UNIVERSITY OF MISKOC

FACULTY OF MECHANICAL ENGINEERING AND INFORMATICS

TRAINING ANOMALY-BASED INTRUSION DETECTION SYSTEMS USING MACHINE LEARNING

Hatvany József Doctoral School of Information Sciences

Head of Doctoral School:

Prof. Dr. Jenő Szigeti

Thesis Booklet

Supervisor: Dr. habil. Zsolt Csaba Johanyák

Name: László Göcs

Neptun: BVX3AL

2023

Defense Committee

Chairperson:	Prof. Dr. Jenő Szigeti	University of Miskolc, full professor
Reserve chairperson:	Prof. Dr. László Kovács	University of Miskolc, full professor
Members:	Prof. Dr. Szilveszter Kovács	University of Miskolc, full professor
	Dr. Attila Pásztor	NJE, college professor
	Dr. Zoltán Király	University of Dunaújváros, associate professor
Reserve member:	Dr. habil. József Vásárhelyi	University of Miskolc, associate professor
Secretary and member:	Dr. Olivér Hornyák	University of Miskolc, associate professor
Reserve secretary and member:	Dr. Péter Veres	University of Miskolc, assistant professor
Reviewers:	Dr. habil. Edit Laufer	Obuda University, associate professor
	Dr. Dávid Vincze	University of Miskolc, associate professor
Reserve reviewer:	Prof. Dr. Attila Kővári	Eszterházy Károly Catholic University, full professor

Contents

1 Research motivation and goals

I have been working in the field of IT security for more than 10 years. In addition to my university education job, as a consultant for companies and IT forensic expert, I have encountered numerous cases of vulnerabilities and problems with IT systems. The biggest challenge of IT systems in operation today is the existence and monitoring of security. Computer and network security systems are constantly under attack, which are now organized attacks.

Human monitoring is almost impossible with today's systems, so automation plays an important role in this area. Intrusion Detection Systems (IDS) are used to detect events, including attacks.

My research motivation was to build on the methods, results and technologies presented in the literature review to find an efficient method to train IDS systems. The study focuses on one of the most common attack modes, brute-force attack detection. To configure IDSs and determine the most appropriate algorithms, there are training data sets that contain various network communications, including attack communications.

My goal was to identify the most relevant features necessary for the training of an efficient IDS classifier module for the case of a selected data set and to find a classification algorithm that can efficiently and accurately identify a potential network attack. To this end, one of my stated objectives was to pre-process the data set and then determine the ranking of the features taking into consideration the feature scores obtained with several legacy methods. This allows to train and test classification algorithms.

The main theoretical result of my research is the development of two ensemble methods that facilitate the efficient feature selection in the case of machine learning problems with high number of dimensions.

2 Applied methodologies

2.1 Data set preprocessing

When processing a data set, especially in high-dimensional cases, it is of paramount importance to apply appropriate pre-processing and dimensionality reduction. High-dimensional data often contain a lot of redundant or noisy information, which can have a negative impact on the accuracy and efficiency of the analysis. Preprocessing steps such as handling missing data, identifying and handling outlier values, and normalization or scaling help to create cleaner and more reliable data. In addition, dimension reduction techniques allow large amounts of variables to be represented in fewer but relevant dimensions of the data. This improves interpretability, reduces noise, and helps improve the efficiency of analysis and modelling.

The data reduction phase focuses on feature selection and dimension reduction, which can have a number of benefits. One key benefit is that many data mining algorithms work better when the number of dimensions - the number of attributes (columns) in the data - is smaller. This is partly because dimension reduction eliminates irrelevant attributes and reduces noise. Another advantage is that it can lead to a more understandable model, as it will have fewer attributes. In addition, the reduced amount of data requires less storage space and less time to process.

The data set used in the course of the investigation related to my PhD studies is CSE-CIC-IDS2018 on AWS [1], which was created by the Canadian Institute for Cybersecurity laboratory. This data set was chosen because it was the most recent data set available at the initial stage of my research, it contains the attacks included in the research, and it meets all the criteria required for the research (e.g. total traffic, tagging, etc.). The data set was processed using the following steps:

1. *Data cleaning* includes deletion of rows (records) containing invalid or missing data, deletion of columns with the same value (e.g. columns where all values are zero), deletion of characteristics (columns) considered irrelevant for classification.
2. *Data transformation* means the transformation of categorical data into numerical data, normalization and splitting of the data set.
    a. *Normalization* consists of the conversion of numeric columns to a common scale.
    b. *Data splitting* to create training and test patterns.

Results:

During the pre-processing of the data set, I achieved a significant dimensionality reduction, since the initial 3 data sets had a column number of 80, and this was reduced to 69. Furthermore, the data transformation allowed to easily split the data sets into different files based on attack type, so further analysis was done by attack type.

Table 1. Training sets after dimension reduction

File name	Number of records	Number of columns
data set-ftp-tr.csv	171 433	69
data set-ssh-tr.csv	170 280	69
data set-web-tr.csv	417 592	69
data set-xss-tr.csv	417 211	69
data set-sql-tr.csv	417 068	69

Table 2. Test sets after dimension reduction

File name	Number of records	Number of columns
data set-ftp-ts.csv	85 716	69
data set-ssh-ts.csv	85 140	69
data set-web-ts.csv	209 101	69
data set-xss-ts.csv	208 720	69
data set-sql-ts.csv	208 577	69

2.2 Feature selection

Feature selection focuses on finding the most relevant attributes that can be used for effective classification or prediction [2] [3] [4].

It contributes to reducing the dimensionality of the problem and thus the resource requirements (storage, computation) and can improve the performance of machine learning algorithms [5], i.e. faster training, reduced overfitting, and sometimes better prediction ability.

Ensemble Feature Selection (EFS) is a technique that exploits the strengths of multiple feature selection algorithms to improve the identification of significant features in a data set. The benefits of ensemble feature selection include increased classification accuracy, reduced overfitting and increased stability of the selected features. This approach can be particularly beneficial in machine learning-driven applications, such as intrusion detection systems, where the diversity of features can affect the accuracy and learning time of the model. By combining the benefits of different feature selection algorithms, joint feature selection can facilitate the identification of the features that are most relevant to a given task, leading to more efficient and effective data analysis. Overall, EFS is an effective and popular technique for data selection that can improve model accuracy and reduce redundancy [6].

Feature selection methods used in my investigation work:

- Information Gain [7] [8] [9]
- Gain Ratio [10] [11]
- Relief [12]
- Symmetric Uncertainty [13] [14]
- Chi-square test [15]
- Analysis of variance (ANOVA) [16]

The six feature selection methods were applied to all five data sets on 30 university laboratory computers and using the ELKH cloud services [17]. Although I performed several tasks in parallel, the whole process took more than two months.

For each data set and each method, I normalized the feature score values obtained at the end of the feature selection process. I then calculated the final feature score for each data set separately as the average of the normalized scores. I then set a ranking threshold for the scores starting at 0.05 and increasing in 0.05 steps up to 0.55. For each threshold, I selected the characteristics with a score higher than the threshold, thus defining a reduced number of different groups of features (see Table 1).

Table 3. Number of selected features in the case of different ranking thresholds

Threshold value	**FTP**	**SSH**	**WEB**	**XSS**	**SQL**
0.05	56	59	65	65	66
0.10	43	53	60	57	64
0.15	32	48	60	57	60
0.20	23	29	58	51	57
0.25	21	22	56	46	48
0.30	13	17	50	36	37
0.35	8	7	44	31	31
0.40	3	2	34	27	26
0.45	2	2	23	10	12
0.50	2	1	9	6	4
0.55	2	1	1	1	2

Using the features defined for the individual thresholds, I tested five machine learning based classification algorithms to achieve acceptable or good classification results with low number of features. For each data set, I evaluated the classification algorithms with different classification performance measures using the features selected for the given thresholds. For each classifier, I selected the highest value taking into account the arithmetic mean of the Accuracy, Precision, Recall, F1 performance values (a number between 0-1, where 1 represents the best performance) obtained by examining the training and test sets. Thus, for each data set, I determined the threshold at which the smallest number of features achieves a good classification result.

Results:

From the 69 features resulted after the data set preprocessing, the number of selected features was determined for each attack type according to the defined thresholds using five classifier algorithms, and the results are shown in Table 2.

Table 4. Thresholds and feature numbers of best performing classifiers in the case of the five attack types

Data set	Threshold value	Number of features
FTP	0.35	8
SSH	0.35	7
WEB	0.35	44
SQL	0.40	26
XSS	0.45	10

2.3 Machine learning-based classification algorithms

Classification methods are used to predict the class of an object instance based on a feature vector. Machine learning-based classification algorithms build models that can learn from labelled data sets and use these to predict the class of new, unseen data points. In this study, I used five different classification algorithms:

- Logistic Regression [18]
- Naive Bayes [24] [25]
- Decision Tree [19] [20]
- Random Forest [21]
- Support Vector Machine (SVM) [22]

The five classifiers were trained and tested using the Orange 3.34 program, an open source data visualization, machine learning and data mining toolkit. The goal was to train classification algorithms based on the training data set. Each record in the data set has a label (classification label) that describes the correct class. To evaluate the performance of the classification algorithms, the metrics are based on the number of occurrence of the cases shown in Table 5.

Table 5. assessment criteria for classifiers

Tag property value	Classification value	Prediction case
0	0	TN
0	1	FP
1	0	FN
1	1	TP

All classifiers were evaluated based on the training and test samples using four measures of classification: accuracy (Accuracy), precision (Precision), coverage (Recall) and F-value (F1), which can be calculated using the following formulae:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}, \tag{1}$$

$$\mathrm{Pr}ecision = \frac{TP}{TP + FP}, \tag{2}$$

$$\mathrm{Re}call = \frac{TP}{TP + FN}, \tag{3}$$

$$F1 = \frac{2(\mathrm{Pr}ecision \cdot \mathrm{Re}call)}{\mathrm{Pr}ecision + \mathrm{Re}call}. \tag{4}$$

Based on the performance of the classifiers, I identified a threshold for each data set where I achieved the best classification performance with the smallest number of features. For each data set, the smallest threshold is 0.35. Based on this, I examined the performance of the classifiers with the feature sets identified for the five data sets at thresholds between 0.35 and 0.55. In order to find the best performing classifier with the least number of features, I took the best of the arithmetic averages of the performance scores (Accuracy, Precision, Recall, F1) obtained by examining the training and test sets for each classifier.

Results:

For each data set, I have identified the threshold at which I could achieve a good classification result with the smallest number of features. For each type of attack, I defined a separate list (see Table 6) of relevant features for each threshold. I have plotted each feature by its row number. Each row of the table contains the characteristics whose score was greater than or equal to the threshold value given in its second cell. In this way, the order of importance of the features can be used to identify attacks more efficiently.

Table 6. Best performing classifiers, the corresponding thresholds, the number of selected features and the ID numbers of the selected features in the case of each attack type

Data set	**Threshold value**	**Best classifier**	**Number of features**	**Features serial number**
FTP	0.35	Random forest	8	*02,17,19,35,00,44,56,59*
SSH	0.35	Random forest	7	*00,02,17,19,57,56,59*
WEB	0.35	Decision Tree	44	*16,20,10,49,66,67,35,38,56,64,34,27,07,09,11, 14,15,50,25,60,62,02,17,19,37,63,06,33,55,18, 58,04, 05,53,54,03,21,22,23,24,52,32,65,57*
SQL	0.40	Random forest	26	*05,26,53,56,25,02,17,19,35,16,18,27,28,34, 06,23,30,55,29,21,22,24,57,37,11,14*
XSS	0.45	Decision Tree	10	*37,56,33,32,03,11,52,04,54,58*

2.4 Weighted average based ensemble method

In the previous study, the cases where the trained classifiers showed poor performance encouraged me to further investigate the weighted averaging approach. Weighted ranking is a widely used approach for evaluating samples that allows differential evaluation of individual components based on their significance, importance, strength or any other criterion mentioned as a weight. Taking into account the contribution of several feature ranking methods, the weighted average of the feature scores was calculated using (5). This equation gives an overall assessment score that reflects the combined assessment.

$$R_{WA} = \frac{R_{IG} \cdot w_{IG} + R_{GR} \cdot w_{GR} + R_{SU} \cdot w_{SU} + R_{\chi^2} \cdot w_{\chi^2} + R_{\mathrm{Re}} \cdot w_{\mathrm{Re}} + R_{AN} \cdot w_{AN}}{w_{IG} + w_{GR} + w_{SU} + w_{\chi^2} + w_{\mathrm{Re}} + w_{AN}}, \qquad (5)$$

where R_{WA} is the feature score calculated by the ensemble method, R_{IG}, R_{GR}, R_{SU}, $R_{\chi 2}$, R_{Re}, R_{AN} are the normalized feature scores obtained by the individual feature ranking methods included in the ensemble, while w_{IG}, w_{GR}, w_{SU}, $w_{\chi 2}$, w_{Re}, w_{AN} are the weights associated with these methods.

Determining the optimal combination of weights is a challenging task, as it takes considerable time to evaluate the different collections of features resulting from the calculation of scores. It is therefore necessary to optimize the weights with a minimum number of trials.

This realization led to the use of the design of experiments (DoE) technique known as the Taguchi method. This approach, developed by Genichi Taguchi in the 1950s, was originally aimed at quality management and design in the manufacturing industry [23]. To determine the optimal parameter setting, the Taguchi method uses the concept of "parameter design". In this approach, process variables are assigned to predefined ranges of values, tested and optimized. The research involves testing six independent variables – the six weights –, each at two levels. Therefore, the orthogonal design $L_8 2^7$ was used. To facilitate a better exploration of the weight search space with minimal experiments, I assigned weight values of 0.0233 and 0.2336 to the two levels of weight variables (called factors in DoE) in the selected DoE design (see Figure 7). The rationale behind this choice was that I used values that were significantly far apart.

Table 7. Specified weight values

	WIG	WGR	WSU	WKhi	WRe	WAN
1	0.023256	0.023256	0.023256	0.023256	0.023256	0.023256
2	0.023256	0.023256	0.023256	0.232558	0.232558	0.232558
3	0.023256	0.232558	0.232558	0.023256	0.023256	0.232558
4	0.023256	0.232558	0.232558	0.232558	0.232558	0.023256
5	0.232558	0.023256	0.232558	0.023256	0.232558	0.023256
6	0.232558	0.023256	0.232558	0.232558	0.023256	0.232558
7	0.232558	0.232558	0.023256	0.023256	0.232558	0.232558
8	0.232558	0.232558	0.023256	0.232558	0.023256	0.023256

In particular, I have focused attention on cases where the previous test using arithmetic averages did not give satisfactory results. Here I had two objectives:

1. either to identify feature sets with fewer features while maintaining the original classification performance, or

2. to find feature sets that can improve classification performance by using classification Accuracy, Precision, Recall and F-value (F1) as performance measures.

Results:

For each data set, the defined reduced set of characteristics is shown in Table 8.

Table 8. Group of weighted average characteristics

Data set	Group of weighted average characteristics
FTP	*19, 02, 17, 56, 59*
SSH	*33, 32, 00, 56, 57, 59*
WEB	*32, 56, 07, 50, 09, 11, 65, 14, 37, 53, 05, 58, 57*
XSS	*57, 56*
SQL	*56, 43, 47, 57, 37, 11, 14*

2.5 Classification using CatBoost algorithm

The previously presented and used Naive Bayes, Decision Tree, Random Forest, Logistic Regression and SVM are classification and modelling methods that have long been part of the machine learning and data mining toolbox. As a continuation of my research, I wanted to investigate whether CatBoost, an implementation of the Gradient Boost approach that has been successfully applied in many fields recently, can provide classification results that are close to or better than those of traditional classifiers.

For each type of attack, I used the same training and test data sets with the CatBoost classifier, as well as features that I had previously identified by summing the scores of each feature with a weighted average.

Results:

For each data set, taking the arithmetic mean of the Accuracy, Precision, Recall, F1 performance values obtained by examining the training and test sets, I determined an average classification performance number, a number between 0-1, where 1 represents the best performance. This resulted in a comparison (see Table 9) between the Naive Bayes, Logistic Regression, Rank Vector Machine, Decision Tree, Random Forest classification algorithms and the CatBoost algorithm.

Table 9. Catboost comparison

Data set	Number of features	Classifier	Average grading performance
FTP	5	Naive Bayes	0.9942
		Logistic Regression	0.9984
		Decision Tree	1.0000
		Support vector machine	0.9998
		Random Forest	1.0000
		CatBoost	**1.0000**
SSH	6	Naive Bayes	0.9999
		Logistic Regression	0.9940
		Decision Tree	1.0000
		Support vector machine	0.9999
		Random Forest	1.0000
		CatBoost	**1.0000**
SQL	7	Naive Bayes	0.2499
		Logistic Regression	0.5252
		Decision Tree	0.9826
		Support vector machine	0.7323
		Random Forest	0.9913
		CatBoost	**0.9694**
XSS	2	Naive Bayes	0.2498
		Logistic Regression	0.2498
		Decision Tree	0.9657
		Support vector machine	0.3183
		Random Forest	0.9697
		CatBoost	**0.9197**
WEB	13	Naive Bayes	0.5017
		Logistic Regression	0.2494
		Decision Tree	0.9363
		Support vector machine	0.2071
		Random Forest	0.8994
		CatBoost	**0.8994**

3 Summary of recent scientific results

In the initial stage of my research, I investigated the basic features and functions of different IDS systems [S15] [S11]. Then I turned my attention to anomaly-based IDS systems, with a particular focus on the training process of their classification module. This process typically involves the use of large data samples containing both benign and malicious traffic data. During my research, I examined several data sets and finally found a suitable one (CSE-CIC-IDS2018 on AWS) that not only met the criteria, but was also fresh, making it ideal for training.

After selecting the data set and performing a number of preprocessing steps, my investigation focused on feature selection methods. Here, my primary result was the ranking of the features based on the arithmetic mean of the normalized feature scores obtained by the six different methods (see Thesis 1). My further research focused on feature selection, with the aim of defining thresholds for the scores obtained by the arithmetic mean-based ensemble method. The aim was to define a relevant set of features that would provide sufficient information for the classification module (see Thesis 2).

I then proceeded with research under the assumption that using a multi-factor method based on a weighted average of the scores for each feature could potentially improve classification performance, or at least reduce the number of features required. In order to verify this hypothesis, I used a Taguchi-type experimental design to keep the number of required trials low. The experimental results confirmed the hypothesis (see Thesis 3).

I continued my research with the hypothesis that the classification performance achieved by the five well-known classification algorithms used previously can be surpassed, or at least approached, by using the relatively new CatBoost algorithm. Experimental results confirmed this hypothesis (see Thesis 4).

THESIS 1

Using a data sample created for training IDS systems, I developed a method for ranking the importance of features for identification of attacks using the mean of the normalized scores calculated with the methods Information Gain, Gain Ratio, Symmetric Uncertainty, Relief, Chi-square test and Analysis of Variance methods.

My publication related to my thesis is [S3].

THESIS 2

I have defined thresholds for the averaged feature scores to identify the minimum set of features that are enough for the training of good performing classifiers.

My publication related to my thesis is [S3].

THESIS 3

I have shown that using the weighted mean of the feature scores produced by different methods can produce the same or better classification results with a reduced number of features considered.

My publication related to my thesis is [S4].

THESIS 4

I have demonstrated that by employing the CatBoost algorithm, classification outcomes are at least as effective as those achieved with the Logistic Regression, Naive Bayes, Support Vector Machine, Decision Tree, and Random Forest classifier models for specific network attack scenarios.

My publication related to my thesis is [S5].

4 Exploitability of the achieved results

By examining network communication (including normal and attack cases) using the weighted ensemble feature selection method, the features in the defined feature sets can be used to configure actual data and information for the sensors of an IDS system for certain attack types. From an initial set of 80 features in the network communication data, only some subgroups of the features in Table 10 need to be considered to achieve a good classification result for an IDS to detect the appropriate attacks.

Table 10. Properties of the defined relevant characteristics

Serial number	Name	Operational meaning of the features
00	*Dst Port*	The destination port to which data packets are sent.
02	*Flow Duration*	The duration of the data stream between the first and the last data packet.
05	*TotLen Fwd Pkts*	Total size of all forward (to source) data packets.
07	*Fwd Pkt Len Max*	The size of the longest forward data packet.
09	*Fwd Pkt Len Mean*	Average size of forward data packets.
11	*Bwd Pkt Len Max*	The size of the longest of the backward data packets.
14	*Bwd Pkt Len Std*	Dispersion of the size of the backward data packet.
17	*Flow IAT Mean*	The average length of the intervals between data streams.
19	*Flow IAT Max*	Maximum length of intervals between data streams.
32	*Fwd Header Len*	Header size of forward data packets.
33	*Bwd Header Len*	The size of the header for reverse data packets.
37	*Pkt Len Max*	The largest data packet size of all packets.
43	*RST Flag Cnt*	Number of data packets with the RST flag.
47	*ECE Flag Cnt*	Number of data packages with the ECE flag.
50	*Fwd Seg Size Avg*	Average segment size of forward data packets.
53	*Subflow Fwd Byts*	The total size of forward data in subcommunication flows.
56	*Init Fwd Win Byts*	The forward initial window size of the TCP connection.
57	*Init Bwd Win Byts*	The backward initial window size in the TCP connection.
58	*Fwd Act Data Pkts*	The number of forward effective data packets.
59	*Fwd Seg Size Min*	Minimum segment size for forward data packets.
65	*Idle Std*	The spread of periods of inactivity between periods in the data stream.

5 Further research directions

I intend to continue my research on IDS systems towards the development of solutions based on data fusion. Here data fusion refers to the process of combining and analyzing data from multiple sources or sensors in order to increase the accuracy of network intrusion detection. This includes integrating information from firewalls, intrusion detection systems (IDS), log files, network traffic data and other relevant sources.

Data fusion aims to take advantage of different data sources and improve the detection capabilities of intrusion detection systems. Patterns and anomalies that may not be detected by individual sensors or detection methods are likely to be more easily identified by combining multiple data streams, which may result in more reliable intrusion detection.

As a first step, a Computer Science Data Lab is being prepared at my workplace, the John von Neumann University in Kecskemét, Hungary, which will include an Expertise and Cybersecurity Lab, where a complete IT lab environment will be set up, similar to the Canadian lab.

A complete IT infrastructure would be modelled, with normal and attack communications. By designing such a system and implementing data fusion, a new proprietary sample data set could be created, which could be freely used by IDS researchers. Using this data set, I plan to investigate other solutions in addition to the classifiers presented in this thesis.

6 Summary

During the initial phase of my investigation, I delved into the fundamental characteristics and functionalities of various IDS systems [S15] [S11]. Subsequently, I directed my attention towards Anomaly-based IDS systems, specifically focusing on the training process of their classification module. Typically, this process involves utilizing big data samples that describe both benign and malicious traffic scenarios. Throughout my research, I explored multiple data sets, eventually discovering a suitable one (CSE-CIC-IDS2018 on AWS) that not only met the criteria but was also recent, making it ideal for training purposes.

Once the data set was chosen and several preprocessing steps were executed, my investigation centered around feature selection methods. The primary outcome here was the ranking of features based on the arithmetic mean of normalized feature scores obtained from six distinct methods (refer to Thesis statement 1). Moving forward, my research targeted feature selection, aiming to establish threshold values for the scores obtained through the arithmetic mean based ensemble method. The objective was to identify a relevant set of features that would furnish sufficient information for the classifier module (refer to Thesis statement 2).

Subsequently, I pursued my investigation with the assumption that utilizing an ensemble method based on weighted aggregation of individual feature scores could potentially enhance classification performance or, at the very least, reduce the number of required features. To validate this hypothesis, I adopted a Taguchi-type DoE design, conducting a low number of trials. The experimental results confirmed the hypothesis (refer to Thesis statement 3).

Continuing my research, I operated under the hypothesis that the classification performance achieved by the five well-known classification algorithms used previously could be surpassed, or at least matched, by employing the relatively new CatBoost algorithm. The experimental results also confirmed this hypothesis (refer to Thesis statement 4).

References

[1] R. B. Basnet, R. Shash, C. Johnson, L. Walgren, and T. Doleck, ‘Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks’, *J. Internet Serv. Inf. Secur*, no. 9, pp. 1–17, 2019.

[2] Z. J. Viharos, K. B. Kis, Á. Fodor, and M. I. Büki, ‘Adaptive, hybrid feature selection (AHFS)’, *Pattern Recognition*, vol. 116, p. 107932, 2021, doi: 10.1016/j.patcog.2021.107932.

[3] K. Muhi and Z. C. Johanyák, ‘Dimensionality reduction methods used in Machine Learning’, *M\Huszaki Tudományos Közlemények*, vol. 13, no. 1, pp. 148–151, 2020, doi: 10.33894/mtk-2020.13.27.

[4] T. Dobján and E. D. Antal, ‘Modern feature extraction methods and learning algorithms in the field of industrial acoustic signal processing’, in *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, IEEE, 2017, pp. 000065–000070. doi: 10.1109/sisy.2017.8080589.

[5] N. S. Chauhan, ‘Decision Tree Algorithm—Explained’. 2020. [Online]. Available: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html/https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html/

[6] V. Bolón-Canedo and A. Alonso-Betanzos, ‘Ensembles for feature selection: A review and future trends’, *Information Fusion*, vol. 52, pp. 1–12, Dec. 2019, doi: 10.1016/j.inffus.2018.11.008.

[7] G. Ayyappan, D. C. Nalini, and D. A. Kumaravel, ‘Efficient mining for social networks using Information Gain Ratio based on Academic data set’, *International Journal of Civil Engineering and Technology*, vol. 8, no. 1, 2017.

[8] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature selection for high-dimensional data*. Springer, 2015. doi: 10.1007/978-3-319-21858-8.

[9] A. G. Karegowda, A. S. Manjunath, and M. A. Jayaram, ‘COMPARATIVE STUDY OF ATTRIBUTE SELECTION USING GAIN RATIO AND CORRELATION BASED FEATURE SELECTION’.

[10] R. P. Priyadarsini, M. Valarmathi, and S. Sivakumari, ‘Gain ratio based feature selection method for privacy preservation’, *ICTACT Journal on soft computing*, vol. 1, no. 4, pp. 201–205, 2011, doi: 10.21917/ijsc.2011.0031.

[11] S. J. Pasha and E. S. Mohamed, ‘Ensemble Gain Ratio Feature Selection (EGFS) Model with Machine Learning and Data Mining Algorithms for Disease Risk Prediction’, in *2020 International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India: IEEE, Feb. 2020, pp. 590–596. doi: 10.1109/ICICT48043.2020.9112406.

[12] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, ‘Relief-based feature selection: Introduction and review’, *Journal of biomedical informatics*, vol. 85, pp. 189–203, 2018, doi: 10.1016/j.jbi.2018.07.014.

[13] B. Singh, N. Kushwaha, O. P. Vyas, and others, ‘A feature subset selection technique for high dimensional data using symmetric uncertainty’, *Journal of Data Analysis and Information Processing*, vol. 2, no. 04, p. 95, 2014, doi: 10.4236/jdaip.2014.24012.

[14] S. Bakhshandeh, R. Azmi, and M. Teshnehlab, ‘Symmetric uncertainty class-feature association map for feature selection in microarray data set’, *Int. J. Mach. Learn. & Cyber.*, vol. 11, no. 1, pp. 15–32, Jan. 2020, doi: 10.1007/s13042-019-00932-7.

[15] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, *Feature selection for high-dimensional data*. Springer, 2015.

[16] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, ‘Feature selection and classification of microarray data using MapReduce based ANOVA and K-nearest neighbor’, *Procedia Computer Science*, vol. 54, pp. 301–310, 2015, doi: 10.1016/j.procs.2015.06.035.

[17] M. Héder *et al.*, ‘The Past, Present and Future of the ELKH Cloud’, *InfTars*, vol. 22, no. 2, p. 128, Aug. 2022, doi: 10.22503/inftars.XXII.2022.2.8.

[18] M. Maalouf, ‘Logistic regression in data analysis: an overview’, *IJDATS*, vol. 3, no. 3, p. 281, 2011, doi: 10.1504/IJDATS.2011.041335.

[19] B. Charbuty and A. Abdulazeez, ‘Classification based on decision tree algorithm for machine learning’, *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, 2021, doi: 10.38094/jastt20165.

[20] R. Davidson, ‘Reliable inference for the Gini index’, *Journal of Econometrics*, vol. 150, no. 1, pp. 30–40, May 2009, doi: 10.1016/j.jeconom.2008.11.004.

[21] L. Breiman, ‘Random Forests’, *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[22] I. Steinwart and A. Christmann, *Support vector machines*, 1st ed. in Information science and statistics. New York: Springer, 2008.

[23] A. Freddi and M. Salmon, *Design Principles and Methodologies: From Conceptualization to First Prototyping with Examples and Case Studies*, 1st ed. 2019. in Springer Tracts in Mechanical Engineering. Cham: Springer International Publishing: Imprint: Springer, 2019. doi: 10.1007/978-3-319-95342-7.

[24] H. Zhang and J. Su, ‘Naive Bayesian Classifiers for Ranking’, in *Machine Learning: ECML 2004*, J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, Eds., in Lecture Notes in Computer Science, vol. 3201. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 501–512. doi: 10.1007/978-3-540-30115-8_46.

[25] F.-J. Yang, ‘An Implementation of Naive Bayes Classifier’, in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA: IEEE, Dec. 2018, pp. 301–306. doi: 10.1109/CSCI46756.2018.00065.

Author’s publications

Journal articles

S1. **László, Göcs**; Attila, Pásztor; Zsolt, Csaba Johanyák: Computer network solutions in modern industrial environment ANNALS OF FACULTY OF ENGINEERING HUNEDOARA - INTERNATIONAL JOURNAL OF ENGINEERING 10: 1 pp. 75-80., 6 p. (2022)

S2. **László, Göcs**; Zsolt, Csaba Johanyák; Péter, András Agg: Protection of Computer Laboratories in Educational Institutions ACTA TECHNICA CORVINIENSIS – BULLETIN OF ENGINEERING 9: 2 pp. 93-98., 6 p. (2016)

S3. **László, Göcs**; Zsolt, Csaba Johanyák: Identifying Relevant Features of CSE-CIC-IDS2018 Data set for the Development of an Intrusion Detection System, 2023 http://gocslaszlo.hu/phd/tezis_1_2.pdf (under publication)

S4. **L. Göcs** and Z. C. Johanyák, ‘Feature Selection with Weighted Ensemble Ranking for Improved Classification Performance on the CSE-CIC-IDS2018 Data set’, Computers, vol. 12, no. 8, p. 147, Jul. 2023, doi: 10.3390/computers12080147.

S5. **László, Göcs**; Zsolt, Csaba Johanyák: Catboost algorithm based IDS cassification module for brute force attacks", ANNALS OF FACULTY OF ENGINEERING HUNEDOARA: INTERNATIONAL JOURNAL OF ENGINEERING 01: 3 pp. 13-18.,6 p. (2023)

Conference publications

S6. **Göcs, László**; Johanyák, Zsolt Csaba: Adatbázis feldolgozása IDS rendszerek tanításához Kutatás és innováció 2021: GAMF Közlemények tanulmánykötete Kecskemét, Magyarország: Neumann János Egyetem GAMF Műszaki és Informatikai Kar (2021) pp. 401-406., 6 p.

S7. **Göcs, László**; Pásztor, Attila; Johanyák, Zsolt Csaba: Modern ipari környezet informatikai hálózati lehetőségei a rendelkezésre állás biztosítása érdekében GRADUS 8: 3 pp. 147-156., 10 p. (2021)

S8. **Göcs, László**; Johanyák, Csaba; Kovács, Szilveszter: IDS rendszerek fuzzy logikával In: Keresztes, Gábor; Kohus, Zsolt; Szabó P., Katalin; Tokody, Dániel (szerk.) Tavaszi Szél 2017 Konferencia. Nemzetközi Multidiszciplináris Konferencia: Absztraktkötet Budapest, Magyarország: Doktoranduszok Országos Szövetsége (DOSZ) (2017) 477 p. p. 311

S9. Agg, Péter András; Johanyák, Zsolt Csaba; **Göcs, László**: Szoftver által definiált hálózatok áttekintése In: Bitay, Enikő (szerk.) A XXI. Fiatal Műszakiak Tudományos Ülésszaka előadásai Kolozsvár, Románia: Erdélyi Múzeum Egyesület (EME) (2016) 452 p. pp. 57-60., 4 p.

S10. **Göcs, László**; Johanyák, Zsolt Csaba; Kovács, Szilveszter: Csapda a hálózaton GRADUS 3: 2 pp. 55-60., 6 p. (2016)

S11. **László, Göcs**; Zsolt, Csaba Johanyák; Szilveszter, Kovács: Review of Anomaly-Based IDS algorithms In: AlumniPress - AlumniPress (szerk.) TEAM 2016: Proceedings of the 8th International Scientific and Expert Conference Trnava, Szlovákia: Alumni Press (2016) 360 p. pp. 58-63., 6 p.

S12. **László, Göcs**; Zsolt, Csaba Johanyák: Virtualization in Network Administration Education In: Kucsinka, Katalin; Kiss, Alexandra; Veres, Erika (szerk.) Matematikát oktatók és kutatók nemzetközi tudományos konferenciája Beregszász, Ukrajna: II. Rákóczi Ferenc Kárpátaljai Magyar Főiskola (2016) 78 p. p. 52

S13. Agg, P; **Göcs, L**; Johanyák, Zs Cs; Borza, Z: Csomagszűrés CISCO routereken ACL-ek segítségével GRADUS 2: 2 pp. 104-111., 8 p. (2015)

S14. **Göcs, László**; Johanyák, Zsolt Csaba: Vállalati informatikai biztonság szerepe napjainkban In: Bitay, Enikő (szerk.) A XX. Fiatal Műszakiak Tudományos Ülésszaka előadásai: Proceedings of the XX-th International Sciencific Conference of Young Engineers Kolozsvár, Románia: Erdélyi Múzeum Egyesület (EME) (2015) 356 p. pp. 155-158., 4 p.

S15. **László, Göcs**; Zsolt, Csaba Johanyák: Survey on intrusion detection systems In: Prof, Aleksandar Sedmak; Zoran, Radakovic; Simon, Sedmak; Snezana, Kirin (szerk.) Proceedings of TEAM 2015: 7th International Scientific and Expert Conference of the International TEAM Society Beograd, Szerbia: University of Belgrade, Faculty of Mechanical Engineering (2015) 650 p. pp. 167-170., 4 p.

S16. Zsolt, Csaba Johanyák; Piroska, Gyöngyi Ailer; **László, Göcs**: A simple fuzzy control design for series hybrid electric vehicle In: Andrea, Ádámné Major; Lóránt, Kovács; Zsolt, Csaba Johanyák; Róbert, Pap-Szigeti (szerk.) Proceedings of TEAM 2014: 6th International Scientific and Expert Conference of the International TEAM Society Kecskemét, Magyarország: Kecskeméti Főiskola Gépipari és Automatizálási Műszaki Főiskolai Kar (2014) 499 p. pp. 159-164., 6 p.

S17. **Göcs, László**: Informatikai biztonság In: Ferencz, Árpád; Borsné, Pető Judit; Lipócziné, Csabai Sarolta; Kovács, Lóránt (szerk.) AGTEDU 2011: a Magyar Tudomány Ünnepe alkalmából rendezett 12. tudományos konferencia Kecskemét, Magyarország: Kecskeméti Főiskola (2011) 406 p. pp. 143-148., 6 p.

Other publications

S18. **Göcs, László**: Covid19 hatása az informatikai rendszerekre (2020) AGTEDU 2020, 2020. november 12., Előadás,

S19. **Göcs, László**: Adataink és az okos eszközök - kémek a lakásban? (2019) AGTEDU 2019, 2019. november 13., Előadás,

S20. **Göcs, László**; Johanyák, Zsolt Csaba: Címkézett adatbázis IDS rendszerekhez (2018) AGTEDU 2018, Kecskemét: 2018. november 15., Előadás,

S21. **Göcs, László**: A digitális világ biztonságos használata: Internet és informatikai biztonság (2018) Hírös Szabadegyetem, Kecskemét, 2018. április 11., Előadás,

S22. **László, Göcs**: Importance of passwords in IT security (2018) WCNCI 2018 (Workshop on Computer Networks and Computational Intelligence), 2018. október 16., Előadás,

S23. **Göcs, László**; Johanyák, Zsolt Csaba; Bors, Ádám: A blokklánc technológia (2017) AGTEDU 2017: Magyar Tudomány Ünnepe: 2017. november 16., Neumann János Egyetem Gazdálkodási Kar, Szolnok, Előadás, Megjelenés: Magyarország,

Educational materials

S24. **Göcs, László**: Szerveroldali megoldások Linux környezetben (Ubuntu 20.04 LTS) Kecskemét, Magyarország: Neumann János Egyetem (2021) ISBN: 9786155817915

S25. Johanyák, Zsolt Csaba; **Göcs, László**: Windows hálózati adminisztráció a gyakorlatban, 153 p. (2014)

S26. Johanyák, Zsolt Csaba; Kovács, Péter; **Göcs, László**: Linux hálózati adminisztráció a gyakorlatban, 113 p. (2013)