

Automated Computer Attack Detection in University Environment

Lukáš Švarc , Pavel Strnad 

Faculty of Informatics and Statistics, Prague University of Economics and Business, W. Churchill Sq. 1938/4, 130 67 Prague 3, Czech Republic

Corresponding author: Lukáš Švarc (lukas.svarc@vse.cz)

Abstract

Since the massive expansion of the Internet into a commercial world, the security of computer systems has become a priority. There are other areas that see an increase in the inclusion of the Internet, like national governments, hospitals, and university systems. All these systems contain highly sensitive information. In an effort to increase the security of internal data, we propose a novel method for the detection of automated computer attacks. This method was tested on a custom dataset prepared from the logs of the university information system at Prague University of Economics and Business. Two datasets were used. The first dataset contained only simple attacks, while the second one comprised the advanced attacks. The compiled and anonymized datasets were uploaded to BigML framework, where K-means, Isolation Forest and Logistic Regression algorithms were used in order to validate the proposed novel method. Our results showed that the proposed method is viable in cases where the attack volume is high and the time spacing between the actions is similar, which was verified on both tested datasets. It reached the detection rate of 93.57% in the case of simple attacks dataset, and 95.37% in the case of advanced attacks dataset. It reached similar detection rates as other algorithms used in the commercial environment. Based on this project, the proposed method can be implemented into the university information system in order to prevent these types of attacks in the future.

Keywords

Anomaly detection, Machine learning, Automated attacks, University environment.

Citation: Švarc, L., & Strnad, P. (2021). Automated Computer Attacks Detection in University Environment. *Acta Informatica Pragensia*, 10(1), 75–84. <https://doi.org/10.18267/j.aip.147>

Academic Editor: Stanislava Mildeova, University of Finance and Administration, Czech Republic

Copyright: © 2021 by the author(s). Licensee Prague University of Economics and Business, Czech Republic.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution License (CC BY 4.0).

1 Introduction

In today's business world, information is one of the most important assets of organizations and therefore needs appropriate protection and management. Information systems play a crucial part because they contain countless important data and grant access to all or part of them to various groups of users. It is important to guarantee the integrity of the internal data and security mechanisms to prevent their misuse. However, in reality, it has been showed that such mechanisms for enforcing the organizational security policies are often used inadequately (Chung et al., 2000). This means that nonintrusive attacks often pass through unnoticed.

This matter is even more important in specific environments. The university environment is very different from the typical business environment for a couple of reasons. First of all, the users connecting to the computer network are very diverse, as every year thousands of students start to study, while many others are leaving. Second, there is a high number of external specialists who collaborate with the university but are not regular employees. Third, the typical user behavior is very different to the behavior in the business environment. Students and regular employees use network resources and study information systems very differently (Stamp, 2018).

In the university environment, users do not attend trainings before using the information system as they do in the business environment, which often leads to unintentionally incorrect usage of the system. Typical threat detection systems can detect this behavior as an attack, which might be a false positive in the university environment.

Automated attacks are those types of attacks which happen automatically or periodically, without users' interference. The most typical examples of these attacks are various scripts which can download user information from a website, going from one user profile to another in a rapid succession. Another example is a script which periodically checks for free examination slots and then registers the person in it if there is one available.

To make things more difficult, using a script or a specific code does not have to mean it is an automated attack. In the university environment, there are several cases where IT administrators use such tools in order to work around a functionality which is not implemented in the system. For example, if there is an entrance examination at the university, there are several thousands of students coming to the computer classrooms in order to make a test. There is a need to disable Internet access in these classrooms, but the information system allows disabling the Internet connection only in one classroom at a time. In order to speed up this process and disable the Internet connection in dozens of classrooms, there is a script which simulates user input and disables the classrooms' connection one by one in a fast succession.

2 Related Work

Attack detection in the Internet of Things (IoT) devices is an important issue. Compared to traditional computer networks, IoT networks have several unique characteristics, which raises the difficulty of attack detection. First, heterogeneous protocols, platforms, software, and hardware are exposed to various vulnerabilities. Second, apart from the traditional high-rate attacks, low-rate attacks are also used a lot by IoT attackers to confuse legitimate and malicious traffic. These low-rate attacks are difficult to detect and can continue to operate in the network for a long period of time. Finally, the attackers are improving day by day, are cleverer and can dynamically adjust their attack methods based on system responses in order to avoid detection, making it much more challenging for security experts to specify a common pattern to identify the attack. Gu et al. (2020) conducted widespread experiments with attacks on a real IoT dataset and demonstrated the effectiveness of their IoT attack detection framework.

Recognizing human attacks from bot attacks was a focus of Udhani et al. (2019). Humans have some specific behavioral characteristics and limits, which can be identified and used to differentiate human

interaction from automated attacks. The network log data collected from their Honeypot uncovered such characteristics, which are not otherwise observable. Their paper analyzed a Honeypot deployed for over a year pretending to be a SSH server in order to identify those human behavioral characteristics that can essentially differentiate an automated attack from human attacks.

Machine learning based social engineering attack detection in online environments are gaining in popularity. These types of attacks are one of the best-known and easiest to perform attacks in the security domain. Lansley et al. (2020) showed that most attacks against computer systems originated from social engineering methods. They considered the importance of technical fields such as machine learning and cybersecurity and developed an algorithm that detects social engineering attacks based on natural language processing and neural networks.

Lin et al. (2015) discussed advanced detection approaches created by combining multiple learning techniques. They present a new approach for anomaly detection called Cluster Center and Nearest Neighbor, which shows better results than any previous clustering anomaly detection methods.

Another research focused on bot detection techniques using unsupervised machine learning technologies. Wu et al. (2016) compared common traffic data to bot traffic data. In their study they tested several different clustering algorithms. The best algorithm for each case was chosen and refined and the results were compared to all datasets with a high success rate of distinguishing bots from real users.

K-Nearest Neighbor is considered one of the easiest classification algorithms. Tsigkritis et al. (2018) focused on an implementation of this algorithm into a logging database of PCCW Global company and observed the network traffic. During their experiment they generated alerts about suspicious IP traffic. Although detailed information about the attacks was not available to them, their algorithm successfully detected security incidents.

Robust Random Cut Forest Based Anomaly Detection based on Random Forest was presented by Guha et al. (2016). If the data are correct, distance is critically important for anomaly detection as well as for calculations. During their research they found that the algorithm achieves good results in detected events and false alarms as well. They tested their algorithm on data from New York taxi drivers and compared the results with Isolation Forest, where they reduced the time needed for detection from eleven hours to seven.

Joshi et al. (2005) tested Hidden Markov Model on the popular KDDCUP'99 dataset. This dataset is quite similar to the dataset presented in this article, therefore these results can be used for a comparison. The researchers used five out of forty-one features to train their classification algorithm and reached 79% detection rate.

Another KDDCUP'99 dataset research was performed by Seong et al. (2005). They tested Fusions of Genetic Algorithm and Support Vector Machines for Anomaly Detection. Their goal was to optimize feature selection in order to minimize the amounts of features and maximize the detection rates. They accomplished their task, as they have reached 98% detection rate.

Clustering techniques were part of Chandrashekhar et al. (2012) research. They experimented with them on KDDCUP'99 dataset. They evaluated four clustering techniques: K-means clustering, Fuzzy c-means clustering, Mountain clustering, and Subtractive clustering. Results showed that accuracy of K-means was 91.02%, Fuzzy c-means clustering was 91.89%, Mountain clustering was 75% and Subtractive clustering was 78.27%. Considering the computational time, K-means algorithm was ranked the best among tested algorithms.

3 Research methods

As mentioned above, there are several different ways to deal with automated attacks. The purpose of our research is to verify that automated attacks in the university environment can be detected adequately. To do so, two core elements are needed. First of all, a public dataset to ensure the validity, integrity and repeatability of our research is essential. Second, viable approaches that would apply machine learning algorithms and achieve high percentage of attack detection are important as well. An appropriate combination of these two elements is the key to accomplishing this task as successfully as possible.

3.1 Dataset

A high-quality, well-defined and easy to follow dataset is an important key for successful research. With this in mind, we knew it was necessary to prepare our own dataset which could be used not only for this article but by other researchers in the future as well. Although there are several usable datasets already available, they are usually intended for intrusion detection purposes, such as KDD or NSL-KDD dataset (Tavallae et al., 2009). They focus on network traffic and network attacks and lack the granularity of information system logs. We did not find a good enough public dataset for information system actions which mark automated attacks.

In order to proceed with our research, we prepared a dataset containing actions in a real university information system. This dataset was created from application-level logs, which contained information showed in Table 1 below about all the actions performed within the system. The real information in this dataset was anonymized so that it could be published in the future, which would enable other researchers to validate our results or perform their own experiments.

Table 1. *University Dataset Structure.*

Name	Type	Description
id_norm	Numeric	ID of an action in the system
user	Categorical	Username
timestamp_norm	Numeric	Time of an action
url	Categorical	Action
ip	Categorical	IP address
parameters_hash	Categorical	Parameter of an action
asn	Numeric	Autonomous System Number
bgp_prefix	Numeric	BGP prefix
as_name	Text	Name of the Autonomous System
net_name	Categorical	Name of the network
country_code	Categorical	Code of a country
attack	Categorical	Value of attack
time_from_last_action	Numeric	Time from last action

id_norm is the id of an action. Each action increases this number by one.

user is the username of a person using the system. Although it is anonymized, the same value can be observed for all actions made by the same user.

timestamp_norm is the value which shows the time of an action in the information system.

url is an action which was performed. Examples of these actions are submitting assignments, checking lecture sheets, registering for examinations, viewing timetables and many others.

ip is the IP address from which an action was performed.

parameters_hash is the hashed value of a specific parameter of an action. For example, if a user views the same examination date, this hash has the same value every time.

asn is the numeric value of the Autonomous System Number of Internet providers all around the world. It is publicly available from several websites. A simple script was implemented to find ASNs from IP addresses.

bgp_prefix is the BGP prefix value directly affected by Autonomous System Number.

as_name is the name of the Autonomous System also directly affected by Autonomous System Number.

net_name is another artificially created value which is specific for the university environment and shows whether an action was performed from the university Wi-Fi, the university wired student network, the university wired employees network or from the global Internet.

country_code is the code of a country belonging to an IP address. An easy script to find which IP address belongs to which country was implemented to obtain this value.

attack is a parameter which indicates whether a specific action belongs to an attack or normal traffic. The actions that were part of the attacks were manually tagged.

time_from_last_action is a crucial value which was not the part of the information system logs, but we calculated it during our research. It measures the time between a particular action and the previous one for each individual user and plays a huge role in the detection of automated attacks.

All values apart from *asn*, *bgp_prefix*, *as_name*, *net_name* and *country_code* were anonymized in order to maintain the confidentiality of information system users and to allow future publication.

3.2 K-means

K-means is one of the most popular and well-known unsupervised clustering algorithms. It begins with a random assignment of all data points to K number of clusters. After that, the centers of the clusters are computed and the data points are assigned to clusters with the closest centers. This process is repeated until the cluster centers do not change significantly. When the cluster assignment is finished, the score is calculated by the mean distance of a data point to cluster centers. K-means algorithm belongs to a group of very popular data mining algorithms which can be used to detect anomalies.

It is used to detect anomalies in user behavior, as well as suspicious behavior in network traffic. The architecture of clusters is updated with each iteration of the algorithm. When the clusters are updated, the data points are moved from one cluster to another. The values of the centroids change while the update of the clusters is in progress. This change reflects the current cluster datapoints. Once these iterations are finished and there are no more changes to any individual cluster, the training of the K-means algorithm is complete, and the algorithm is ready for the classification of traffic (Zhengxi, 2001).

3.3 Isolation Forest

Isolation Forest is an unsupervised algorithm that works by building an ensemble of iTrees for any given dataset. Instances with short average path lengths on the iTrees are considered anomalies. This method has only two variables: the number of trees and the sub-sampling size. Isolation Forest detection performs very well with a small number of trees. It requires a small sub-sampling size to achieve a high detection

performance with high efficiency. As a result, Isolation Forest has a linear time complexity with a low constant and a low memory requirement, which is ideal for extensive datasets.

Fundamentally, Isolation Forest is an accurate and effective anomaly detection algorithm, which works best for large databases. Its ability to handle high volume databases is very suitable for real environment applications (Liu et al., 2008).

3.4 Logistic Regression

Logistic Regression is an example of supervised algorithms. Unlike linear regression, it works with binomial response variables. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

The biggest advantage is that it can use continuous explanatory variables and it easily handles more than two explanatory variables simultaneously. This characteristic is essential when the impact of various explanatory variables on the response variable is a subject of interest. While this algorithm is quite simple and relatively fast compared to other supervised methods, its accuracy suffers. Because of its simplicity it runs into problems while looking for complex relationships between variables (Sperandei, 2014).

4 Solutions and Results

A novel method to detect simple and advanced attacks in the University Dataset was proposed by the authors. Using cluster analysis, clusters by *user*, *asn* and *time_from_last_action* were created. These features are the most important in the detection of automated attacks, because in these clusters it is possible to distinguish valid actions from batched actions which have the same time spacing between each action. Also, the number of automated actions is much higher. Logistic regression was used to distinguish which of these clusters contain attacks. This process can be applied to any automated attack which occurs in high volumes with same time spacing. This method was verified in two parts of the dataset with two different attacks.

The first dataset *log_simple_automated_attack* includes gray software, used by students to find if there is a free slot in a specific course. The software attacks the system by using its own script that is supposed to enroll a student in the preferred course date when all others are probably already full. The script repeatedly downloaded course capacity in order to find a free slot on the preferred date. One simple action was repeated at the same time interval. The dataset contains real traffic from the university information system captured over 15 days. It contains 4,070,564 actions from 17,223 unique users. There were 835 unique system actions from 64,947 unique IP addresses, out of which 61,225 originated from a non-university computer network. In total, 21,548 actions were tagged as an attack. 0.53% of all the traffic was considered to be attacks.

The second dataset *log_advanced_automated_attack* includes an attack using a script to download information about students. It used a compromised student account that downloaded sensitive information about other users and was performed by a script with the same time interval between each action. It was classified as an advanced attack as the script performed different types of system actions and it would be harder to find by simple machine-learning algorithms focused on repeated actions over time. It contains real traffic from the university information system captured over 15 days. It contains 4,632,385 actions from 16,167 unique users. There were 808 unique system actions from 66,971 unique IP addresses, out of which 58,637 originated from a non-university computer network. In total, 3,362 actions were tagged as an attack. 0.07% of all the traffic was considered to be attacks.

Both datasets were uploaded to BigML framework, which is a robust cloud-based platform that offers Machine Learning as a Service. Its Pro version, which we used, supports up to 4 GB dataset size, which was enough for our research because *log_simple_automated_attack* is 688.6 MB and *log_advanced_automated_attack* is 766.8 MB.

Considering simple attacks first among all the dataset values, the highest weight was given to *user*, *asn*, and *time_from_last_action*. K-means algorithm with K value set to 300 was used to create clusters, 300 being the highest possible value in the framework. Using a higher K value might provide even better results, but it was limited by BigML platform.

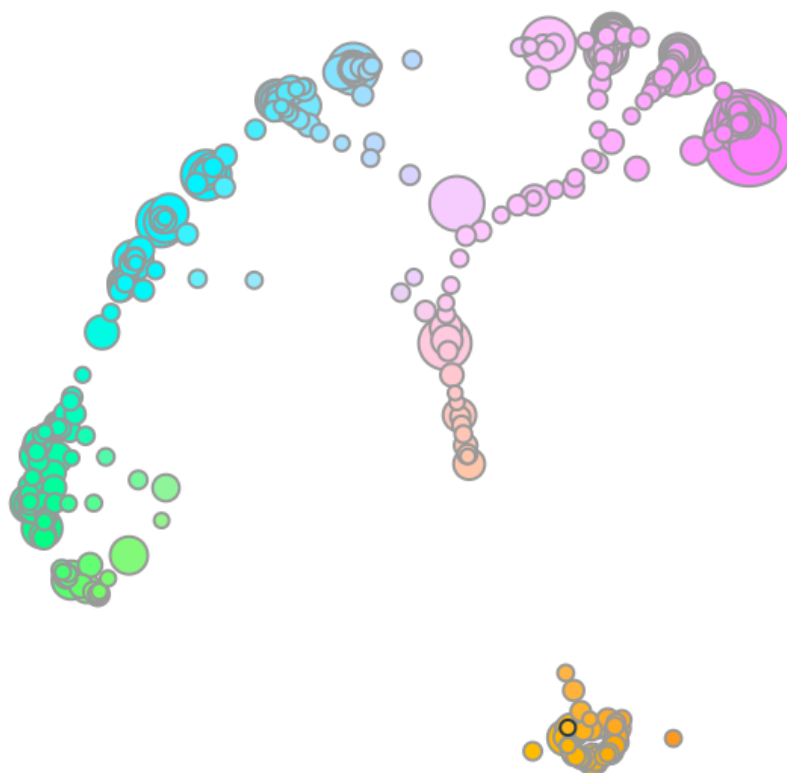


Figure 1. *K-means Generated Clusters.*

In Figure 1, all generated clusters can be observed. The logistic regression algorithm was run to determine which of these clusters contain automated attacks. A cluster, shown in Figure 1 as a black circle, was found and it contained 93.57% instances of automated attacks. We experimented with different K values during this research and observed the percentage of detected attacks. When K value was set to 50, the cluster contained only 82.66% instances of automated attacks. When K value was set to 100, it contained 86.37% instances and when K value was set to 200 it contained 90.25% instances. The impact of K value on the percentage of detected automated attacks is shown in Figure 2.

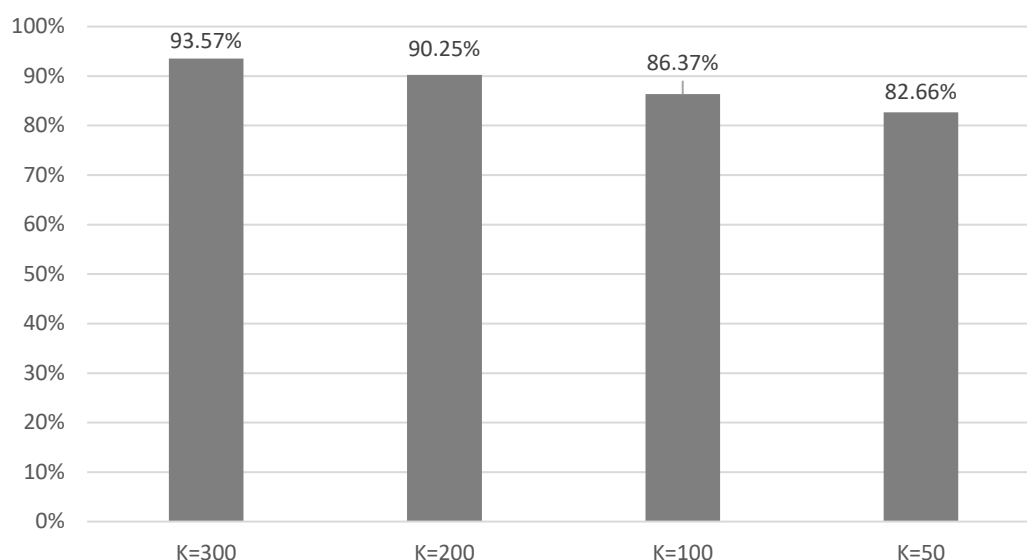


Figure 2. Automated Attacks Detected Based on K Value in K-means Algorithm.

For the second dataset, `log_advanced_automated_attack_times`, iForest algorithm was used and it detected 95.37% instances of automated attacks. We experimented with different values of sampling size parameter and reached the best ratio with sampling size 1,024.

5 Discussion

A series of experiments was performed to validate the novel method in BigML framework using K-means, Isolation Forest and Logistic Regression algorithms. We prepared a unique dataset from the university information system at Prague University of Economics and Business with tagged attack actions and anonymized it to enable future publication.

The results showed that the proposed novel method is successful in the detection of automated attacks due to the fact that these attacks are characterized by a high volume of attack data and similar time spacing between each action. Udhani's research was especially helpful in recognizing the specific patterns distinguishing human and bot attacks (Udhani, 2019). Simpler approaches could be used, such as counting HTTP request per second per IP address or user, but this can lead to a lot of false positive cases. For example, if enrollment for courses is about to start, many users interact with the system manually by pressing F5 over and over again. With our method, we try to differentiate between this behavior and automated software behavior.

Compared to the results of other researchers such as Chandrashekhar, who reached the detection rate with K-means algorithm of 91.02% (Chandrashekhar et al., 2012), Joshi, who reached 79% detection rate with Hidden Markov Model (Joshi et al., 2005), or Seong, who reached the detection rate of 98% with Fusions of Genetic Algorithm and Support Vector Machines (Seong et al., 2005), our results are comparable, even though different datasets were used, which needs to be taken into consideration.

Based on this research, we would like to propose an implantation of our method into the university information system in order to suppress specific simple and advanced automated attacks.

In the future, we would like to extend our research and try to create a method that will enable detecting even more sophisticated types of attacks. Right now, more sophisticated types of attack, for example hack attempts or unauthorized actions are not within the scope of the proposed method. In order to do so, we would need to improve the dataset we prepared and tag more complicated attacks manually.

Another major contribution for the scientific world would be the release of the anonymized datasets used during our research, so that other scientist could experiment with their algorithms and compare their results to ours, including the verification of our proposed method.

Additional Information and Declarations

Funding: This work was funded by Internal Grant Agency of Prague University of Economics and Business, project no. F4/37/2021.


Conflict of Interests: The authors are employees of Prague University of Economics and Business.

Author Contributions: L.S.: Conceptualization, Formal Analysis, Investigation, Methodology, Supervision, Visualization, Writing – original draft. P.S.: Data curation, Funding acquisition, Project administration, Resources, Software, Validation, Writing – review & editing.

Data Availability: The data that support the findings of this study are available from the corresponding author.

References

- Chandrashekhar, A.M. & Raghuveer K. (2012). Performance evaluation of data clustering techniques using KDD Cup-99 Intrusion detection data set. *International Journal of Information & Network Security*, 1(4), 294–305.
- Chung, C., Yip, M. G. & Levitt, K. (2000). DEMIDS: A Misuse Detection System for Database Systems. In *Integrity and Internal Control in Information Systems: Strategic Views on the Need for Control* (pp. 159–178). Springer. https://doi.org/10.1007/978-0-387-35501-6_12
- Gu, T., Allaukik A., Hao F., Huanle Z., Debraj B. & Prasant M. (2020). Towards Learning-Automation IoT Attack Detection through Reinforcement Learning. ArXiv:2006.15826 [Cs], June 29, 2020. <http://arxiv.org/abs/2006.15826>
- Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016). Robust Random Cut Forest Based Anomaly Detection on Streams. In *Proceedings of the 33rd International Conference on Machine Learning*. JMLR.
- Joshi, S. S., & Phoha V. V. (2005). Investigating Hidden Markov Models Capabilities in Anomaly Detection. In *Proceedings of the 43rd Annual Southeast Regional Conference* (pp. 98–103). ACM. <https://doi.org/10.1145/1167350.1167387>
- Lansley, M., Mouton F., Kapetanakis S., & Polatidis N. (2020). SEAD++: Social Engineering Attack Detection in Online Environments Using Machine Learning. *Journal of Information and Telecommunication*, 4(3), 346–362. <https://doi.org/10.1080/24751839.2020.1747001>
- Lin, W.-C., Ke, S.-W., & Tsai, C.-F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, 78, 13–21. <https://doi.org/10.1016/j.knosys.2015.01.009>
- Liu, F. T., Ting, K. M., & Zhou, Z. (2008). Isolation Forest. In *Proceedings of the Eighth IEEE International Conference on Data Mining*, (pp. 413–422). IEEE. <https://doi.org/10.1109/ICDM.2008.17>
- Seong, K. D., Nguyen, H., Ohn S., & Park J. S. (2005). Fusions of GA and SVM for Anomaly Detection in Intrusion Detection System. In: *Advances in Neural Networks* (pp. 415–420). Springer. https://doi.org/10.1007/11427469_67
- Sperandei, S. (2014). Understanding Logistic Regression Analysis. *Biochemia Medica*, 24(1), 12–18. <https://doi.org/10.11613/BM.2014.003>
- Stamp, M. (2018). *Introduction to machine learning with applications in information security*. CRC Press.
- Tavallaei, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. In *Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE. <https://doi.org/10.1109/CISDA.2009.5356528>
- Tsigkritis, T., Groumas, G., & Schneider, M. (2018). On the Use of *k*-NN in Anomaly Detection. *Journal of Information Security*, 9(1), 70–84. <https://doi.org/10.4236/jis.2018.91006>
- Udhani, S., Alexander W., & Masooda B. (2019). Human vs Bots: Detecting Human Attacks in a Honeypot Environment. In *Proceedings of the 7th International Symposium on Digital Forensics and Security*. IEEE. <https://doi.org/10.1109/ISDFS.2019.8757534>
- Wu, W., Alvarez, J., Liu, C., & Sun, H.-M. (2016). Bot detection using unsupervised machine learning. *Microsystem Technologies*, 24(1), 209–217. <https://doi.org/10.1007/s00542-016-3237-0>
- Zhengxi, C. (2001). *Data Mining and Uncertain Reasoning: An integrated approach*. Wiley.

Editorial record: The article has been peer-reviewed. First submission received on 11 March 2021. Revisions received on 19 April 2021 and 29 April 2021. Accepted for publication on 3 May 2021. The editor in charge of coordinating the peer-review of this manuscript and approving it for publication was Stanislava Mildeova .

Acta Informatica Pragensia is published by Prague University of Economics and Business, Czech Republic.

ISSN: 1805-4951
