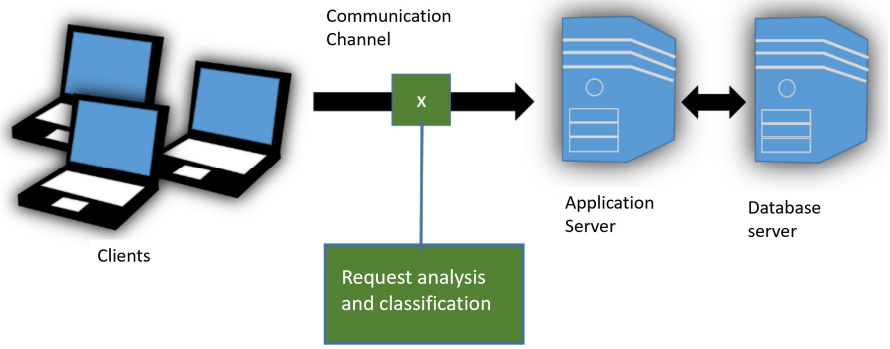

Fig. 4 The general overview of request analysis method.

[6] OWASP Top 10 https://www.owasp.org

In this context, the research presented in **[CISIM14]** proposes a innovative technique for feature vectors extraction based on the LZW compression method[7]. First, learning data containing genuine and malicious requests is collected. In the case of a web system, genuine requests can be obtained during it use (e.g. during functional testing). The malicious requests can be collected with tools that automate penetration tests. Nevertheless, in the described solution, to compare the results, I proposed using a publicly available CSIC benchmark database[8]. First, the dictionary for training dataset is extracted:

$$D: word \rightarrow \{i: i \in N\}$$

It converts a given string (word) into a numerical value. In the described solution, the string is the entire body of the HTTP request, which is transmitted via the front-end interface to the web application.

Data: Set of HTTP payloads S
Result: Dictionary D
s = empty string
while *there is still data to be read in* S **do**
  $ch \leftarrow$ read a character
  if $(s + ch) \in D$ **then**
    $s \leftarrow s+ch$;
  else
    $D \leftarrow D \cup (s + ch)$;
    $s \leftarrow ch$;
  end
end

Fig. 5 The algorithm for incremental dictionary building.

The feature vectors extracted with the dictionary-based method are additionally coded using a histogram. That way, the set of fixed length vectors is obtained and used to train the classifiers. The proposed method detected 96% of anomalies, while generating about 3.5% of false alarms. The ROC curve for different classification methods is shown in Fig. 6.

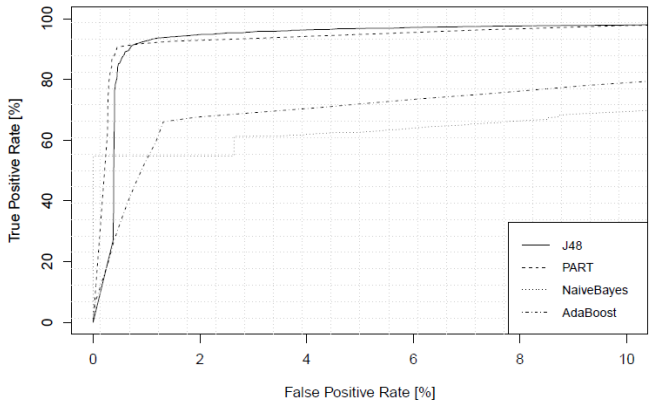

Fig. 6 ROC curve for selected classifiers.

The research presented in **[CISIS15]** is a continuation of the method described in [CISIM14]. In the present paper, it was hypothesized that the information about the request structure allows for a smaller final classification error. The present research focuses on the HTTP protocol, which is one of the most popular protocols of the application layer. The structure extraction algorithm is a two-step process. First, a group of requests is clustered according to the request

[7] LZW Lempel-Ziv-Welch https://www2.cs.duke.edu/csed/curious/compression/lzw.html
[8] CSIC'10 dataset http://www.isi.csic.es/dataset/

method (e.g. GET, POST, PUT, DELETE) and the URL of the resource on which the request is executed. The example of this process is shown in Fig. 7.

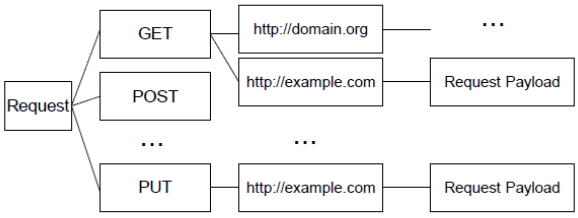

Fig. 7 The example of the request structure categorisation.

Each request that is assigned to the specific group, based on the request method and the URL, has a request payload, which contains additional parameters. Some examples of request parameters for the POST method are shown in Fig. 8.
.

```
POST /test/demo_form.php HTTP/1.1
Host: w3schools.com
cht=p&chs=500x250&chdl=
first+legend%7Csecond+legend%
7Cthird+legend&chl=first+label%
7Csecond+label%7Cthird+label&
chco=FF0000|00FFFF|00FF00,6699CC|
CC33FF|CCCC33&chp=0.436326388889&
chtt=My+Google+Chart&chts=000000,
24&chd=t:5,10,50|25,35,45
```

Fig. 8 Example of request body (in bold) – POST method.

A characteristic feature of the request body is that its parameters can be serialised in various formats (e.g. key-value fields combined with the '&' sign or a series of values connected by the '|' sign). A typical format for storing more complex objects in a modern web applications is the JSON notation (JavaScript Object Notation). However, regardless the serialisation standard, requests sent to the specific URL address will have a similar structure (e.g., names and parameter values, order of their occurrence, etc.). Therefore, in this area of research I have focused on proposing an algorithm for its extraction.

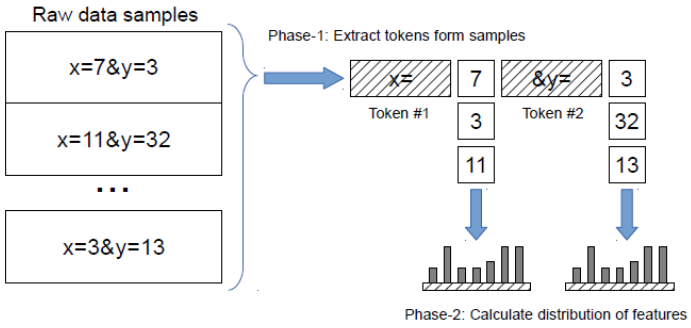

Fig. 9 Token-based request structure extraction.

In the proposed approach, the structure of the request is described using the so-called tokens. The token is defined as a string of characters that appears repeatedly in a set of requests sent to a given URL. An example of token-based HTTP requests structure description is shown in Fig. 9. Tokens divide the content of an HTTP request into smaller parts. Strings of characters that are located between tokens are described using the histogram of characters. This way, a single request (depending on the structure) can have several feature vectors.
In this work, the LZW method was used to identify a potential list of tokens. In that case, the list of tokens is a collection of strings maintained in the LZW dictionary. This list is additionally processed in the next steps. First, the tokens which do not appear in all requests are deleted. Moreover, the tokens that are part of other tokens are also removed. The experiments on the benchmark dataset proved the correctness of the hypothesis that information about the request structured help to improve the classification process.

In **[IGPL15]**, another and innovative algorithm for extracting tokens from HTTP requests has been presented. The proposed method uses the technique of iterative grouping to produce regular expressions. An example of joining two strings "value = 5 & sid = 1" and "value = 16 & sid = 1" into one component described with the regular expression "^ value = [0-9] + & sid = 1 $" is shown in Fig. 10.

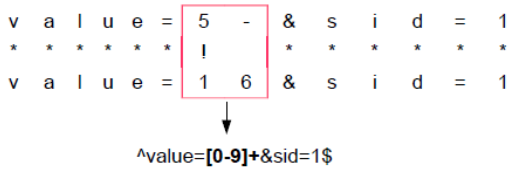

Fig. 10 An example of joining two strings of characters into a regular expression.

In the proposed approach, the set of requests is represented as a graph $G = (V, E)$, where V represents set of vertices (each vertex represents particular HTTP request), and E represents set of edges describing similarity between vertices. To describe the distance between two vertices (the difference between two HTTP requests), the Levenshtein distance was used[9]. In each iteration of the algorithm, a single edge of the graph is analysed. The vertices, which have the distance smaller than the global threshold, are joined and described with a regular expression. To generate a regular expression from two strings, the Needleman-Wunsch algorithm was used[10]. It finds a match between two strings by entering modifications (inserting a blank character or deleting a character) to both of them. For the example shown in Fig.10, in the first string the character 5 will be removed (the "!" character in the middle row) and one empty character ("-" symbol) inserted. In the second string, the character 1 will be removed. Characters removed or overlapping with the offset (empty character) are described using the following regular expressions:

- [0-9]+ for digits only,
- [a-z]+ for lower-case characters only,
- [A-Z]+ for upper-case characters only,
- [special characters]+ for special characters such as !,*,?, etc.

If the selected characters are the combination of several of the above patterns (e.g. lowercase letters and numbers) they will be described as [0-9a-z]+. The proposed method obtained results

[9] Levenshtein distance http://www.levenshtein.net/
[10] Needleman, Saul B. & Wunsch, Christian D. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins". Journal of Molecular Biology. 48 (3): 443–53. doi:10.1016/0022-2836(70)90057-4

similar to the method presented in [CISIM14] (94% of detected anomalies with 4% false positives). However, the value added for this method is the ability to directly generate regular expressions that represent a group of HTTP requests. These expressions can potentially be used as signatures in a firewall or as validation filters in the application.

Three modifications to [IGPL15] and [CISIS15] have been proposed in **[3PGCIC15]**. The first one concerns the method for the HTTP request structure extraction and the second method for anomaly detection based on machine learning. In contrast to the previous approach, the N list of candidate tokens $T = \{t_1, \dots, t_N\}$ is extracted using the suffix tree. It allows for identification of common substrings (also the longest one), which build a list of potential tokens. From the list of potential tokens, the S subset is selected, because not all tokens from the T list appear in all the analysed requests (P). Moreover, the tokens may appear in a specific order. Therefore, in the next iteration of the algorithm, the sequence of selected S tokens and the matrix $m_{i,j}$ are updated. The matrix holds information where *i*-th token appears in *j*-th HTTP request. The procedure is described in the following way:

```
1  S ← {0}
2  WHILE ∀ p ∈ P, p ∉ ∅: //while the sequence is not empty
3    d ← {0}
4    FOR t_i ∈ T:
6      // find the position of t_i token
7      d ← d ∪ max_j m_{i,j}
8    END
9    // find token which closest to the beginning of the sequence
10   k = argmin_i d_i
11   S ← S ∪ t_k
12   For each sequence p ∈ P remove characters containing t_k
13   Update m matrix
14 END
15
```

The entire algorithm is repeated until none of the HTTP request sequences are empty. In the first step of each iteration, a list of tokens is analysed and for each token the its furthest position in the HTTP sequence is stored (in the vector d). Then, based on the list d, the token that is closest to the beginning of the analysed HTTP sequences is selected. The selected token is stored in the list of S solutions. Additionally, for all sequences in P the characters belonging to t_k are deleted. The last step of each iteration is to update the matrix *m*.
At a final stage, as in previous solutions, strings between tokens are described using a histogram. The presented paper also evaluated the possibilities of improving the detection efficiency using classifiers hybridization techniques (boosting and bagging). The studies considered two base classifiers named Decision Stump (decision tree with a single node) and Reduced Error Pruning Tree (RepTree).
In **[HAIS16]**, I proposed to use an innovative approach adapting genetic algorithm to extract the structure based on a series of HTTP requests. In the proposed solution, similarly as in [3PGCIC15], the N list of candidate tokens $T = \{t_1, \dots, t_N\}$ are selected by means of suffix tree. However, the way of its selection to extract the structure of the request is relatively different. This paper proposes a token selection algorithm that utilises the global cost function:

$$C(x) = \sum_{i}^{n} v_i x_i$$

where v_i indicates the value of the token (number of characters included in the token - longer tokens are favoured) and $x_i \in \{0,1\}$ indicates that *i*-th token (out of n available) has been selected. Optimization of the C (x) function is constrained with the following condition:

$$\sum_{i}^{n} w_i x_i \leq W$$

where w_i of selecting particular token. This cost is related to the position on which the given token occurs in a set of requests. For example, if the token appear in the final position and is selected first, then it blocks the option of selecting tokens in front of it (shown in Fig. 11).

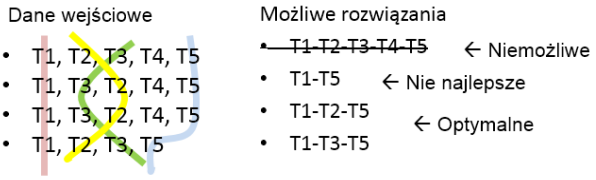

Fig. 11 Various combinations of tokens selection (possible solutions) for sample input data sequences.

In such situation w_i is assigned to a high cost. On the other hand, the value of W is extracted experimentally. It allows avoiding the situations where only a small number of tokens are chosen as a solution.
Finally, as in previous solutions, strings between tokens are described using a histogram. These vectors are used to train a balanced Cost-sensitive AdaBoost classifier. The proposed method allows obtaining a hihg detection rate of attacks (91.5%) with a low ratio of false alarms (0.7%).

In **[SCN16]** the continuation of research on the evolutionary method of tokens extraction has been proposed. In this work, the sections on experiments have been significantly expanded. In particular, signature-based methods (PHPIDS, ApacheMod) and solutions based on anomaly detection have been compared. The following methods have been considered:

- ICD (Idealised Character Distribution), which uses a histogram of characters (extracted from the entire HTTP request) and a Chi-square metric.
- ICDSeg, which analyzes the arguments (parameters in the form of url?paramer1=value1& parameter2=value2) of the HTTP request and use the ICD method for each value.
- Compression-based[11], which analyze statistical parameters of requests after compression.
- LVSD, which extracts nonparametric model based on Levenshteina distance.
- ADS{Chi, RT}, which are a combination of the proposed method for structure extraction with a Chi-square (Chi) metric and the classifier "Reduced Error Pruning Tree" (RT).

[11] Gordon Rueff, Lyle Roybal, Denis Vollmer, SCADA Protocol Anomaly Detection Utilizing Compression (SPADUC), The INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, scientific report, 2013

Tab. 1 Comparison of selected methods with the proposed solutions (ADSChi and ADSRT)

Type	Method	CSIC'10+ True Positive Rate	CSIC'10+ False Positive Rate	CSIC'10+ Precision
Signature-based	PHPIDS	0.2040	0.0125	0.9071
	ApacheMod	0.2630	0.0034	0.9786
Anomaly-based	ICD	0.3320	0.0010	0.9890
	Compression	0.4300	0.0000	1.0000
	LVSD	0.6230	0.0010	0.9970
	ICDSeg	0.8340	0.0110	0.9650
	ADSChi	0.9110	0.0070	0.9810
	ADSRT	0.9190	0.0070	0.9800

The results of the comparison are summarized in Table 1. The experiments confirmed that the proposed method of structure extraction achieves better detection results (True Positive Rate) with a relatively low level of false alarms (False Positive Rate).

In **[IGPL17]**, an additional algorithm for clustering HTTP requests was proposed. It makes the algorithm of token extraction independent from the HTTP protocol. In previous solutions, the request structure was analysed for each unique URL address. The general algorithm of processing HTTP requests is shown in Figure 12.

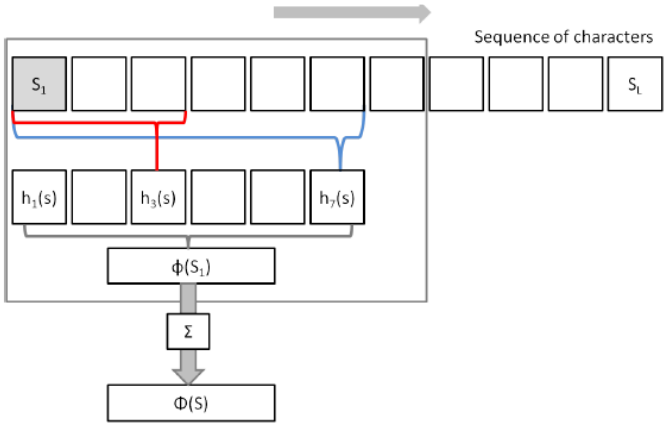

Fig. 12 A general algorithm for mapping HTTP (S) request strings into a feature vector ($\Phi(S)$) based on a sliding window and a set of 7 hash functions (h) that are concatenated (ϕ) and accumulated (Σ) at each position of the sliding window.

In this approach, each request is coded using a sliding window and a set of 32-bit hash functions in the form of:

$$h: A \rightarrow \{0,1\}^d$$

where A is an string of any length and d is a constant of 32. Each hash function generates a 32-bit string for the given part of the analysed sequence. The bit strings are joined in the process of concatenation. The bit sequence is added to the current state of the internal accumulator (the number of ones at specific position are counted). At each step of the algorithm, the analysis

window is moved to the next position in the request sequence and the whole procedure is repeated. The entire algorithm is terminated when the entire HTTP request string is analysed.

The extracted features vectors describing the HTTP request are grouped using the k-means algorithm. For each cluster, a procedure for extracting (and describing) the structure of requests is triggered. In the presented approach (in contrast to [SCN16]), the problem of structure extraction was formalized as the k-LCS (Multiple Sequences Longest Common Subsequence) problem. LCS problem for k = 2 is defined as:

$$LCS(A_i, B_j) = \begin{cases} LCS(A_{i-1}, B_{j-1}) + a_i & a_i = b_j \\ longest(LCS(A_i, B_{j-1}), LCS(A_{i-1}, B_j)) & a_i \neq b_j \\ \emptyset & otherwise \end{cases}$$

The computational complexity of a defined in this way problem is $O(n^k)$, where n indicated the length of the longest sequence and k the total number of sequences. Therefore, the present method uses a heuristic algorithm that iteratively adjusts selected pairs of character strings. In other words, the LCS problem for two strings is solved many times. The procedure takes place in the following steps:

1. The matrix of mutual similarity between all analysed strings of HTTP requests is calculated.
2. The so-called guiding tree is built on the basis of the similarity matrix. It allows for controlling the process of matching individual pairs.
3. The pairs that have the highest similarity coefficient are matched first. Matching pairs are joined into a single string.
4. In the next steps, pairs that are more distant to each other are matched.

The quality of the final result can be controlled by the global threshold value (T value shown in Fig. 13).

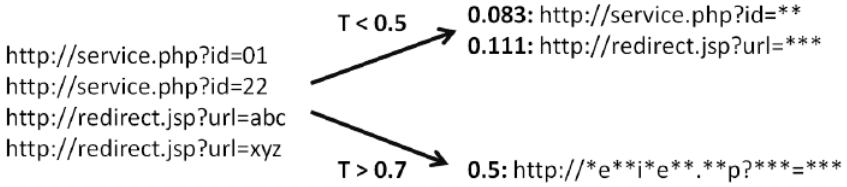

Fig. 13 The result of the requests structure extraction for a different threshold values T.

A comparison of the proposed method with other solutions is shown in Table 2. A comparable results have been obtained in contrast to [SCN16]. Nevertheless, the proposed pre-grouping procedure using the k-means algorithm makes the process of extracting the structure independent from the URL and request method.

Tab. 2 Comparison of the proposed method (StrEXT + RF) with other algorithms.

Type	Method	Performance Characteristics	
		True Positive Rate	False Positive Rate
Signature-based	PHPIDS	0.204	0.013
	ApacheMod	0.263	0.003
Anomaly-based	StrExt	0.771	0.01
	ICD	0.697	0.064
	RF	0.849	0.056
	n-grams	0.809	0.142
	StrExt + RF	**0.927**	**0.064**

In **[IGPL18]** it is presented the continuation of the research shown in [SCN16]. In particular, a different technique that uses suffix tables has been used to extract the list of candidate tokens. In addition, the modification of the Random Forest classifier was used in the classification process. The modification allows assigning appropriate weights to classification errors (e.g. false alarms). This allowed obtaining a 5% of false alarm rate while heaving 93% of attack detection rate.

The results of my work, the proposed methods and algorithms may be used as additional tools to improve the security of ICT systems. These tools include:

- Network traffic monitoring systems for the purpose of detecting anomalies in the behaviour of machines building a ICT network.
- Models of threat detection based on network flows.
- Firewalls and anomaly detection system in the application layer.
- Filtering and validation systems for requests sent by the client to the server

The main scientific contributions, being a contribution to the field of computer science, include:

1. Innovative method for aggregated traffic analysis based on machine learning and Big Data tools.
2. Implementation of ELM (Extreme Learning Machine) classifier learning mechanism base on cloud computing resources.
3. Conceptualisation of techniques for detecting cyber attacks in application layer based on request content analysis.
4. Conceptualisation and implementation of innovative methods for request structure extraction based on collected traffic for increasing the effectiveness of anomaly and cyber attacks detection targeting application layer.

5. Presentation of other scientific achievements.

5.1. Before obtaining the Ph.D. degree (2008-2013)

Before obtaining the Ph.D. degree, the area of my research was substantially different from the one I had focused now.
First of all, my research before obtaining a Ph.D. degree was focused on the methods of image processing and analysis in various applications such as biometrics and assisting the blind people in everyday activities. A list summarising bibliometric indicators is shown in the table below.

Total IF	0.814
Publications on JCR list	1
Publications in WoS	21
Number of citations WoS	15
Number of citations WoS (without self citations)	13
h-index WoS	2

5.2. After obtaining the Ph.D. degree (2013-2018)

After obtaining the Ph.D. degree, the subject area of my research focused on cyber security, critical infrastructure protection and the use of machine learning methods and data analysis to improve the quality of the software development process. A list of bibliometric indicators is shown in the table below.

Total IF	14.945
Publications on JCR list	14
Publications in WoS	58
Number of citations WoS	131
Number of citations WoS (without self citations)	86
h-index WoS	6

In addition to 7 publications from the JCR list, which were shown in the postdoctoral achievement, I am also a co-author of 7 other publications from the JCR list.

The subject of my research focuses on three main thematic groups, which are presented below along with selected publications.

Critical infrastructure protection and crisis management

The main research topics concerning to the critical infrastructure protection and crisis management are related to the European research projects in which I participated, i.e. FP7 CIPRNet, FP7 Inspire and FP7 Intersection. In my research, I addresses issues such as:

- modelling of key aspects of cyber security [1,2] for the purpose of simulating the behaviour of ICT systems,

- defining dedicated ICT services for the purpose of crisis management [3],
- the possibility of using machine learning algorithms to model system behaviour and attack patterns [4],

1. Kozik Rafał, Choraś Michal, Hołubowicz W., Renk R., Increasing Protection and Resilience of Critical Infrastructures - Current challenges and approaches, Journal of the Polish Safety & Reliability Association, vol. 6, number 3, 79-84, 2015
2. Massimo Ficco, Choraś M., Kozik R., Simulation Platform for Cyber-Security and Vulnerability Analysis of Critical Infrastructures, Journal of Computational Science, vol. 22, pp.179-186, 2017 **IF=1.748**
3. Kozik Rafał, Choraś Michal, Flizikowski A., Theocharidou M., Rosato V., Rome E., Advanced services for critical infrastructures protection, Journal of Ambient Intelligence and Humanized Computing, vol. 6(6), 783-795, Springer, 2015 **IF=0.835**
4. Michał Choraś, Rafał Kozik, Machine Learning Techniques for Threat Modelling and Detection, Security and Resilience in Intelligent Data-Centric Systems and Communication Networks / Massimo Ficco, Francesco Palmieri, Elsevier, pp.179-192, 2017

Analysis of the quality of the software production process

Another area of my research interests associated with the assumption that problems with the security holes in the software are closely related to the quality of its software development process. In my research [5,6,7,8] I focus on:

- aspects of measuring the quality of the code,
- methods of measuring the process of managing software development,
- methods and algorithms for early detection of design problems that may affect the quality of the code.

5. Kozik Rafał, Choraś M., Puchalski D., Renk R., Data Analysis Tool Supporting Software Development Process, In proceedings of 14th International Scientific Conference INFORMATICS, ISBN 978-1-5386-0888-3, IEEE catalog number CFP17E80-PRT, Poprad, pp.179-184, 2018
6. Kozik R., Choraś M., Puchalski D., Renk R. (2019) Platform for Software Quality and Dependability Data Analysis. In: Zamojski W., Mazurkiewicz J., Sugier J., Walkowiak T., Kacprzyk J. (eds) Contemporary Complex Systems and Their Dependability. DepCoS-RELCOMEX 2018. Advances in Intelligent Systems and Computing, vol 761. Springer, Cham
7. Kozik Rafał, Choraś M., Damian P., Rafał R., Q-Radpis Framework for Advanced Data Analysis to Improve Rapid Software Development. Journal of Ambient Intelligence and Humanized Computing https://doi.org/10.1007/s12652-018-0784-5 **IF=1.423**
8. Choraś M., Kozik Rafał, Puchalski D. et al. Increasing product owners' cognition and decision-making capabilities by data analysis approach. Journal of Cognition, Technology & Work. https://doi.org/10.1007/s10111-018-0494-y **IF=1.26**

Distributed processing

Another important aspect of my research is the use of distributed computing techniques and tools for analysis, visualization and cyberthreat detection [9]. In that regards, I'm also interested in the new techniques for providing services and applications in the cloud environments. These solutions are designed to allow the user to reduce the time of access to the service [10].

9. Rafał Kozik, "Distributed System for Botnet Traffic Analysis and Anomaly Detection," 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, 2017, pp. 330-335.

10. Sebastian Łaskawiec, Michał Choraś, Rafał Kozik. New Solutions for exposing Clustered Applications deployed in the cloud. Cluster Computing Journal, DOI: 10.1007/s10586-018-2443-1 **IF=1.601**

Practical application of data mining and machine learning

Another thematic group of my interests and research areas are various practical applications of machine learning techniques and data analysis. In particular I participated in a research related to techniques of detecting anomalies in ICT networks [11,12] and methods for extracting features from the image based on a simplified model of vision cortex [13].

11. Saganowski L., Andrysiak T., Kozik Rafał and Choraś Michal, DWT-based anomaly detection method for cyber security of wireless sensor networks , Security and Communication Networks, vol. 9, Issue 15, 2911-2922, Wiley, 2016 **IF=1.067**

12. Tomasz Andrysiak, Łukasz Saganowski, Michał Choraś, and Rafał Kozik. Proposal and comparison of network anomaly detection based on long-memory statistical models, Logic Journal of IGPL, vol. 25, no. 6, 944-956, 2016 **IF=0.575**

13. Rafał Kozik, A simplified visual cortex model for efficient image codding and object recognition, Image Processing and Communications Challenges 5 / Ed. Ryszard S. Choraś, Berlin, Heidelberg : Springer-Verlag, 2014